POLITECNICO DI MILANO
Master's Degree in Computer Science and Engineering
Department of Electronics, Information and Bioengineering

A Data-Driven Analysis of Training Habits in Amateur Endurance Runners

Supervisor: Ing. Daniele Loiacono

Author:
Stefano Cassini, 899176

Academic Year 2020/2021

Alla mia famiglia e chi mi ha sostenuto in questo percorso.

Abstract

Within the running community, coaches and sport scientists often find themselves in disagreement on what are the best strategies to follow for the preparation of their athletes. At the same time, more and more non-professional runners are becoming increasingly interested in trying to optimize their training performance autonomously. In this context, with the recent advent of platforms dedicated to the storage and analysis of sport activities, the amount of data relative to the routine training sessions of amateurs has drastically increased. The scope of this thesis is to exploit such availability of information to perform an analysis on the running habits employed by this type of athletes. To achieve this, we developed a framework able to process all the data relative to the workout sessions and extract the main features characterizing them. In particular, we utilized several methods to estimate some physiological parameters of the athletes, by exclusively relying on workouts that are in no way the result of controlled experiments, so that we could generate a totally personalized and high-level view of their individual activities. Starting from this information, we then carried out a series of experiments to compare the training methods popular among different groups of runners. The results we obtained show a promising potential in this type of approach, however they also highlight a necessity to improve some aspects related to the data processing itself.

Sommario

Nel mondo della corsa, spesso allenatori e scienziati dello sport si trovano in disaccordo su quale sia il metodo migliore da seguire per la preparazione dei propri atleti. Allo stesso tempo, sempre più corridori non professionisti sono interessati ad ottimizzare i propri allenamenti in maniera totalmente autonoma. In questo contesto, con il recente avvento di piattaforme dedicate all'archiviazione e analisi di attività sportive, la quantità di dati relativi alle sedute di allenamento provenienti da amatori è aumentata drasticamente. Lo scopo di questa tesi è quello di sfruttare tale disponibilità di informazioni per svolgere un'analisi sulle abitudini di corsa adottate da questa tipologia di atleti. Per fare ciò, abbiamo sviluppato un sistema in grado di processare i dati relativi alle sessioni e ricavare da esse gli elementi principali che le caratterizzano. In particolare, abbiamo utilizzato diversi metodi per stimare alcuni parametri fisiologici degli atleti, basandoci esclusivamente su allenamenti in nessun modo derivanti da esperimenti controllati, in modo da poter poi costruire una visione personalizzata e ad alto livello di ogni loro attività. A partire da queste informazioni, abbiamo poi eseguito una serie di esperimenti per confrontare le modalità di allenamento utilizzate da diversi gruppi di podisti. I risultati ottenuti mostrano delle potenzialità promettenti in questo tipo di approccio, anche se evidenziano una necessità di miglioramento per quanto riguarda alcuni aspetti legati all'elaborazione dei dati stessa.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the years, both research scientists and running coaches have focused their efforts on trying to understand the secrets behind training for performance maximization. The two sides, however, have always been in contrast on most of the major subjects central to the debate, as pointed out by Magness in his book [23]. On one hand trainers almost solely base their knowledge on actual field experience, on the other hand researchers seem only interested in finding the optimal workout that works well in all situations. As a consequence, each diverging point of view always seems to fail considering some important aspects of the problem itself. In particular, scientists often rely on overly supervised and short-term studies that have limited practical use in a real environment. On the contrary, it's hard to find coaches willing to experiment training methods different from the ones they have been traditionally taught.
In this context, lately it has become increasingly widespread the need among amateur runners to optimize their training routines in order to obtain more satisfactory results. Thanks to the recent advances in technology and with the introduction of affordable activity tracking devices, this possibility has now turned into a reality. The accessible monitoring of physiological parameters such as heart rate, in fact, has allowed these athletes with no access to professional coaching to start gaining some advanced insights on the quality of their workout sessions. Following this trend, the past few years have seen the birth of a multitude of platforms offering tools to the single runners willing to analyze their training data.
This phenomenon has led to an increasing availability of information that consequently opened up the possibility to apply several data-driven approaches for the analysis of training habits. Particularly, by combining the theoretical concepts with a considerable amount of practical data, it has became possible to extend, in a way, the scientific experimentation far beyond its conventional restricted scope.

1.1 Scope of the thesis

In this thesis, we focus our attention on amateur endurance runners and try to understand their training habits by performing a large scale analysis that includes athletes coming from different parts of the world. For this purpose, we develop a framework able to directly process extensive amounts of raw workout data, without the need of additional information outside of the running sessions themselves. In order to extract the various aspects characterizing a possible training routine, within this architecture we test different approaches that deal with the unreliable and/or incomplete information provided. In particular, great attention is given to the personalized estimation of the physiological parameters and to the creation of an high level view for each running activity. This allows us to introduce several functions that utilize the in-depth knowledge generated from the low level activities to describe in detail the training patterns present among the workouts. Finally, we design a set of experiments to search for the influence of several factors on the training habits observed in the dataset. Namely, the effect of the athletes' nationality and the seasonality of training are considered by analyzing the weekly workout routines of the runners in exam.

1.2 Thesis structure

The thesis is structured as follows.
In Chapter 2, we introduce the main concepts related to running and the theory behind its training, along with an overview of the approaches employed by companies and researchers aiming to provide useful tools to the athletes.
In Chapter 3, we describe the framework developed in order to process the runners workout data for analysis purposes. Particularly, the individual athlete's parameters estimations and high level activities view are here explained in detail.
In Chapter 4, we illustrate the design phase of our experimental analysis including a description of the data collection process, the choices relative to the methodologies employed and the statistical workflow used for the final tests.
In Chapter 5, we present and comment the results obtained from the experiments executed during the analysis.
In Chapter 6, we draw the conclusions of our work and propose some future developments.

Chapter 2

Background

In this chapter, we will introduce the theoretical aspects related to the sport of running and its training, as well as the modern approaches employed by companies and researchers of this field. First, some base concepts and definitions will be addressed by also considering the tools available nowadays to athletes. Then, we will present the theory behind training optimization and how it evolved historically through the work of several sport scientists and coaches. Lastly, we will give a general overview on how these concepts are actually implemented in the most popular fitness tracking platforms and what efforts have been made to take advantage of the huge amount of data available.

2.1 Preliminary concepts about running

When considering a specific running activity, or running in general, there are many terms that are commonly used in order to point out its various aspects. The majority of this vocabulary is related to what can be objectively measured during the course of an effort and, depending on the metrics used, multiple vocables may refer to the same concepts.
Therefore, before tackling more advanced notions associated to running and its training, it's necessary to first introduce the definitions and terminology that will be used in the current and following chapters.

2.1.1 Physiological measurements

Modern activity trackers offer a lot of functionalities when it comes to sampling and processing running data. This information is often stored and subsequently analyzed in order to gain insights on the quality of training on a short and long term period. At the level of single activity several parameters can be measured to get an objective view of the actual effort. Here follows an account of some of the most important ones along with their relative metrics.

Duration and distance

The most simple and intuitive concepts in this context are the ones referring to the duration and distance of a run.
Whilst the total time of a running session can be universally represented by means of

seconds/minutes/hours time spans, distance is usually expressed either in kilometers or miles, depending on the unit of measure adopted by the specific athlete.
To track the evolution of these parameters during an activity, most devices make use of an internal clock along with a dedicated GPS [1] functionality, which is consistently able to pinpoint the runner location inside a geographical map. This setup allows for a quick calculation of the instantaneous traveled distance that can consequently be employed for both live tracking and a posteriori analysis purposes.
In addition to the basic concept referred to a single running effort, when considering multiple days or activities, the term *volume* is also used in order to indicate the amount of distance covered in total (e.g. a weekly volume of 30 miles). As we will see later, tracking training volume over time can be a good strategy when trying to understand the impact of one's workout routine on the body. Therefore, it's a fundamental tool that should be available to every type of running athlete.

Speed and pace

Running speed is one of the main objective indicators of effort during a run, with a standard unit of measure that is commonly denoted either in km/h (kilometers per hour), mph (miles per hour) or m/s (meters per second).
As usual, using activity tracking devices, we can get an estimate of its instantaneous values thanks to the automatic GPS calculations that are executed at each instant.
In general, runners are mostly concerned with the statistics related to the speed maintained throughout their training sessions (e.g. average or peak speed). However, with the advent of live tracking, this real-time information can be also used to follow a specific target velocity required by the workout at hand.
As an alternative way of expressing the same concept, *running pace* can be used in place or jointly with the notion of speed. This metric is as widely, if not more, utilized since it can be considered more intuitive for an athlete, being generally defined as the number of minutes it takes to cover a mile or kilometer. Having said that, obviously one can always convert a value between these two metrics if necessary, given that they are in fact interchangeable.
Finally, by taking in consideration the speed (or pace) characterizing a specific effort, is also possible to distinguish the different categories of running styles that typically range from easy jogging all the way to sprinting.

Heart Rate

Thanks to the recent advances in affordable and portable technology, *heart rate* monitoring has gained popularity among runners and scientists.
The aim of this measurement is centered around the study of the beating rate at which the heart muscle operates during different types of efforts. As a consequence, when discussing the relative sampling results, an HR value is always expressed in bpm (beats per minute).
Athletes typically keep track of heart rate values in order to estimate the intensity of a workout, since it can be considered a good indicator of the actual strain experienced by

[1] `https://en.wikipedia.org/wiki/Global_Positioning_System`

the organism. Intuitively, when the body is subject to increasing physical demands, the heart starts beating faster to provide the oxygen needed by the entire muscular system for its function. On the contrary, as the intensity lowers, we can observe that this rate decreases down to its resting value.
This relationship between heart rate and effort can therefore be used to gain useful insight on the quality of training. However, for a correct personalized analysis, some of the following individual parameters must be first measured by the single athlete:

- Resting Heart Rate (HR_rest): how fast the heart beats when not executing any physical activity (usually measured sitting or laying down).
- Orthostatic Heart Rate (HR_orth): slightly higher than HR_rest represents the heart rate when an individual is standing.
- Maximum Heart Rate (HR_max): the fastest rate an individual's heart can beat (must be tested in a maximal effort).
- Heart Rate Reserve (HRR): difference between HR_max and HR_rest (or HR_orth). Can give a better idea of exercise intensity as it considers the personal range of bpm values.

Modern devices allow HR measurements to be taken continuously and at each instant, achieving an acceptable accuracy even when utilizing the cheaper wrist based monitors.

Cadence

Another critical aspect of running, referred as *cadence*, relates to the time that elapses between each ground contact of the feet. To determine its value, we count the number of steps, or alternatively strides (considering one foot), taken in one minute.
The study of this parameter has been the focus of a certain number of studies among the sport science community. In particular, researchers have often tried to understand the impact of cadence on running form and injury prevention [21]. In general, it turns out that an optimal rate for most people can be considered around 90 strides/minute (or 180 steps/minute), which is slightly higher than what is observed in the typical amateur runner.
Nowadays, aside from manual counting, most running trackers offer a cadence monitoring feature that can provide fairly reliable estimates either using an internal accelerometer or a sensor placed in the athlete's shoe.

2.1.2 Running on different terrains

Depending on the type of terrain where an activity performed, different aspects need to be taken in consideration and some specific metrics are used.

Altitude, cumulative elevation gain and incline

When running on flat ground the parameters described in section 2.1.1 are more than enough to get a good idea on the quality of an activity. However, in reality there can be a considerable amount of training performed outside of controlled environments (such as tracks), especially with amateur runners. For this reason, in the recent years a need to track efforts on hilly or uneven terrain has gained interest among fitness enthusiasts. Thanks to modern activity trackers and their integrated sensors [2] [3], it's possible to acquire elevation data in order to quantify these aspects easily and with remarkable precision.

Altitude (or elevation) is the distance, commonly expressed in meters, from the Earth's sea level and can be estimated essentially with two different approaches. The most reliable method, if correctly calibrated, requires using a barometric altimeter sensor which measures the changes in air pressure in order to determine elevation. In the absence of a barometer, trackers can rely on GPS data gathered during the activity and elevation plots of the surrounding area.
From altitude other metrics can be derived such as *cumulative elevation gain*, which is defined as the sum of every gain in altitude registered during the activity (elevation losses are not included in the count). This measure comes in very handy when trying to quantify the effort of a run because, by not considering descent, can clearly differentiate a flat course from a one with comparable up/down hill which would result in a zero value if subtracted.
Finally, *incline* (also referred as slope or grade) is computed as the ratio of change in altitude over the distance traveled and is indicated by a percentage. In the case of a negative incline, we usually refer to it as decline.

Treadmill running

Indoor running can be achieved by means of machines such as exercise treadmills. These can be valid alternatives to outdoor running, for example in case of adverse weather. They offer some advantages such as options to set a constant pace or incline but, when it comes to tracking the activity, can pose some problems. In fact, unless the treadmill comes equipped with an integrated tracker, measurements taken with a GPS watch rely on using the accelerometer to estimate the distance covered and the speed reached. Some alternatives require the use of pedometers[4] or foot pods[5] to get better estimates comparable to the outdoor ones.

2.1.3 Activity as a collection of data streams

With all the metrics in sections 2.1.1 and 2.1.2 introduced, is now possible to give an informal definition of running activity based on its measurable variables.
If during a run we sample the available data in a more or less regular fashion, we can consider the running activity simply as the collection of these data streams. Because of this, in the next sections these signals will often just be referred as "streams".
Naturally the possible time series representing any given run can be many and some metrics have been omitted for simplicity (e.g. temperature). Nonetheless, the general concept of data streams, in relation to a running activity, should now be clear to the reader.

[2] https://support.garmin.com/en-US/?faq=dRY70Lc6yv2oY3eam1ZWxA
[3] https://support.polar.com/us-en/support/altitude_measurement
[4] https://en.wikipedia.org/wiki/Pedometer
[5] https://en.wikipedia.org/wiki/Inertial_footpod

2.2 Fundamentals of training theory

In this section we will provide an overview on the most important theoretical concepts related to the training theory behind the sport of running. Since all this information will be used in the next chapters, knowing its details it will help understanding the decision made when developing our own framework for this thesis.

2.2.1 Stress and adaptation

The building block on which all the training theory concepts are based, is the notion of how the body reacts when presented with a stressing stimulus and how it consequently adapts to it. Exercise training, indeed, has the only objective of continuously provoking these adaptations in order to accustom the physique to increasingly difficult efforts by reaching new levels of fitness.
This basic principle, often referred in the literature as *supercompensation* [3], is the basis of progressive overload and can be applied to many different sports (e.g. weight training). The process can be easily summarized by looking at Fig. 2.1.

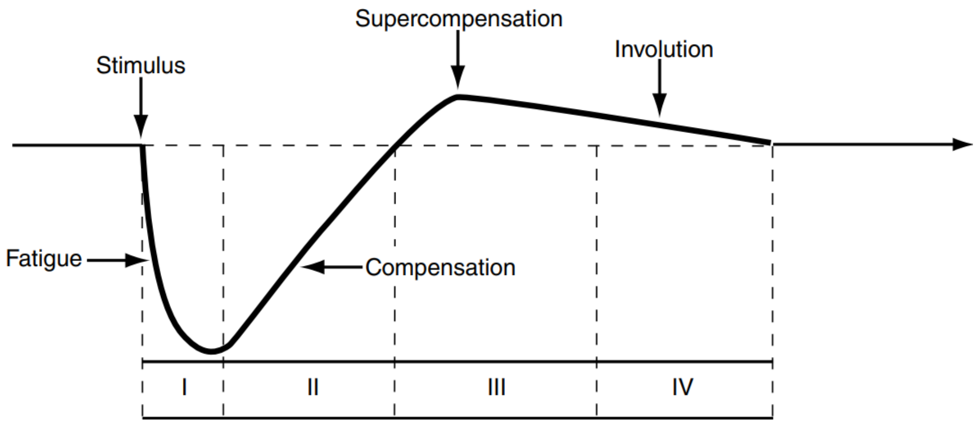


Figure 2.1: Supercompensation phases after a training session (from [3]).

First off, a training stimulus is applied resulting in decreased fitness with respect to the initial baseline (Phase I). This is due to the fact that a state of fatigue is induced which, in turn, reduces the functional capacity of the body. At this point a phase of compensation slowly returns the athlete to a recovered state until it reaches the previous baseline (Phase II). Following this phase, if recovery is sufficient, supercompensation brings the performance further up into a new increased level, thus employing adaptation (Phase III). Finally, if the time between two consecutive stimuli is too long, the performance will start to fade leading to involution (Phase IV).
By repeating this process, one can progressively reach ever increasing levels of fitness. However, if the recovery phase is not consistent enough, the body cannot reach a state of supercompensation and might even start decreasing fitness in the worst cases. This phenomenon is called *overtraining*.
As a side note, is important to consider that the amount of stimulus and recovery needed in order to correctly apply this process changes from one individual to another. Thus, as we will see, training individualization is of fundamental importance.

2.2.2 Athlete performance parameters

When running, various body systems react in different ways to the increasing levels of stress posed by the effort. Several biological parameters affect the performance of an athlete and, as a consequence, are critical aspects to be considered when trying to optimize training. In fact, by considering these features, it's possible to get an idea on the potentials of a specific athlete and accordingly tailor individual training intensities and adaptations. Listed below are some of the most important concepts related to this matter.

VO2max

Maximal oxygen consumption, in short $V\dot{O}2$ max (the dot over V indicates per unit of time), refers to the maximum rate of oxygen consumption attainable during incremental exercise and it's expressed in $\frac{mL}{(kg \cdot min)}$ (milliliters of oxygen per kilogram of body mass per minute). This is considered one of the most important indicators of endurance performance since it represents the athlete's aerobic ability (the capacity of creating energy in presence of oxygen) and high values of this parameter have been observed among world-class athletes. In order to measure it, the athlete is set up on a treadmill while wearing a mask to capture the inhaled and exhaled air. Then the intensity is progressively increased until maximum effort is reached. Observing the data, one can typically see that the oxygen consumption increases with intensity and eventually reaches a maximum, although the workload is still increasing. This happens because the body has reached its maximum aerobic capacity and has to utilize other means (anaerobic systems) to fulfill the energy needs. This parameter has gained a big popularity in the past thanks to its measurability and some studies [25] were made specifically to find a way to maximize it. In reality, as Magness states in his book [23], there has been a lot of research discrediting the use of this variable to estimate an athlete's performance. For example, it has been observed that in many elite athletes $V\dot{O}2$ max does not change, even with performance improvements. This is a consequence of the fact that, in reality, there are many more variables involved with the process of oxygen to energy transformation. Nonetheless, some approached based on $V\dot{O}2$ max can still be of interest as we will see shortly.

Anaerobic/Lactate Threshold

Historically, lactate (or lactic acid) was considered the main cause of fatigue during anaerobic exercises. However, in the recent years researchers revealed that lactate concentration is not actually at source of the problem. Instead, it's the buildup of by-products created during intense effort that is causing this condition and lactate concentration just happens to be highly correlated with it (as also stated in [23]). Despite this discovery, scientists have found great interest in studying a particular moment in the process of lactate accumulation. This point, which corresponds with a substantial rise in blood lactate, has been given various names, but most notably among runners it's known as Anaerobic or Lactate Threshold. Put in other words, it can be expressed as the transition stage that marks the passage of body mechanisms from a mainly aerobic energy production to an heavily anaerobic one.
Monitoring this parameter in athletes has proven very useful when trying to estimate running performance and is often considered to be a better predictor than $V\dot{O}2$ max. In

fact, the faster is speed when reaching the anaerobic threshold, the better will be the athlete's running ability, given than an effort around or above this point can be usually sustained for about an hour maximum.
Considering the importance of this variable, several methods for determining the threshold point have been developed. We will focus on one method in particular that has the main advantage of not needing specific equipment for blood lactate detection: the Conconi test.

In 1982 Prof. Francesco Conconi published a paper [6], later refined in 1996 [7], where he described a test able to determine the anaerobic threshold of an athlete without invasive laboratory procedures. The test is based on the idea that, by monitoring heart rate during an effort of incremental intensity, a change in relationship between speed an heart rate occurs when the anaerobic threshold is reached. In fact, as shown in the example of Fig. 2.2 present in the first paper, initially heart rate grows linearly with speed, but at some point there is a deflection that seems to almost precisely coincide with the sharp accumulation of blood lactate.

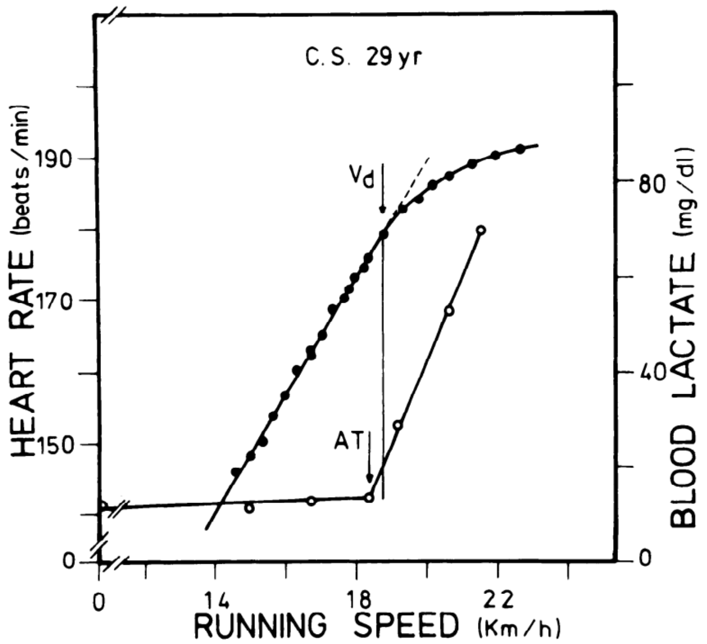


Figure 2.2: Speed and heart rate relationship compared with blood lactate levels (from [6]).

The hypothesis advanced by Conconi is that this deflection is caused by the activation of the anaerobic lactacid mechanism which, as we have seen, marks the lactate threshold.
Despite the wide success of this test in the running community and among scientists, some studies such as [8] have questioned the validity of this finding and is still considered a controversial topic by many.

Tightly connected to the concept of lactate threshold there is a measure that is often referred as *functional threshold pace*. It's defined as the maximal running pace that an individual can sustain for an effort of 45 to 60 minutes. In other terms, this is just a more practical approach to the problem we have been discussing and, considering what we previously said about the maximum duration of an anaerobic activity, can be easily approximated with the speed observed at the point of anaerobic threshold.

Running Economy

There is a third interesting component that can heavily influence running ability and is associated to the athlete's efficiency.
Running Economy measures the utilization of available energy during an aerobic running effort at sub-max speeds [9]. Traditionally, in order to estimate this parameter several values of maximum oxygen consumption (VO_2) are sampled at different constant speeds. If two athletes are compared, the one with better running economy will have a smaller consumption when considering the same velocity.
Unfortunately, this approach seems flawed as it doesn't account for all the variables involved in the process but instead merely reflects their consequences. In this regard, Magness [23] proposes a three factor idea of running economy by considering bio-mechanical, neural and metabolic efficiency. The first component refers to anything that can impact the mechanical cost of running such as cadence, foot strike, body structure or elastic energy. Neural efficiency, instead, focuses on the communication between the nervous system and the muscles by taking in consideration factors that improve signaling and contraction. Lastly, metabolic efficiency deals with how oxygen is delivered and utilized as well as how energy resources are used during short and prolonged efforts.
As a practical mean of measuring running efficiency, the ratio between speed (output) and heart rate (input) is often used to track running economy over time. Intuitively, a lower heart rate for the same pace or a faster speed at the same heart rate indicates an increased efficiency when observed consistently over time. It should be noted, though, that no ideal value exists when considering this parameter as it should only be used to track progress relative to one's historical data.
As a final remark, it's worth mentioning that this relationship is sometimes also utilized to detect the phenomenon of *heart rate drift* [6] that shows an increased heart rate as exercise progresses due to the body getting fatigued.

2.2.3 Training Intensity

Running workouts are usually performed at various intensities in order to target the different physiological systems that contribute to improved running performance. Balancing these stimuli is key for a successful progression and researchers have spent a lot of effort in trying to understand their effect on the important parameters involved in running performance.
Here are presented the different approaches commonly adopted when trying to assess the intensity of a running workout.

Zones approach

One of the most practical and utilized ways of expressing the intensity of a run is to simply divide pace or heart rate values into different zones and then consider the time passed in each one during the activity.

[6] https://fellrnr.com/wiki/Heart_Rate_Drift

In order to do this a set of zones is created, by using percentages of a certain athlete's parameter (e.g. max heart rate, speed at $V\dot{O}2$ max, etc.), in a way that tries to separate the distinct energy systems involved at different intensities. In practice, most of these divisions consider some broad intensity categories that can be usually summarized as:

1. Sub-Maximal
2. Lactate Threshold
3. $V\dot{O}2$ max
4. Supra-Maximal

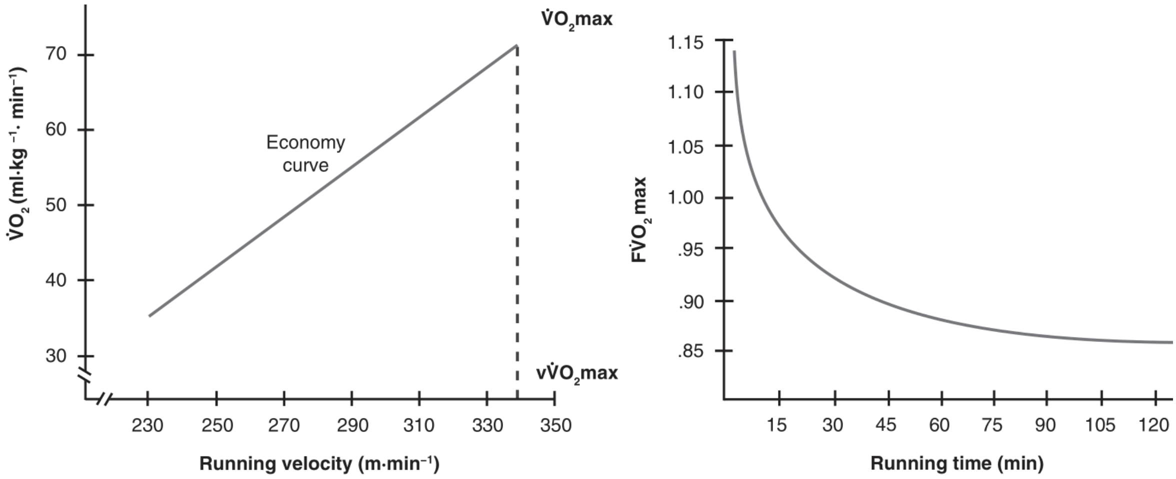
The first one refers to all the exertions that are characterized by a mainly aerobic contribution. Often, this general zone can be further divided in low and moderate sub-zones but, in general, refers to all low intensity efforts that can potentially last several hours.
A moderate intensity area, instead, is found around the Lactate Threshold point where, as we have seen previously, the body starts to transition towards anaerobic energy resources. Since this is a very important phase, sometimes the zones are actually specified as a percentage of LT heart rate or pace.
Lastly, $V\dot{O}2$ max and supra-maximal zones represent the high intensity spectrum of the scale. Activities of this intensity span from very hard to maximal efforts and can be only sustained for few minutes.
Obviously, this general configuration is not fixed and many zone based models adopt slightly different conventions. For example, heart rate is often divided in five zones by considering percentages of either maximum or lactate threshold heart rate.
When available, pace and heart rate can also be used in conjunction in order to have a better final estimation of the effort. In fact, both have some advantages and disadvantages depending on the type of workout performed. Pace has the advantage of being an objective measure but it can be difficult to follow in practice, especially in low intensity runs, as it's often viewed as something to exceed rather than respect. Heart rate, on the other hand, can have problems when applied to high intensity workouts that have a lot of speed variations. This is due to the phenomenon of cardiac lag which consists in a delay between pace changes and corresponding heart rate variations.

Despite being very popular, this approach based on zones is not exempt from problems considering that, intuitively, the real mechanisms behind the efficacy of training are never isolated and cannot really be divided into precise boundaries. As we will se in the Periodization section, sometimes training intensities should just be expressed by taking into account a precise goal pace and deriving their specificity in relation to it.

Jack Daniels' VDOT

An interesting take on training intensity estimation is given by the popular VDOT tables of world famous coach Jack Daniels.
In 1979 with the help of Jimmy Gilbert he published a book [10] that contained a series of tables predicting all-out racing times for a multitude of distances. This was achieved by measuring both the oxygen demands at various velocities and the aerobic capacity

of many runners of different abilities. From this data, knowing their best performance at different competitive distances, they derived two regression equations. The first one, whose curve is shown in Fig. 2.3a, relates $V\dot{O}2$ (not the maximum value) to running velocity by assuming an average economy for all runners. The second equation instead, illustrated in Fig. 2.3b, describes what percent of an individual aerobic capacity ($V\dot{O}2$ max) the runner is capable of working at and for how long.



(a) Relationship between velocity and oxygen cost given an average economy. $vV\dot{O}2$ max represents the velocity at maximum oxygen consumption.

(b) Relation of race times and fractions of $V\dot{O}2$ max.

Figure 2.3: Daniels regression curves (from [9]).

Given these two equations and knowing a runner race performance (all-out effort) is possible to then estimate their $V\dot{O}2$ max with a value that Daniels calls $VDOT$. Consequently, race times can be predicted for set distances starting from the personal VDOT index of the athlete, as demonstrated by the tables provided by Daniels.
It's important to highlight that the VDOT value does not necessarily correspond to the real value as measured by a laboratory test. This is also stressed by the author himself but it's not a problem since, as we have seen, real race performance is a better indicator than the actual $V\dot{O}2$ max.

From the VDOT value Daniels then uses a similar approach, as we have seen previously, that creates several intensity zones by expressing them as percentages of the personal VDOT index [9]. These pace zones are: Easy, Marathon, Threshold, Interval and Repetition.

Rating of Perceived Exertion

A totally different approach to the problem is based on using the subjective perception of the athlete, rather than objective data, in order to quantify the intensity of a given workout.
The RPE scale, developed by Gunnar Borg [4], is a widely employed method in sports and medicine used to monitor an individual effort during a physical exercise. As shown

in Fig. 2.4, it utilizes a numerical scale from 6 to 20 (or the alternative 0-10) to assign points based on the athlete's answer on how hard the workout felt.

6	
7	Very, very light
8	
9	Very light
10	
11	Fairly light
12	
13	Somewhat hard
14	
15	Hard
16	
17	Very hard
18	
19	Very, very hard
20	

(a) Original scale.

0	Nothing at all	
0.5	Very, very weak	(just noticeable)
1	Very weak	
2	Weak	(light)
3	Moderate	
4	Somewhat strong	
5	Strong	(heavy)
6		
7	Very strong	
8		
9		
10	Very, very strong	(almost max)
•	Maximal	

(b) CR10 scale.

Figure 2.4: Borg rating of perceived exertion scales (from [4]).

Using RPE can have several advantages, compared to objective approaches, because of its ability to detect any factor that affects the body that might not be otherwise measurable (e.g. sleep quality, stress). However, a downside of this method, apart from the fact that it needs time to be familiarized with, is that the rating can be influenced by fatigue when used on longer workouts.

2.2.4 Training Load

As we have seen in section 2.2.1, the objective of training revolves around the concept of stressing the body and, through recovery, cause it to get stronger. Other than assessing the intensity of a workout, it's then of fundamental importance being able to quantify the amount of strain placed on the athlete's body. For this purpose, various measures of either external (e.g. distance) or internal (e.g. heart rate) load have been proposed over the years.

In general, load (or effort) can be expressed as the interaction between intensity and volume (typically duration) of training. As a matter of fact, two workouts can have totally different intensities but create the same amount of stress depending on their duration.

By tracking training load over time, it's also possible to gain insights on the runner personal response to specific sessions, avoid risk of injury and make sure enough stimuli are applied to the body in order to make progress.

Banister TRIMP

Dr Eric Banister developed the training impulse model (TRIMP) [2] with the objective of trying to measure the internal load of an activity by considering as inputs heart rate responses and duration. The method is computed using the following equation weighted based on the relationship between heart rate and blood lactate:

$$TRIMP = D \cdot (a \cdot \Delta HRratio \cdot e^{b \cdot \Delta HRratio}) \tag{2.1}$$

With $a = 0.64, b = 1.92$ for men and $a = 0.86, b = 1.67$ for women. D = duration (in minutes) of the session and $\Delta HRratio$ computed as:

$$\Delta HRratio = (HR_{ex} - HR_{rest})/(HR_{max} - HR_{rest}) \tag{2.2}$$

In which HR_{ex} is average heart rate of the session, HR_{rest} is resting heart rate and HR_{max} is maximum heart rate.
The main limitation to this approach is given by the fact that there is a separation only between males and females, which is obviously not the only factor of difference in athletes.

Edwards TRIMP

A second very popular method, known as Edwards TRIMP [13], is based on the idea of dividing heart rate into zones based on percentages of HR_{max} and then multiply the time passed in each zone by an arbitrary coefficient. The proposed zones, each of 10% width, are shown in Table 2.1 along with their respective coefficients.

% of HR_{max}	coefficient
50-60%	1
60-70%	2
70-80%	3
80-90%	4
90-100%	5

Table 2.1: Edward's TRIMP zones and coefficients.

At this point, the total load of an activity is expressed as:

$$TRIMP = D_{zone1} \cdot 1 + D_{zone2} \cdot 2 + D_{zone3} \cdot 3 + D_{zone4} \cdot 4 + D_{zone5} \cdot 5 \tag{2.3}$$

Where D_{zonek} is time passed (minutes) in each of the five zones.
Despite its success, the coefficients and boundaries used in this method are not supported by some particular phenomenon or neither have been validated on the basis of known physiological responses. Additionally, small changes in the boundary points between each zone can lead to big variations of scores. These problems, coupled with the fact that scaling factors are linear, have made the use of this approach questionable for some scientists.

Lucia TRIMP

In 2003 Lucia et al. [22] proposed a modified version of Edwards method that utilizes only three zones demarcated by two ventilatory thresholds. The first point that separates low from medium intensity is found right around the lactate threshold. Instead, medium and hard zones are separated by the respiratory-compensation threshold (where breathing rate changes to address abnormal blood pH). If D_{zonek} is the time passed in each zone (low, medium, high), Lucia TRIMP score is defined as:

$$TRIMP = D_{zone1} \cdot 1 + D_{zone2} \cdot 2 + D_{zone3} \cdot 3 \tag{2.4}$$

Again, these coefficients are not based on scientific evidence (like in the Edwards case) but this time the zones boundaries have a significance, at least in principle, with respect to some physiological factors.

Other measures

Aside from the aforementioned measures, several other TRIMP modifications exist and, as we will see later, some platforms have even developed their own algorithms by exploiting this theory. Some methods, for instance, try to capture the individualization of athletes by weighting the score based on group or personal blood lactate response curves [29] [24]. Daniels, in his book [9], provides a point based system to quantify load by assigning scores depending on intensities expressed as percentages of $V\dot{O}2$ max.
Finally, as an alternative to heart rate based methods, Session-RPE [16] applies the RPE philosophy to the training load domain by simply expressing the total effort of an activity as $RPE \cdot minutes$.

Load Balancing

The usefulness of training load metrics is not limited to the single workout domain. In fact, when Banister introduced the original impulse-response model [5], his main objective was being able to predict the changes induced by training load over time. In order to do this, he modeled training as a series of negative (fatigue) and positive (fitness) impulses whose relationship represents performance. In this paradigm, whilst fatigue initially generates a negative effect, over a longer time frame it is outweighed by an increase in fitness. Consequentially, in any moment is possible to compute the difference between fitness and fatigue to determine the *training stress balance*. The idea is that, by manipulating load, is then possible to balance this relationship and improve in the long term.
Building on this concepts, the index of *acute: chronic workload ratio* (ACWR) was recently introduced by Gabbett and colleagues [19].
Here, the acute workload represents fatigue and is typically computed by taking the 7-day rolling average daily load. The chronic workload (fitness) instead is created considering a longer period of time (usually 21 or 28 days). The exact calculation of these values can be performed in different ways but often the exponentially weighted moving average [33] is preferred since it better captures the decaying nature of fitness. Given these two values then ACWR at any time is simply computed as:

$$ACWR = \frac{\text{Acute Load}}{\text{Chronic Load}} \tag{2.5}$$

Several studies [11] have found interest in analyzing the relationship between this measure and the risk of injury in different sports. Fig. 2.5 shows the ranges of values that have been found correlating the most with under, optimal and over training.
This shows, for example, that a 'sweet spot' has been found approximately around values of 0.8-1.3. However, it's important to notice that these ranges should be used carefully and may not reflect all individuals or disciplines.

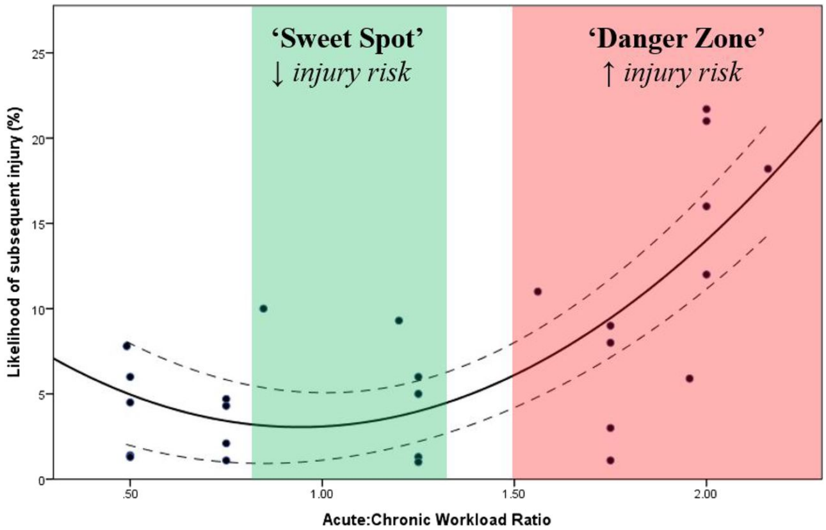


Figure 2.5: Relationship between ACWR and injury risk (from [17]).

2.2.5 Workout types

When it comes to putting into practice the theory concepts we have seen until now, different types of runs can be used in order to create and execute a successful training program. Although it's difficult to categorize every single type of activity, a collection of universally recognized and practiced workouts exists. In his book [23], Magness provides a traditional classification that also considers the different physiological effects involved in each type of session. In a similar fashion, here are illustrated some of the most important classes of workouts available to runners.

Mileage and recovery runs

According to coaches and sport scientists, the runs belonging to this category are at the base of general training and should ideally make up the majority of total volume (about 80% [15]). Nonetheless, often amateur runners struggle to follow this ratio and easily underestimate the importance of low intensity running. Here, for convenience, these activities are summarized as distance, recovery and long runs.

A 'standard' *distance run* is simply a workout performed at easy to steady pace. Its main function is to increase aerobic capacity and accustom the body to the bio-mechanics of running. Despite its simplicity, it plays a major role in every athlete preparation since it can be considered the main source of the general endurance required to face more difficult workouts.
At an intensity step below, the *recovery run* embodies what can be thought as an active approach aimed at enhancing the phase between or after harder sessions. In order to have the desired effect, this type of activities must be performed at a slower pace than normal and typically for a short amount of time. The stress applied to the body should be minimal so that a quicker adaptation occurs without impairing the supercompensation effect. Sometimes, these recovery runs can also be split throughout the day to increase

the beneficial effects and lower their individual stress.
By prolonging a standard run in distance and time, it's possible to obtain some stimuli that are otherwise absent when dealing with a shorter effort. This popular workout, known as *long run*, is considered by many a fundamental part of distance running and it's an essential tool for the preparation of longer races like marathons. The main adaptations brought by such an activity are relative to structural and fuel changes along with muscle fiber activation. In fact, if initially the easiest fibers are used, at some point they become fatigued and are necessarily replaced by the less used ones. Additionally, during a long run the body is forced to employ all the available fuel resources and adapt the joints to a continuous and extended stress.

Tempo/Threshold Running

This category refers to all the workouts characterized by a longer sustained effort. Although the intensity can vary, the typical speed of these runs lies between marathon and anaerobic threshold pace. The idea behind this type of training is to let the athlete familiarize with a comfortably hard pace which, however, should be not extreme and unsustainable. Among the physiological benefits brought by these workouts, typically there is an increment in speed at threshold (caused by an increased efficiency at clearing exercise byproducts) and a better overall running economy.

Interval/Repeat workouts

Another important class of workouts is the one referring to activities that alternate high intensity efforts with rest or low to moderate recovery periods. This type of sessions, often referred as interval training or repeats, are usually divided in sets where each set is composed by a number of same distance repetitions (e.g. 2 sets of 4x800m). Since the pace during a repetition can be rather challenging, in between each effort there is a phase of recovery that can be either static or dynamic. In the first case, the athlete waits a certain amount of time standing or sitting. In the other case, the recovery time is spent walking or jogging. Depending on intensity, rest periods, number of repetitions and pace changes the effect of this training can be varied. For example, other than speed and anaerobic development is possible to elicit adaptations in the aerobic system by keeping the interval distance short at an intensity that is higher than normal aerobic runs.
Another way of manipulating repeat workouts is through alternations where, instead of a rest phase, high an low efforts are alternated in such a way that full recovery is not possible.
Lastly, sprints can be seen as a particular type of repeat training where intensity is maximum and distances are kept short. Some benefits of these workouts include better running mechanics, explosive power and increased strength. Moreover, by executing them on hills instead of flat terrain one can obtain approximately the same effects at slower and safer speeds.

Workouts additions

Several elements can be added on top of the base workouts we have seen until now. For instance, is very common to precede a moderate or intense session with a warm-up

stage to let the body prepare for the subsequent effort. Similarly, an hard workout is often followed by a cool-down phase where a slow pace is held in order to enhance future recovery.
However, other than these extensions solely aimed at enhancing a given workout, there are some actual additions that can be inserted during the session to change its resulting adaptations.
Strides, for example, are short repeats that can be added at the end of a normal workout to include a speed component and work on bio-mechanics while recruiting muscles in a semi fatigued state. If these repeats are added in the middle of a run instead (*surges*), they can help the body adapt to transitions between paces and get it used to utilize lactate as fuel for the slower parts of the activity. Sometimes, especially if the speed changes are arbitrary, this is referred as *fartlek training* (speed play).
Finally, *progressions* are a very popular way of enhancing the stimulus of a standard run. They are carried out by gradually increasing the intensity of an activity so that different systems of the body are employed during the entirety of the workout. These runs can be broken up in several ways to produce distinct effects (e.g. thirds). If the progression happens only at the very end of a run in the form of a fast finish, the workout is usually called *pickup*.

2.3 State of the Art

2.3.1 Tracking hardware and suites

Nowadays a wide range of hardware solutions is available to athletes and fitness enthusiasts willing to track their running activity. Beginners can take advantage of modern smartphones which include several integrated sensors and benefit from a rich app environment. For more advanced users, however, it's generally worth investing in dedicated devices such as GPS watches and heart rate monitors. These specialized equipment allows for a detailed data collection and often provides an included software suite that gives access to some kind of training analysis (e.g. Polar Flow [7], Garmin Connect [8], etc).
More recently, foot pods (sensors placed in the runner shoe) like Stryd [9] have gained popularity as well thanks to their vastly increasing accuracy compared to traditional watches. These devices can offer fairly precise estimations by analyzing the foot movement during the course of a run. Also, since a GPS signal is not required, it's always possible to collect their data, even in adverse conditions.

2.3.2 Activity tracking platforms

The availability of training data has given birth to a myriad of online platforms offering tracking and data analysis capabilities to their users. These services allow athletes to upload their training activity by interfacing with the most popular fitness devices. Users can keep a log of their past workouts and get some insights on the quality of their progress thanks to advanced visualizations and statistics. Sometimes these platforms also incorporate some social network features such as the ability to share training activities publicly or with friends. This often results in the creation of communities that can be a source of increased motivation for athletes that want to better themselves and show off their improvements. However, getting too competitive when comparing performance with others can occasionally result in a negative impact on training quality.
With more than 75 million users, **Strava** [10] is one of the most popular digital services occupying a place in the fitness tracking industry. The platform has a main focus towards cycling and running but also supports plenty of other sports. Among its main features, it offers the possibility to upload or record activity directly from the proprietary app and share it with a follower base. Fitness goals are promoted through monthly challenges and it's possible to join clubs or take part in sponsored races. Each workout is analyzed in detail by means of data graphs showing speed, heart rate and other useful visualizations of effort and training progress. Additionally, performance can be tracked considering segments which are portions of user created routes where athletes can compare their times with past efforts or other members in a ranking leaderboard. That being said, in order to access some of the advanced features like training load stats and segments, the payment of a subscription fee is required.
Less concerned with the social aspects and more focused towards analysis, **Training-**

[7] `https://flow.polar.com`
[8] `https://connect.garmin.com`
[9] `https://www.stryd.com`
[10] `https://www.strava.com`

Peaks [11] strongest suit resides in a rich support for training plans, powerful data analysis and coaching capability. This platform, in fact, provides several workout plans that can be purchased and followed by all free users along with the possibility of investing in a real coach who can monitor and create custom training plans. Additionally, a paid version exists where a broader set of short and long term analysis tool is given directly to the athlete.
As a final note, other noteworthy online tracking services include **Final Surge** [12], **Endomondo** [13] and **Today's Plan** [14].

2.3.3 Proprietary algorithms

Some of the aforementioned platforms have developed a few proprietary algorithms that have also been implemented in several other tracking services. In particular, we consider the work done to estimate the equivalent flat pace on a hilly surface and some proposed training load metrics.

Pace adjustments with inclines

When dealing with irregular terrain, pace can be influenced by the grade of incline present during the activity. This results, given an equal running effort, in a slower or faster speed (depending if the grade is positive or negative) compared to what would be in absence of incline. For this reason, sometimes it can be useful to have an idea of the equivalent flat speed when dealing with uphill and downhill segments. Moreover, having this estimate is fundamental in the case that pace is used as a measure of intensity to compute training load.
The approach used by **Strava** to provide a *grade adjusted pace* (GAP) has changed substantially over the years. Initially it was based on the work of Minetti et al. [26] and relied on the measurement of metabolic energy cost observed in a group of elite athletes running at different inclines. More recently, however, the platform exploited a huge amount of their athletes data in order to build a model using the principle of equivalent heart rate [12]. In short this means that GAP represents the pace obtainable by an athlete at the same heart rate while running on level ground. According to the Strava engineers this model seems to have significantly increased accuracy, especially in the downhill portions of activities (see Fig. 2.6).
To solve the same problem **TrainingPeaks** uses a different measure called *normalized graded pace* (NGP) [31].

[11] `https://www.trainingpeaks.com`
[12] `https://www.finalsurge.com`
[13] `https://www.endomondo.com`
[14] `https://www.todaysplan.com.au`

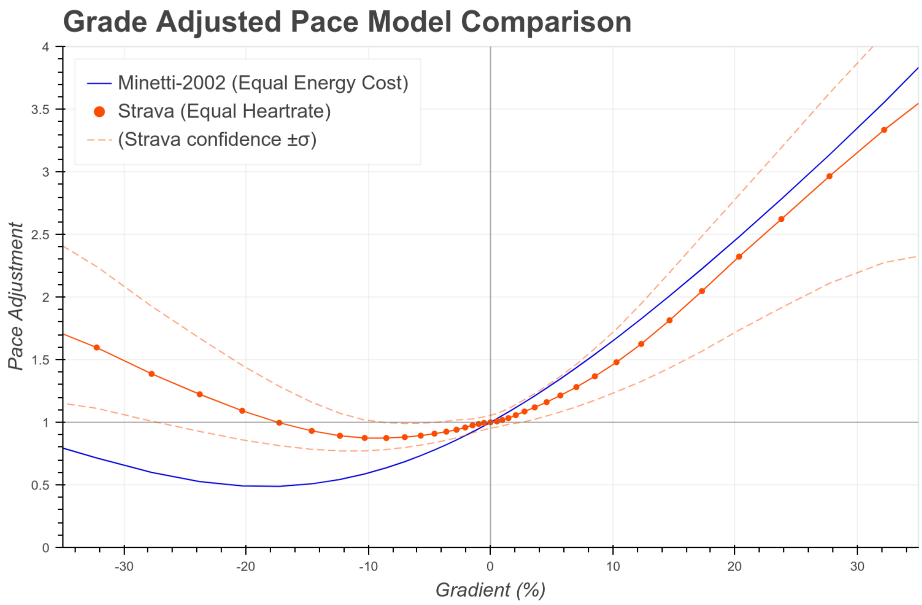


Figure 2.6: Comparison between Minetti and Strava grade adjusted pace (from [12]).

Training load scores

In most training load metrics, such as the ones introduced in Section 2.2.4, heart rate is used to estimate exercise intensity and consequently assess load. This approach can have some disadvantages since heart rate sometimes has problems reflecting the real intensity of a workout.

For this reason, some alternative scores have been created and, in particular, a very popular one called *running training stress score* (rTSS) was developed for **TrainingPeaks** by considering the work of Skiba [27] as described in [30]. This metric considers pace related to an individual's current lactate threshold in order to represent the intensity and stress of a workout. Furthermore, normalized graded pace is used to ensure that uneven terrain is accounted for during the calculation.

The rTSS formula is the following:

$$rTSS = \frac{s \cdot NGP \cdot IF}{FTP \cdot 3600} \cdot 100 \tag{2.6}$$

Where s is the time in seconds of the activity, NGP refers to the normalized graded pace, IF is the intensity factor ($\frac{NGP}{FTP}$) and FTP is the athlete's functional threshold pace (speed in m/s at lactate threshold). Note that by definition and effort of one hour at theoretical maximum intensity (FTP) corresponds to an rTSS score of 100.

This metric is then used by TrainingPeaks to compute some indices relative to the training load balance. In fact, from daily rTSS is possible to derive chronic training load (CTL), acute training load (ATL) and training stress balance (TSB).

In **Strava**, instead, the load of an activity is expressed with the Relative Effort score. This model is based on the concept introduced by heart rate zones TRIMP but with the coefficients of each zone computed in a data driven way. To do this, Strava confronted a large subset of 10km running races to compare different all-out activities and tuned the zones weights to minimize the variance of scores for equivalent efforts [32]. Similarly to TrainingPeaks, relative effort is likewise used to compute the fitness and fatigue parameters of the impulse-response model (load balance).

Chapter 3

Framework

In this chapter, we will outline the architecture developed in order to process training data for our experimental analysis. The design choices and relative implementation will be discussed in detail starting from a low level view of running activity. Next, we will present the work done to estimate the athlete's parameters and consequently build an high level representation of each run. Finally, we will move away from the single entity domain of an individual activity, considering the relationship between multiple training sessions in the context of a broader time period.

3.1 Architecture summary

Before addressing each part singularly, an high level summary of the whole framework is given in Fig. 3.1.

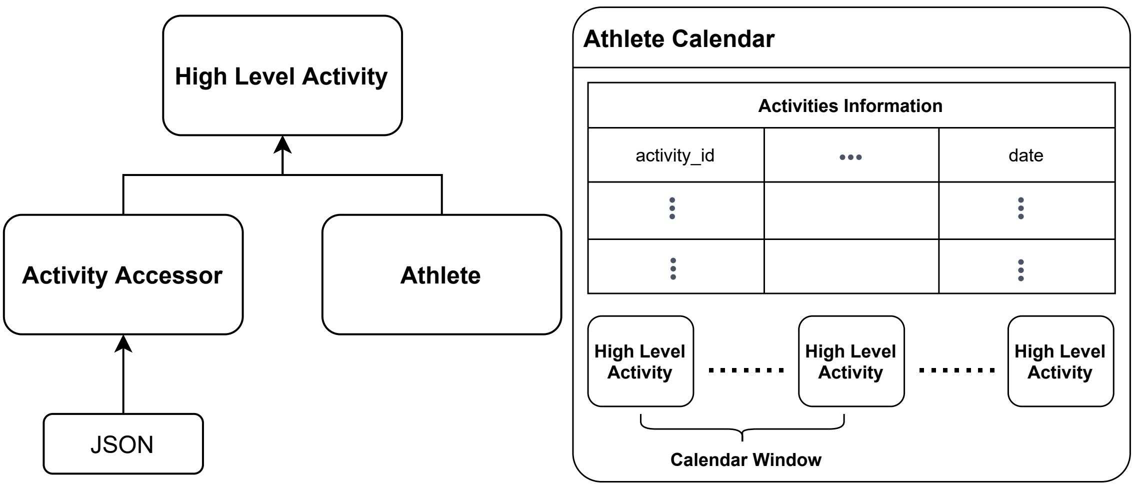


Figure 3.1: Sketch of the framework for a single run (left) and multiple activities (right).

As shown, a single running activity is represented on the left by an High Level Activity object which is composed of two further elements. The Activity Accessor constitutes the low level interface that communicates with a JSON file containing the raw measurements of the run. The Athlete object instead, gives access to some known athlete's parameters

available at the time of the training session. Together, these components allow us to characterize the activity efforts, taking into account the physical condition of the individual with respect to the objective data observed. This method gives access to a set of features that can provide more advanced insights on the quality of training such as load assessment and workout classification.
On the right, instead, we can see the Athlete Calendar object, representing a collection of runs (and potentially other kind of activities) belonging to the same athlete. Since in our analysis multiple training sessions are gathered for each runner, this structure allows us to easily store them and eventually access the date information relative to their execution. Among other things, this framework is responsible for the correct estimation of the athlete's parameters in time and the High Level Activity objects creation for each run. Additionally, it also provides an interface for the extraction of date-based windows (calendar windows) with the possibility to compute interesting statistics on the activities contained inside them for short and long term analysis.

Architectural requirements

When designing the whole framework, we based all our choices around the idea of being able to process workout activities directly coming from online tracking platforms (particularly Strava). For this reason, we had to take some measures in order to deal with a number of problems related to the uncontrolled nature of this data, which is not comparable with the one typically obtained in a laboratory setting.
First off, we could not count on a consistent quality of the information present among the different athletes analyzed. In fact, depending on the hardware used by the runners (e.g. heart rate monitors), we had to manage more or less precise data readings, potentially containing a considerable amount of errors. Moreover, given that all the activity processed was the result of amateur routine training (not performed with analysis purposes in mind), we were forced to work with runs including pauses and other arbitrary behaviors that would be normally avoided when executing any kind of experiment. In this regard, sometimes as far as entire activities tuned out to be completely erroneous and therefore had to be discarded. Lastly, due to the limited information available, we had to estimate all of the athletes parameters only using past workout sessions. This aspect in particular generated a series of issues since, as mentioned earlier, we obviously did not have control on the type of activity performed and consequently used inside the estimation process itself.
All of these aspects and more ultimately pushed us towards building an architecture capable of exploiting as little (and inconsistent) knowledge as possible, to allow for a generalized study that could be extended to most of the workout data available.

3.2 Low level running activity

As we have stated previously in Section 2.1.3, for our purposes a running activity can be considered as a collection of data streams. Typically, data is sampled every few seconds during a run resulting in a bunch of measurements that are not always equidistant in time.
In our case, each activity comes in the form of a JSON file containing these streams as same-length arrays, among which *time* is the most important one. In fact, inside it are contained all the actual timestamps of the activity for which data measurements are available. This configuration allows for a quick mapping between sensors readings and their corresponding time with respect to the start of the activity.
For example, if we consider Fig. 3.2, we can see that each heart rate value corresponds to a precise second of the run (instant 0 → 134 bpm, instant 1 → 137 bpm, instant 4 → 140 bpm and so on...).

"time": [0, 1, 4, 7, 9, 15, 17, 18, 20, 21, 25, 28, 30, 32, 38, 43, 51, 59, 63, 70, ...]

"heartrate": [134, 137, 140, 136, 136, 139, 143, 142, 139, 136, 140, 143, 140, 140, 141, 144, 143, 143, 142, 144, ...]

Figure 3.2: Mapping between heart rate values and time in a running activity.

Sometimes, however, activities can be manually paused by the athlete resulting in the presence of another important array called *timer_time*. Differently from the time stream, the instants here are referred to the seconds of actual activity excluding the time passed with the run paused. By considering either time or timer_time is thus possible to include or skip the manual pauses present in these running activities.
Depending on the device utilized and the sensors available, activities may not all have the same number and type of data streams. Table 3.1 shows all the different arrays that can be found in the JSON of a running activity along with a short description and the datatype of their elements.

The Activity Accessor

In order to work with an activity and easily visualize it, we created a low level interface that encapsulates the previously mentioned arrays and automatically provides values for each second of the run.
This object, called *ActivityAccessor*, gives access to all the available streams by correctly assigning each data value to the corresponding timestamp and interpolating the missing values between the points. In addition to this, some simple information can be requested such as total time of the activity or total distance covered. If the run was paused manually, it's always possible to switch between a representation with and without pauses (if kept a fill value for each stream can be specified). Finally, the interface offers the possibility to check if a run can be considered mostly flat or if it was performed on a treadmill (by looking for the presence of geographical coordinates).
As a visual example, some streams of a sample run are displayed in Fig. 3.3.

Stream Name	**Description**	**Element Type**
time	Contains all the timestamps that have data with respect to the true elapsed time from the activity start	Integer that represents a timestamp
timer_time	Contains all the timestamps that have data with respect to device time (pauses excluded). Only found when manual pauses are present	Integer that represents a timestamp
altitude	Contains the altitude measurements at each timestamp	Float expressed in meters
total_elevation	Contains the cumulative elevation gained at each timestamp (as seen in 2.1.2)	Float expressed in meters
grade_smooth	Contains the terrain grade of incline at each timestamp (as seen in 2.1.2)	Float expressed in percentage of incline
latlng	Contains the geographical coordinates at each timestamp	List with two float values (latitude, longitude)
distance	Contains the cumulative distance covered at each timestamp	Float expressed in meters
velocity_smooth	Contains the instantaneous speed at each timestamp	Float expressed in m/s
grade_adjusted _speed	Contains the instantaneous speed adjusted w.r.t. grade of incline at each timestamp (as seen in 2.3.3)	Float expressed in m/s
heartrate	Contains the heart rate reading at each timestamp	Integer expressed in bpm (beats per minute)
cadence	Contains the cadence at each timestamp	Integer expressed in strides/minute (one foot)
temp	Contains the temperature at each timestamp	Integer expressed in °C (Celsius)

Table 3.1: Streams that can be found in the JSON of a running activity.

Figure 3.3: Speed/Grade Adjusted Speed (blue/purple), altitude (green), cumulative elevation gain (orange) and heart rate (red) of a running activity.

3.3 Athlete representation

In Section 2.2 we discussed the importance of planning training intensities and loads based on the individual athlete performance parameters. For this reason, when we want to characterize and understand a specific running activity, we need to consider the runner physical condition at the time of the effort.
In order to apply this in our case, we decided to create an *Athlete* entity for each low level activity to represent the athlete's status at the time of each session. This object exhibits several values that denote the physiological variables of the runner and can be subsequently used to create an high level view of each activity, as we will see in the next section.
All the parameters of interest are estimated by exclusively using training data composed of real workouts with no need to perform any laboratory test. This approach allows us to easily carry out a large scale analysis by requiring very little information and involvement of the athletes in the process. Furthermore, it offers the advantage of using real performance data observed during training as opposed to measures of variables that are supposed to reflect it (e.g. $\text{V}\dot{O}2$ max).
It should be noted, however, that the quality of the estimates is heavily dependent on the kind of data provided by the athletes. As a consequence, the available equipment and workout diversity play a very important role in the accuracy of the results and, sometimes, can even preclude a successful estimation. Having said that, we provided the ability to rely on multiple parameters for intensity estimation in order to offer a more robust overall outcome.
We will now present the solutions adopted to obtain an estimate of the most important physiological parameters for running performance.

3.3.1 Maximum heart rate

Having an approximate value of the athlete's maximum heart rate can be a significant advantage when trying to assess training intensity or load of a given set of running activities. As a matter of fact, we have seen that many zone based methods actually utilize HR_{max} as a reference value to express the intensity regions during an effort (e.g. Edwards TRIMP as described in 2.2.4).
Typically, this parameter is estimated using some formulas that take into account solely the age of the athlete. This comes from the fact that it has been consistently observed that, in general, the HR_{max} value decreases proportionally to the age of a person. For this reason, many formulas have been developed in this sense and have gained a huge popularity due to their ease of computation [28].
For example, in the following list we mention some of the most used ones:

- Fox formula: $220 - Age$ (the original one)
- HUNT formula: $211 - (0.64 \cdot Age)$ (suggested for active people)
- Tanaka formula: $208 - (0.7 \cdot Age)$ (for people over 40)

The problem with this whole approach is that the only factor of diversity considered between individuals is the age itself. This obviously can result in a poor estimate in some

particular cases since, actually, the parameter mostly depends the individual genetic make up (before the age factor). However, despite this problem, these formulas are still used nowadays by a lot of amateur and professional athletes.
A much better way of finding this value involves a field test where the runner performs bouts of increasing effort while monitoring the highest heart rate observed. This method has the advantage of providing a totally personalized and quite precise result (based on the actual performance, not some general statistic like before) but can be challenging to execute, especially for beginner athletes.

Our approach

Our approach for the estimation of HR_{max} consists in finding the highest heart rate consistently reached by the athlete, considering all the available past workouts.
Since we do not control the kind of runs provided, we can't count on the presence of any HR_{max} field tests in the data. Neither we can assume that the athlete ever reached the true max value in any run. Because of this, we decided to make use of all the sessions available, knowing that the resulting estimate would still reflect the max effort exerted.
The main problem faced with this method, was relative to the noisy nature of the heart rate measurements observed during the activities. For this reason, we adopted a number of steps in order to reduce as much as possible the anomalous values.
A pseudo-code of the methodology used is provided below, followed by an explanation of all the stages involved (the values used in our configuration are shown in Table 3.2).

Algorithm 1: Maximum heart rate estimation

```
Input: A list activities, where each element is an ActivityAccessor.
Output: Integer that represents HR_max.
initialize an array that will contain (speed, hr) samples;
for a in activities do
    gap_ma ← moving average of the grade adjusted speed stream in a;
    hr_ma ← moving average of the heart rate stream in a;
    divide hr_ma in multiple windows based on variance;
    discard the windows below a certain minimum window size;
    for window in remaining windows do
        create a number of samples equal to window size by taking the median
          value of the window on gap_ma;
        create a number of samples equal to window size by taking the median
          value of the window on hr_ma;
    end
    save the samples created in the initial array;
end
filter out samples (speed, hr) under a minimum and over an high percentile of
  speed;
return the value corresponding to an high percentile of all remaining hr points;
```

Parameter	**Value used**
moving avg. window	30 seconds
windows variance thr.	5
minimum window	60 seconds
minimum speed (filter)	2 m/s
maximum speed (filter)	99th percentile
final value (HR_{max})	99th percentile

Table 3.2: Chosen values for the parameters used during the estimation of HR_{max}.

First of all, a set of activities containing heart rate information is given as input in the form of an *ActivityAccessor* objects list. For each activity, a moving average of both the grade adjusted speed and heart rate streams is computed in order to smooth the signals in the attempt to partially reduce noise.
Next, the smoothed HR stream is divided in a number of windows created by setting a threshold variance for the values allowed in each window. This a very important step because, given a properly set threshold, the result will show small windows in correspondence of high heart rate fluctuations.
A this point, the shorter windows are discarded to get rid of unstable measurements and focus on the consistent intervals remaining. Then, for each window n identical samples (where n is the window size) are created by taking the median of the actual values. This is done for both the GAP and the HR stream resulting in n two-dimensional samples for each window.
Once all the activities are processed, an additional filtering is done on the final samples. The ones with speed under a selected minimum value or over a percentile of all the speeds are removed (in general different outlier detection methods could be used at this point). Finally, the value corresponding to an high percentile of the remaining HR points (as shown in Fig. 3.4) is returned as the actual HR_{max} estimate.

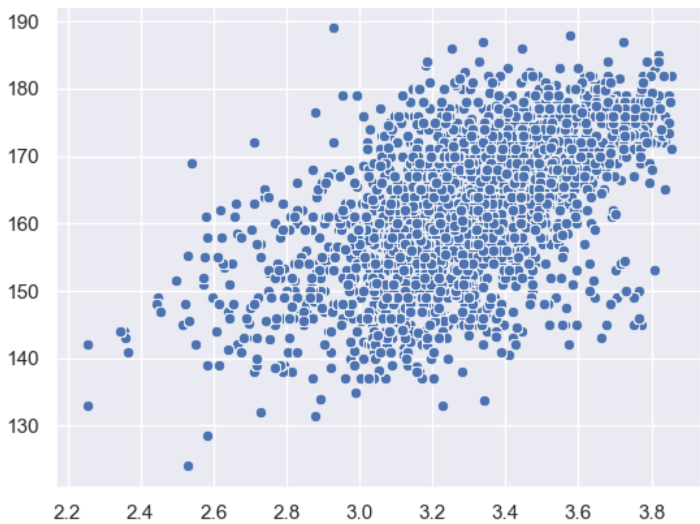


Figure 3.4: Resulting samples after a HR_{max} estimation process. Speed is expressed in m/s, heart rate in bpm.

3.3.2 Anaerobic threshold estimation

The importance of having an estimate of the athlete's anaerobic/lactate threshold (AT) was already emphasized in Section 2.2.2. Particularly, we mentioned that for intensity and load estimation is considered a better predictor with respect to other parameters such as V$\dot{O}$2 max or even HR_{max}.
As we saw, one approach for the estimation of this value that stands out for simplicity and ease of execution is found in the test conceived by Prof. Conconi. In fact, the main advantage of this method comes from the fact that it's based on the conduct of a practical field test as it does not require any laboratory testing, which is ideal especially in our case. However, because in our analysis we needed to work with past running data, we had to come up with a way to apply the same principles introduced in the original work with the absence of a proper incremental test for each athlete.
Instead of trying to understand the relationship between speed and heart rate from a single increasingly difficult effort (such as the real Conconi test), we exploited the availability of multiple workouts data in order to map the relation of these two variables and look for a change as observed by Conconi. Moreover, we used a segmented regression analysis to model and automatically detect the presence of the eventual deflection point in the data. The whole procedure is covered in detail below with the help of some examples and a discussion on the limitations present.

Our approach

The main difficulties we had to overcome when approaching this estimation method were relative to the nature of the data considered during the process. In fact, having to deal with all kind of workout sessions, made us confront with a series of issues that would not normally be present when conducting the original test. Indeed, heart rate response to fatigue involves many variables that contribute to its variation (w.r.t. a controlled incremental test). For this reason, we worked to provide a processing pipeline that tries to address as much as possible these problems, before even searching for the actual deflection observed by Conconi.
The approach followed in our framework for the estimation of AT can be divided in four main phases:

1. Samples extraction from activities with heart rate data
2. Samples cleaning
3. Distribution estimation and new samples generation
4. Deflection point modeling and validation

In the paragraphs below we present each one of these steps together with some motivations behind the choices we made and the issues we encountered along the way (the values used in the final configuration are shown in Table 3.3).

Parameter	Value used	Phase
moving avg. window	30 seconds	n.1
windows size	20 seconds	n.1
minimum window	5 seconds	n.1
speed drop threshold	0.2 m/s	n.1
speed drop interval	30 seconds	n.1
minimum speed (filter)	2 m/s and 5th percentile	n.2
speed intervals sizes	0.2 m/s each	n.3
speed intervals n. samples	100 each	n.3
breakpoint speed and HR boundaries	min 30th, max 95th percentile	n.4
breakpoint first slope angle boundaries	0 < slope angle < 90	n.4

Table 3.3: Chosen values for the parameters used during the estimation of Anaerobic Threshold.

In the first step, a set of two-dimensional samples (speed, hr) is collected from a series of low level activities by using an approach very similar to what we saw in the HR_{max} estimation. The main difference is that here the windows are created with a fixed size since during testing we found out that this configuration worked better compared to the dynamic one (based on variance). In this context, grade adjusted speed is used when available, in order to partially compensate the inclines effect which might skew the values of some samples (for example high HR at low speed during an ascent). In addition to deleting the resulting small windows to remove some fluctuations, in this case we also discard the ones present inside a certain time interval after a significant drop in speed. This is done to avoid ending up with some low speed samples that exhibit high heart rates due to the effort just exerted.

Before moving to the next steps of the process, a cleaning of the samples is done to ensure that no particularly anomalous data point influences the analysis results. First, samples with a speed under a set minimum threshold (not considered running) are deleted. Then, the remaining data is trimmed by considering percentiles to remove extreme speed values. Finally, some additional outliers are identified and filtered by considering the distribution of the ratio between HR and speed.
As a side note, we also experimented but eventually avoided DBSCAN clustering [14] which presented some issues since, as we will mention shortly, the imbalance in speed distribution of our data was causing the removal of some important samples.

After the cleaning procedure, we initially moved directly to the segmented regression analysis in order to detect the Conconi deflection (if present). However, we noticed that many tests where failing due to a severe imbalance in the number of samples at different speed points. In practice, most cases showed a greater concentration of data points towards the middle speeds with respect to the outer zones. Since the deflection point is usually found near the higher paces, this issue was compromising the work of the regression model we used. Clearly this was a problem caused by the fact that most of the activities considered for each test rarely showed speeds constantly greater than the actual anaerobic threshold. Despite that, by plotting a set of heat maps on some speed intervals, we could intuitively detect a deflection on the majority of the cases (as shown in Fig. 3.5). Given that, we decided to include an additional step in our process to compensate for this problem. In particular, after dividing of the original samples in consecutive groups of increasing speed intervals, for each interval we estimate a Gaussian distribution using the real HR values observed. Then, for every interval we generate a fixed identical number of new samples with an arbitrary speed (inside the interval) and an HR value randomly extracted from the relative estimated distribution. This allows us to obtain a balanced sequence of data points covering all the speed domain reached by the activities analyzed. After their generation, these new samples are then passed into the next step of the algorithm in place of the original ones.

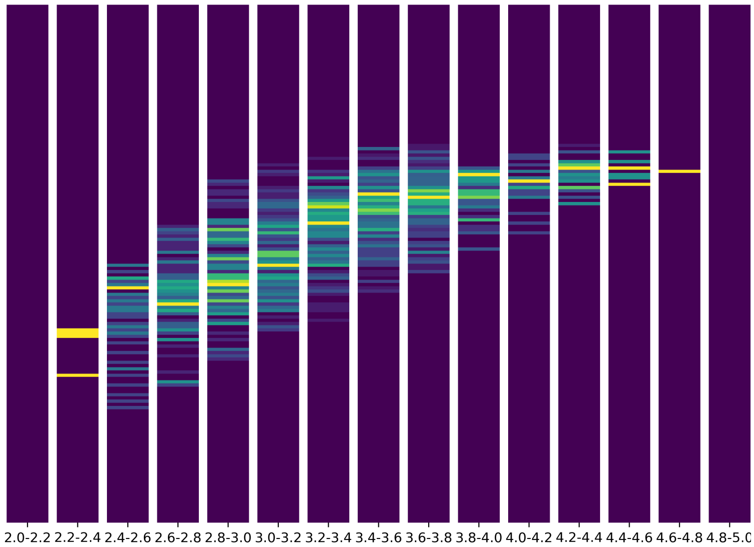


Figure 3.5: Heat maps showing the distribution of samples used for the anaerobic threshold estimation (the interval values of each map are shown at the bottom in m/s).

In the last phase, a two segments piece-wise linear regression model [20] is fit in order to find the point in which the speed/hr relationship changes. This model looks for a breakpoint separating the speed domain in two intervals where an abrupt change in the heart rate increase can be observed. To do so, it tries to fit separately the two segments that make up the segmented line while simultaneously guessing the location of the critical point. Because we are looking for a deflection, the first slope should be positive and greater than the second one. To ensure this, in case an increase in slope is found instead, the model is refitted by removing the points preceding the breakpoint. This can happen sometimes since we have observed that, especially in the low values of speed, there can be an initial positive inflection probably due to noise or other physiological aspects. As a matter of fact, we eventually found out that in some cases more than one slope can be present in this relationship (before the actual one in correspondence with the threshold). For this reason, we also tried fitting models with three separate segments to capture these cases. However, given the inconsistent results and the added complexity, we ultimately decided to keep the two segment models and instead work on the removal of these additional slopes. As a last step, the model found is validated by comparing it to a simple one degree linear regression and by checking some boundaries on the position of the found breakpoint and the resulting slopes. If the conditions are not all satisfied, we cannot conclude that a deflection point exists so a threshold estimation is not made. Otherwise, the speed and heart rate in proximity of the critical break point are labeled as the anaerobic/lactate threshold ones (Fig. 3.6 shows a successful example).

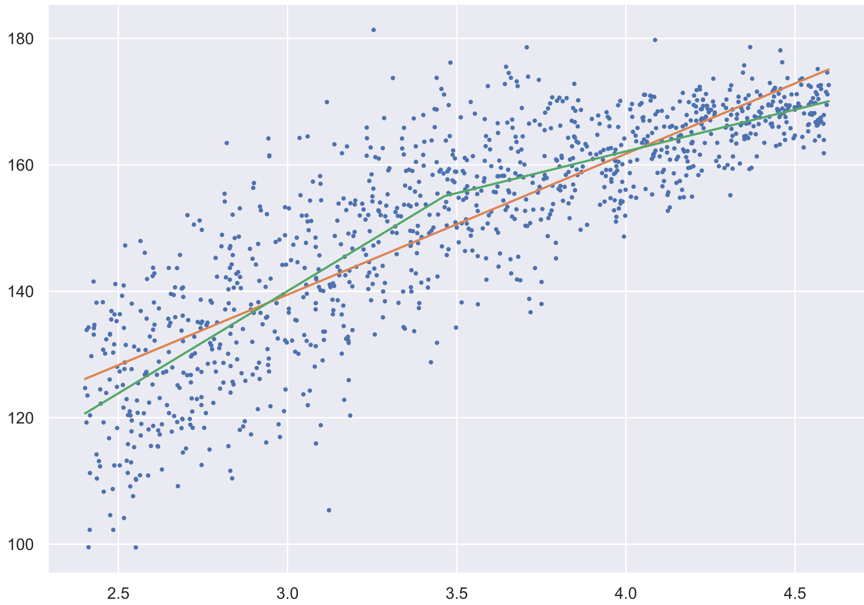


Figure 3.6: Deflection point modeling example for the estimation of anaerobic threshold. The piece-wise linear regression model (green) is presented against a standard one degree linear model (orange).

Limitations and drawbacks

The lack of individuation of a suitable deflection can be the result of different factors, both related to the Conconi test itself and our subsequent implementation, given the data available.

As we anticipated in Section 2.2.2, the approach devised by Conconi presents some problems and limitations that do not always ensure a successful anaerobic threshold estimation. In this regard, a recent study [18] found that a certain percentage of tests do not offer an unambiguous way of finding a deflection point and identified different HR response variants to increased speed in individuals. For this reason, even in perfect conditions and with all the data needed available, the presence of a deflection point in the speed/hr relationship cannot be taken for granted.

In addition to these intrinsic issues, in our implementation we sometimes have to deal with insufficient and/or noisy data that also includes efforts on irregular terrains. These problems can compromise the determination of a correct threshold even when it would be observable in ideal conditions. Furthermore, differently from the original test, to make an estimate we are forced to consider multiple sessions that can cover a time span of various days. Even though we make sure to do not include too wide intervals, it can happen that in the window of activities considered the athlete's anaerobic threshold slightly changes rendering its identification more challenging.

3.3.3 VDOT

The VDOT score, as introduced in Section 2.2.3, is another parameter that we try to compute whenever is possible. It can be a useful piece of information to have when trying to estimate training intensity and is also used to compare multiple athletes based on real performance results.
The default way to obtain this value consists in using a race or all-out effort in order to estimate the percentage of V$\dot{O}$2 max utilized in the session and consequently compute the theoretical maximum. This is accomplished using the two equations provided by Daniels [10], based on the regression lines established with data observed in a number of different athletes. First, the activity duration is used to calculate the percentage V$\dot{O}$2 max that can be technically sustained for the given time. Then, the oxygen cost is estimated considering the velocity maintained during the effort. Finally, the cost is divided by the percentage resulting in the desired VDOT value.
With a collection of past workouts like in our case, however, it's not always possible to count on the presence of races or similar activities. Because of this, we arrive to a VDOT value starting from the knowledge of an athlete's functional threshold pace (ftp) which, as discussed in Section 2.2.2, can be practically approximated with the speed at the point of anaerobic threshold (found with the approach described in the last paragraph). Since ftp can be maintained for about an hour, we simply reuse the two-equations method above by simulating an effort of 60 minutes at threshold speed. This way, as long as we have recent workout data, we can provide an estimation even if we don't know any actual race result.

3.4 High level view

As we anticipated earlier, one main challenge we had to face when designing the processing architecture was finding a way to generate an high level view of our running activities. The framework and solutions adopted will now be presented in detail following a short introduction of the basic concepts.
In order to understand how an high level activity is created, we need to combine the two main framework elements introduced thus far. On one side, we have the *ActivityAccessor* object, which represents the low level interface to the activity and provides the objective measurements in the form of data streams. On the other side, the known athlete's physiological parameters (at the time of the run) are represented by means of an *Athlete* object. By joining these components together, is possible to consider a running activity in a way that takes into account the athlete perspective and, consequently, characterizes all efforts with respect to the runner personal physical condition.
Thanks to this approach, we gain the possibility to perform more advanced data analysis and better understand the real impact of each session on the individual athlete performance. For this reason, failing to capture these aspects, can leave us with an incomplete picture of the actual training effort and is therefore a fundamental step in our processing phase.
The practical implementation of this construct requires the introduction of a new object, that in the proposed architecture is denoted as *HighLevelActivity*. This interface is used whenever we need to access some high level information, without having to be concerned with the underlying low level details of a run. To initialize it, an ActivityAccessor and Athlete are provided during creation in order to allow the subsequent construction of the *HighLevelStream* objects.
These high level streams are the key to access a number of functionalities that will be described in more detail in the following dedicated subsections. Among these features we include: customized intensity zones mapping for speed and heart rate, training load measures and workout classification through activity blocks.

3.4.1 Intensity zones and labeled streams

When we take a look at low level activities, we are presented with a multitude of data that describes quite precisely each session in all its measurable aspects. These streams, however, lack information about the actual relative effort and thus can be of limited use when trying to understand an athlete's training habits.
For example, if we take two runners of substantially different performance levels, an activity ran at the same identical pace could be considered an easy run for one but a threshold run for the other. Similarly, when extrapolated from the context of a single athlete, heart rate values can't assume any particular meaning since the individual genetics, along with other factors, make all comparisons pointless.
Because of these considerations, we needed to find a way to represent some of the relevant streams of our running activities as a sequence of specific efforts, by considering the values observed with respect to some known parameters of the athlete executing it.

The HighLevelStream object

In our specific case, the streams for which we build an high level view are the ones containing speed and heart rate measurements. As we have seen in Section 2.2.3, this data is often used to quantify training intensity since it's really easy to divide it into zones once some athlete's parameters are known. Furthermore, it can give a reasonably good idea of the actual intensity while being relatively accessible in terms of acquisition.
For this purpose, inside the HighLevelActivity object we create an HighLevelStream for both the speed (also grade adjusted if available) and heart rate streams. These resulting objects are noting more than containers holding a sequence of windows created by slicing each stream based on a measure of variance. Every one of these windows includes the original values but can also return some information on that specific interval of the activity, such as duration or distance.
Once the high level streams are generated, is then possible to query the activity object in order to obtain multiple intensity mappings of these items. To do so, the windows inside each HighLevelStream are individually labeled by considering the given zone map and the parameters contained in the Athlete object. In the returned stream all windows display a label that corresponds to the intensity zone reached by the window reference value (usually the median). Additionally, the zone shifts between adjacent windows are computed with reference to the specific zone order and a summary is provided of the time spent in each zone during the whole activity. As an example, some windows of a labeled HighLevelStream are shown in Fig. 3.7.

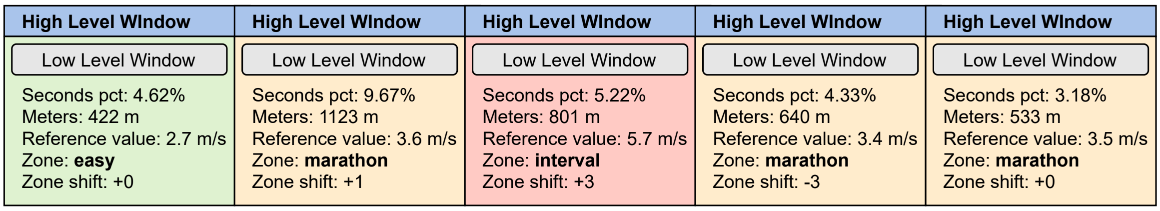


Figure 3.7: High level windows in an labeled HighLevelStream of speed (using Jack Daniels intensity zones).

Using this approach that separates the objective data from the intensity partitioning method, allows us to easily obtain different views of speed and heart rate efforts, according to the zone mapping provided. This can be of real importance, since sometimes a single perspective is not sufficient to fully understand all the aspects of a running workout session. We will also see that the other features available inside the HighLevelActivity object make an heavy use of these high level streams information in order to deliver more interesting representations of the single run.

High level sliding window

An additional tool that we developed in order to work with the high level streams of an activity, is given by the high level sliding window function.
When we consider a labeled HighLevelStream, the sequence of windows inside it can often display multiple short intensity zones variations that result in an oscillation of

labels in some points of the run. These brief changes, however, do not always represent an actual zone shift that is then protracted consistently for a substantial percentage of the activity. Instead, the fluctuations can be the result of noisy measurements or, in general, sudden intensity variations that only span short intervals of time. Since in our architecture we wanted to distinguish between consistent and inconsistent changes in intensity, we developed a method to able to exclusively detect the points in which a true zone change occurs, whilst ignoring eventual false shifts encountered in the process. This exact functionality is embodied by the high level sliding window algorithm, which implements the change points search on a given HighLevelStream object. In addition to this basic feature, this tool also offers the possibility to configure custom zones boundaries and other constraints for the research of particular change points that respect a given set of conditions.

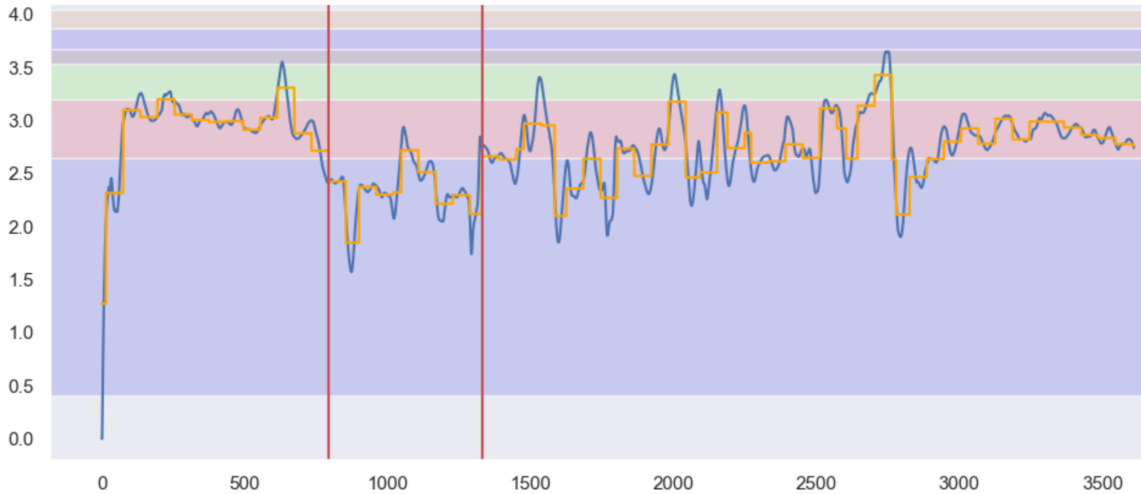


Figure 3.8: Change points detection on a labeled speed stream. The colored bands represent the various intensity zone boundaries (Jack Daniels VDOT zones). The raw stream (blue) along with the windows reference values (orange) is plotted w.r.t activity time. The vertical red lines show the found change points.

The practical implementation of this function includes two different execution modes.
In the "infinite change points" mode the high level stream is analyzed to find an arbitrary number of zone shift points inside the whole activity. Every time a point is found, the windows preceding it are divided from the remaining ones. This results in a final container that exhibits the sequence of sections created by this slicing, where each section is given by the group of high level windows between two change points. Fig. 3.8 shows an example where this is process applied to an activity speed stream.
In the other execution mode instead, the user has to specify the number of points to look for and some conditions that must be respected in the sections before them. If the constrains are violated in this scenario the algorithm fails, meaning that the requested zone shift points could not be found. Otherwise, the resulting outcome is returned in the form of sections, implying that all the demands were met. In practice, this feature is used whenever we are looking for any particular behavior inside an activity.
In any case, the logic behind the detection of a change point is exactly the same in all the execution modes mentioned above. To achieve it, the high level windows of the stream

are processed one by one in an incremental fashion. At the start of a new section (or initially at the start of the stream), a set of zone boundaries is defined. If we are not in the algorithm's infinite points mode, the user boundaries are applied. Otherwise, they are created dynamically by sampling a number of windows that make up a certain time percentage of the activity (lower and upper boundary are both given by the prevalent zone contained in these objects). Every time a new window is added to the processed ones, the composition of some "recent" windows belonging to the current section is inspected. If a certain high percentage of the values inside them is either above or below the actual boundaries, these windows are separated from the previous ones (change point) and a new section is initialized with them. The algorithm then continues its execution repeating the steps above until the end of the stream is reached or, in the case of custom boundaries mode, either all conditions are satisfied or a violation occurs.

Essentially, the whole approach relies on the fact that, in order to trigger a change point, there must be a somewhat consecutive number of efforts strictly below or above the given boundaries. By considering only the recent windows, past fluctuations are ignored resulting in a overall more robust approach that can be easily tuned changing the various percentage parameters, depending on how sensitive we need it to be for the task at hand. It must be noted, however, that the result highly depends on the intensity zone mapping chosen for the high level stream labeling.
Either way, the use of this tool allow us to further analyze the athlete's sequential efforts by capturing some aspects that would be otherwise ignored or overlooked. In particular, all the features of this function are employed in our framework to help with the workout classification procedure that will be presented later in this section.

3.4.2 Activity training load

An important element that we wanted to capture, when considering a running activity, is the stress brought by the workout on the specific athlete.
In Section 2.2.4 we introduced the concept of training load and we saw that multiple methods exist in order to compute the total amount built up during a session. Later, in Section 2.3.3, we also considered some proprietary metrics developed by some of the most famous tracking platforms for this purpose.
In our framework, the HighLevelActivity object includes the possibility to compute a number of different load scores, depending on the data available in the individual run and inside the associated Athlete object. If the activity allows it, several metrics based on both speed and heart rate can be simultaneously used to obtain a more reliable estimate of the actual training load.
While some of the original formulations utilize the average value observed in the whole session, our implementation leverages the HighLevelStream objects capability to provide a more precise result, especially in non-constant runs. For example, if we consider a short interval training activity, it should be immediately clear that the average speed is not a good indicator of the effort. In fact, since also the recovery phases are included in the computation, the resulting average will be rather low. This can make a big difference in the load estimation of some workout types, therefore we decided to use the high level streams to cut the run in smaller windows for the calculation.

As a final note, whenever pace is employed as a measure of training load, the grade adjusted pace is used in place of the real observed speed to account for the eventual inclines present during the session.
The load scores implemented in our framework are shown in Table 3.4.

Name	**Description**	**Formula**
rTSS	TrainingPeaks running training stress score. Based on time spent at speeds relative to the athlete's functional threshold pace.	$\sum_{w=1}^{N} rTSS(w)$ Where each w is one of the N windows composing the grade adjusted high level speed stream. rTSS is computed as in Section 2.3.3 given the reference speed and duration of each w.
edwards	Edwards TRIMP score. Based on time spent in the five different heart rate zones.	$\sum_{z=1}^{5} coef_z \cdot mins_z$ Where the summary of minutes passed in each zone comes from the reference values in the HR stream high level windows.
daniels	Jack Daniels score. Based on time spent at speeds relative to the athlete's VDOT.	$\sum_{w=1}^{N} Daniels(w)$ Where each w is one of the N windows composing the grade adjusted high level speed stream. Daniels assigns a certain amount of points per minute depending on the percentage of VDOT ([9]) given the reference speed and duration of each w.

Table 3.4: Training load scores available inside an high level running activity.

3.4.3 Workout classification

With the tools presented so far, it's possible to generate an high level view of a run that can characterize quite well an athlete's effort, considering a number of different aspects. In our analysis, however, we wanted to gain a more in-depth understanding of the runners habits by also considering the different types of workouts adopted during training.
Unfortunately, activity tracking platforms don't provide this kind of data or, at least, not in a structured and homogeneous way. In fact, the users of these services are usually equipped simply with the ability to comment their activities or assign some general labels to them. For example, Strava only offers four different sub-categories to choose from within the running activity type (Run, Race, Long Run, Workout).
In the case of these platforms, this approach makes sense considering that the main aim of the services is to give the capability to track training activity, hence it's the athlete that should provide this specific information in the first place. In our case, on the other hand, it's impossible to perform an analysis based on this fragmentary data since, by leaving the classification directly to the users, the result might contain incomplete, uninterpretable or erroneous knowledge.
For this reason, we couldn't count on a precise description of the workouts coming directly from the source. Therefore, in order to overcome this problem, we thought it would be interesting to attempt a classification of the available runs based entirely on the activity

data provided as input to our framework.
In practice, this proved to be quite a difficult challenge, given that a number of problems arise when tackling this approach in an unsupervised way, as we will discuss shortly. Perhaps, this is also a reason why the tracking platforms delegate the task to the users without trying to automatically produce a classification for the provided runs.

Developing a classification approach

Given that we can already count on a set of available tools inside an high level activity, a first naive approach would be to leverage the summary of time spent in each intensity zone (for the relevant high level streams) in order to perform a rough classification of a specific activity. For instance, if we consider the labeled speed stream of a run, some boundaries can be indicatively set on the existing zones to detect if the session was mostly executed at some speeds that can be considered specific for a certain workout class (e.g. threshold run, recovery run, etc.).
Unfortunately, this solution works only if the activity exhibits a constant effort, since it does not account for any sequential aspects characterizing the workout. This is particularly evident if we take in consideration an interval training session where maybe most of the time is given by the recovery portions between each repetition. In this case, the zones summary does not give much information on the workout category itself, thus it's not really a feasible approach for this kind of activities.
In Section 2.2.5 we presented a categorization that divides the possible workouts in a number of definite classes. In reality, however, we need to understand that an athlete does not necessarily follow this division and can customize the actual sessions in many different ways. Furthermore, it should not be ignored the fact that each training could be preceded and followed by a warm-up and cool-down phase respectively, which can interfere with the classification task. All these issues render the act of classifying an activity in its entirety an ill-posed problem.
Because of these considerations, we decided to instead follow an approach that first tries to break down a running activity in a sequence of "blocks" representing the various phases making up the training session as a whole (e.g. warm-up, intervals, etc.). These blocks are identified by employing some heuristics that look for a set of characterizing elements contained in the run data. Once divided into these components, is then possible (if needed) to perform a classification that considers the blocks characteristics and their position.
This modular approach has the advantage of supporting all kinds of workouts, without the need to forcefully include each one of them in some predefined class if not required. Moreover, it offers the possibility to analyze in more detail the activities by creating an high level representation of its internal components (e.g. checking the presence of warm-up in a collection of runs).

Heuristics for blocks individuation

During the execution of a running activity, pace is the main parameter that an athlete can control in order to determine and modify the type of workout performed. As a consequence, by observing the evolution of this variable in time, is often possible to notice some patterns that can reveal the nature of the training session from a classification point of view.

In an effort to decompose a run in several meaningful blocks, we identified a few characterizing aspects that can be usually located by analyzing the high level speed stream of a given activity. These components can provide us with the ability to distinguish the phases constituting all kinds of possible workouts, thus allowing us to perform the division mentioned earlier.
To pinpoint these elements, we make use of a number of functions that check their existence by utilizing a series of custom defined heuristics on either the whole workout or some parts of it.
In particular, we try to locate the potential presence of the following patterns:

- Intensity peaks sequences (intervals)
- Warm-up phases
- Cool-down phases
- Pace zone changes

The advantage of basing this search on the speed alone is that, theoretically, there is no need for heart rate data in our classification method (if we already know the athlete's parameters). Additionally, this approach avoids the problems of dealing with cardiac lag, which can really impair the detection of quick intensity changes due the intrinsic nature of the heart rate responses.
A disadvantage of this procedure, however, resides in the fact that often the run is not performed on a level surface. Because of this, the actual observed speed can be influenced by the terrain changes and consequently render the block individuation more challenging. In fact, even when using grade adjusted pace, we ran in some issues (both coming from the adjusted estimation and our method) that sometimes impair the correct determination of the searched features. For this reason, in our analysis hilly activities might present more problems that their flat counterpart.
As a final note, it must be also mentioned that in all the following we assume that the activity segments containing manual pauses are not considered part of the workout and hence are skipped.
In the next paragraphs we will now explain in detail the logic behind the detection of all the elements involved in the activity blocks creation.

Intensity peaks sequences represent abrupt changes in speed that result in several repeated higher intensity efforts (peaks) alternated with recovery time periods in-between. These components are the building blocks of any interval/repeats workout and, consequently, constitute a fundamental pattern for a successful block extraction.
In order to detect their presence and divide an activity accordingly, we perform a number of processing steps that, starting from the entire high level speed stream, look for the existence of one or more peak sequences. These phases can be summarized in four main stages:

1. Peaks detection from the reference high level speed stream signal
2. Mapping of found peaks and recovery intervals to high level windows

3. Research of peaks sequences and division of the activity in sections of high level windows

4. Check of conditions and eventual elimination of erroneous peaks sequence sections.

In the first step, we check the speed stream for the presence of any signal peaks with the help of Scipy's *find_peaks* function [1]. Since the peaks we are looking for must respect a series of conditions in order to be considered part of some interval workouts, we adopt some measures to filter out the eventual false positive generated by the search. Rather than providing the base speed stream as input to the function, we exploit the reference stream (given by all the reference values contained in the high level windows) to create a step function representing the whole activity. This has the effect of ignoring most short or single value peaks while preserving the larger ones. Additionally, we make use of the find_peaks parameters by specifying a minimum and maximum peak width along with the minimal required prominence [2] for the detection of the interesting peaks (example in Fig.3.9). Still, it can happen that some false peaks are identified during this phase because, for example, the athlete suddenly decreased speed creating a downward spike that influences the adjacent values. This usually can give rise to a number of problems, but we will see shortly that for most of them we adopted several filtering solutions in the last step of the procedure listed above. At the end of this stage, the properties of the found peaks are returned by the Scipy function. If either none or a single peak is discovered the whole process ends, since we did not find a peak sequence. Otherwise, the information acquired until now is passed into the next phase.

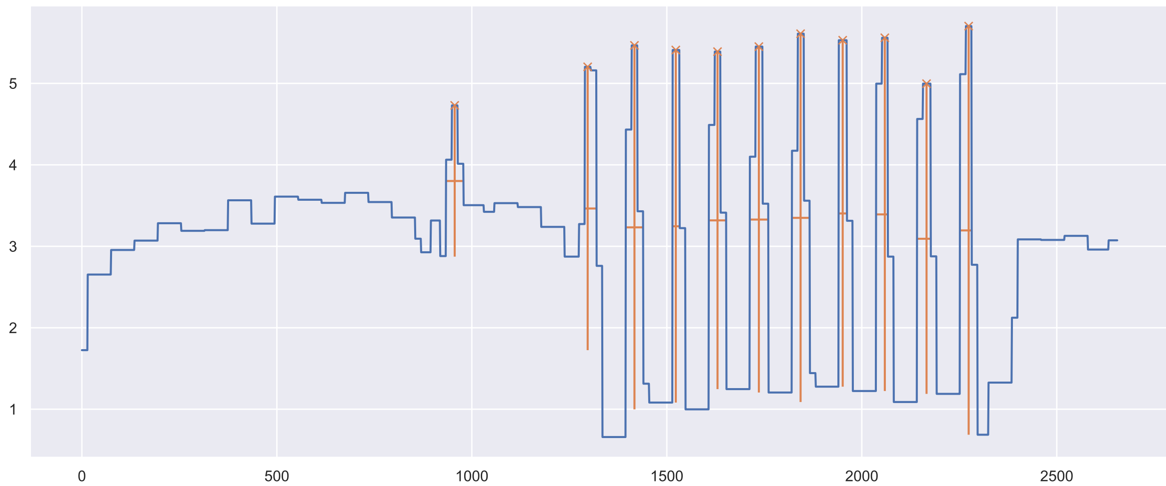


Figure 3.9: Peak detection example on the reference values speed stream of a running activity. Each peak found by the Scipy function is highlighted with an orange cross.

The second step of the operation is used to map the peaks and recovery intervals identified on the time stream into the corresponding high level windows. To do this, each window contained in the stream is given one of the following labels: "peak", "recovery", "pre-sequence", "post-sequence". For every peak, we extract the start and end timestamps

[1] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find_peaks.html`
[2] `https://en.wikipedia.org/wiki/Topographic_prominence`

(boundaries) from the properties within the Scipy function result. All the high level windows contained partially or totally inside the boundaries of a found peak are assigned the "peak" label. Conversely, to those that are located between each pair of peaks is given the "recovery" label. Finally, all the windows preceding the first peak or following the last one are labeled as "pre-sequence" and "post-sequence" respectively. The final result is a collection of lists where for each found peak (and recovery in-between) we have the corresponding windows along with the ones before and after the very first and last repeat of the session.
The third step involves the activity division in windows sections to delimit the peak sequences identified from the rest of the run. Since in our task we are not looking for multiple detached single peaks, but instead sequences of them, there is also a need to check the distance between each spike and eventually exclude the isolated ones while aggregating one ore multiple groups of closely spaced peaks. Because of this, we set a maximum length in percentage of the session time, after which the recovery is considered too big and a sequence is consequently split. A the start of this phase, pre and post sequence windows are separated in their own two sections at the activity edges. Next, peak and recovery windows are processed in an alternate fashion following the session order. If a recovery interval between two peaks is too wide, a new section containing it is created to signify a separation of the single peak sequence in two different ones. This process can generate some isolated peaks that are accordingly merged to the non peak sequence sections. When the procedure terminates, we end up with a container of ordered sections that are either labeled as peak sequences or not. At this point, we check if at least one sequence is found, otherwise we state that no peaks sequences exist.
The fourth and last step is only executed when at least a peaks sequence section is found inside the activity. In this case we need to make sure that the results found are appropriate given our definition of intervals/repeats in the context of a run. In particular, two conditions are checked to ensure that the erroneous intervals sections are reduced to a minimum: total repeats length (in time) w.r.t. the whole activity and intensity zone increase during repeats. The first one intuitively sets a minimum threshold that must be passed so that the peaks sequences can be considered a relevant part of the session. The second one instead confronts the intensity zones of the peaks/non-peaks sections to verify that there was a consistent increment in effort between the two. For example, speed oscillations and peaks can also happen due regular terrain changes (e.g. hills laps) which create the typical repeats shape but do not exhibit a significant variation in intensity. For this reason, peaks sequences that do not positively change zone are excluded and eliminated. As a consequence, an activity must pass both these checks to be considered a run that contains sections of repeats.

Warm-up and cool-down phases can be often found inside many high intensity and interval workouts (but in general all kind of workouts). Their essential purpose is to allow the athlete to perform a gradual transition to and from a challenging bout of effort. For this reason, we can identify them as relatively short low intensity time periods that are located at the start (warm-up) and at the end (cool-down) of a running activity.
Aside from the intrinsic value of their discovery, detecting their presence inside a training session can generally simplify the extraction of other types of blocks and generate an overall better classification result. Consequently, we always make an effort to spot this

kind of behavior whenever we are dealing with a given activity.
In an attempt to extract these patterns, we look for areas of low intensity speed at the edges of a run that are separated from the central portion by a significant change of intensity zone. This task can be accomplished by exploiting the already introduced high level sliding window function in its custom boundaries mode. Thanks to this approach, we can specify the properties that our warm-up or cool-down segment should respect and then search for a change point separating it from the rest of the activity.
In the warm-up case, first we decide what intensity zones of the current mapping can be considered acceptable as low intensity (for our purpose). Then, we set a minimum and maximum length (in percentage of the activity) for the segment itself. Finally, we provide as input to the function the labeled high level speed stream of the activity. All these parameters result in the search for a single change point that must be preceded by an area complying with the given specifications. If the boundaries are not respected or no change point is found, the algorithm signals that no warm-up is present. Otherwise, the high level windows before the change point are separated from the rest of the run to display the actual activity partition.

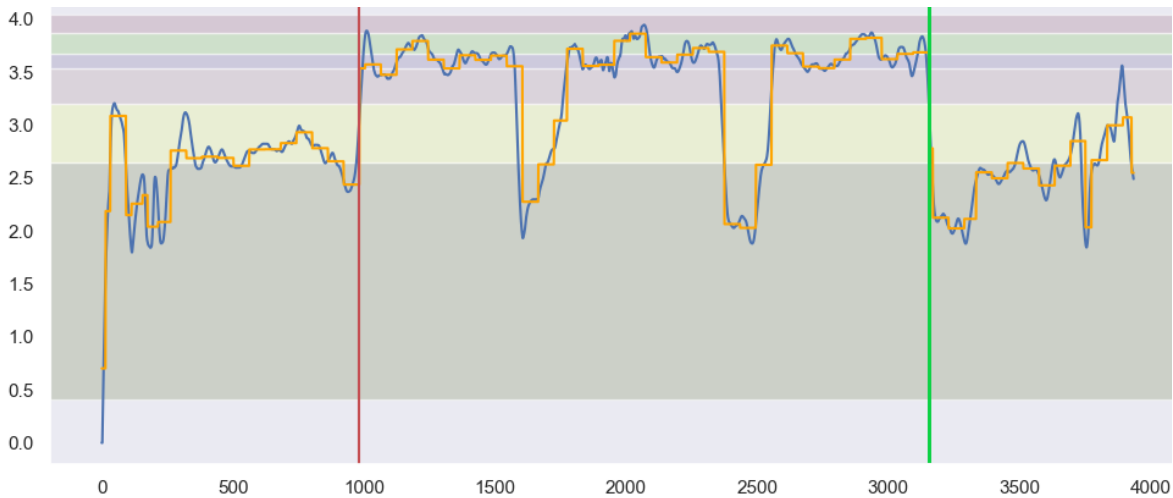


Figure 3.10: Warm-up and cool-down individuation inside an interval workout. The two vertical lines represent the found transition points that separate warm-up (red) and cool-down (green) from the rest of the activity.

The logic behind cool-down detection is almost identical to what we already considered in the previous case. In fact, the only difference is that, in this instance, the sequence of high level windows composing the stream are reversed before being given as input to the sliding window function. This allows us to reuse the same kind of change point search simply starting from the end of the activity instead of the beginning. Obviously, also in this case there can be either a failure or a success depending on the presence of a suitable cool-down phase. Consequently, when a positive result is returned, the windows order is restored before the activity is partitioned in the two resulting sections.
During both warm-up and cool-down identification, sometimes it can happen that a true pace change that, however, is not really prolonged in time (e.g. peak) accidentally triggers a false detection. Because of this, in the case of positive individuation we always make sure that the majority of the remaining activity (after warm-up or before cool-down) is

of an increased intensity, otherwise a negative result is returned instead.
As a visual example of the process described so far, in the high level speed stream portrayed in Fig. 3.10 the detection of both warm-up and cool-down is shown when considering the second zone from the bottom as upper boundary.

Pace zone changes and their absence constitute the last characterizing element that we take in consideration with the objective of building the block view of a workout.
Until now, we have dealt with some activity patterns (e.g. peaks) that can be quite easily spotted when looking at the speed stream of a session. However, for a correct identification of all workouts components, we also need to be able to detect more subtle pace changes that cannot be considered equivalent to a constant effort. This is mainly done to distinguish the sections of a run executed at uniform pace from the ones with different kind of pace changes (e.g. progression). Moreover, it allows us to cover all the parts of an activity that do not consist of the other elements discussed already.
In order to perform this search, we once again take advantage of the features available in the high level sliding window function. Only this time, instead of specifying particular boundaries, we utilize the "infinite change points" mode, since the number of zone changes in this case is unknown. The high level speed stream of the activity (or a part of it) is hence given as input to the function, which returns one or more sections depending on the change points located.
If a single section is returned, we assume that it was a uniform pace effort (no change points detected). Otherwise, we analyze some characteristics of all the resulting sections, such as their composition and size, to check if some actual consistent intensity changes are present or not.

Blocks extraction from activity

The heuristics presented so far are at the base of our activity decomposition process. For the actual block extraction procedure, however, we needed to devise a method able to turn the complete running activity into an ordered sequence of *Block* objects representing the workout phases contained in the individual training session. In this context, we created an algorithm capable of correctly employing all the available heuristics in order to identify the characteristic patterns and manage their proper extraction, following a precise logic that starts from the high level grade adjusted speed stream of a given run.
The algorithm is composed of a series of functions that call each other in a predefined sequential manner. Each function, given a part of the activity, is responsible for the individuation and creation of a certain type of Block object. Depending on what was found, it then delegates the rest of its remaining windows to some other functions, waiting for their results in order to merge them. To do this, all the sections that are not yet blocks are given as input to the following procedures. Eventually, a catch-all function returns a single Block without making any further call.
This process produces a gradual transformation of different activity sections into Block objects, where each one contains the corresponding sequence of high level windows. When no unassigned windows are left, the result is merged to create a container only made of the final blocks constituting the whole activity. A general sketch of the intuitive idea behind the block extraction process is shown in Fig. 3.11.

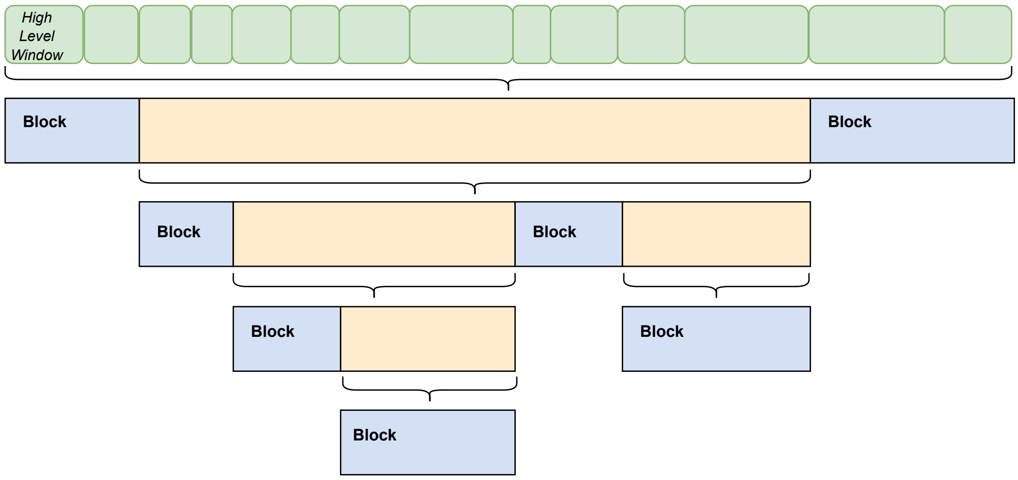


Figure 3.11: General process of block extraction from a running activity.

All the possible types of resulting blocks are presented in Table 3.5 along with a brief explanation of what they represent and the function that creates them.

Starting from the high level stream, the order adopted by the extraction procedure involves the following phases: first the individuation of peak sequences, then warm-up and cool-down search, finally the creation of uniform or non-uniform blocks.
The reason why it's done this way, is to ensure that no block extraction interferes with the other ones during the entire algorithm execution. For example, we have observed that by separating warm-up or cool-down first instead, the function for peak detection has some problems finding the first and last peaks due to the windows cut off by the removal of those two patterns. Because of this, we first extract the eventual peak sequences and only after that all the remaining blocks are created.
A problem of this approach, however, is that sometimes warm-up and cool-down must be identified inside a partial activity which can lack the change point we are looking for in the original heuristic. This is easily solved by also testing the whole section as a potential warm-up or cool-down (if the pace change is absent) with a check on the boundaries that are normally used for the search of these patterns.
The logic of all the functions involved in the process is presented by the pseudo-code in the next few pages.

Block Name	**Description**	**Creator Function**
PeaksSequence	Represents a sequence of higher intensity interval efforts with recovery phases in-between. It's possible to find multiple objects of this type inside a single activity (different interval phases).	extract_peaks_sequences()
Warmup	Represents the warm-up phase at the start of an activity. Can be found at most once and is always the first block in the resulting sequence (if present).	extract_warmup()
CoolDown	Represents the cool-down phase at the end of an activity. Can be found at most once and is always the last block in the resulting sequence (if present).	extract_cool_down()
UniformBlock	Represents a phase where no significant pace intensity zone changes are present. Can be found zero or multiple times inside a single activity.	extract_standard_block()
NonUniformBlock	Represents a phase where there are one or more changes in pace intensity (e.g. progression). Can be found zero or multiple times inside a single activity.	extract_standard_block()

Table 3.5: Types of Block objects that an activity can be composed of.

Algorithm 2: Block extraction procedure

```
function extract_blocks(high_level_activity):
    /* starting point of the whole extraction process */
    get the high level stream of grade adjusted speed inside the high_level_activity;
    call the extract_peaks_sequences func. on the labeled h.l. stream;
    return the flattened sequence given as result of the previous call;
end

function extract_peaks_sequences(high_level_windows :list):
    use heuristic to check for peak sequences presence;
    if present then
        create PeakSequence objects for each section with a peak sequence;
        for the other sections check position and act accordingly;
        if first section of activity then
            call extract_warmup func. on this section;
        else if last section of activity then
            call extract_cool_down func. on this section;
        else
            call extract_standard_block func. on this section;
        end
        return all the resulting sections (in order);
    else
        call the extract_warmup func. on the high_level_windows (whole activity);
        return the result of the previous call;
    end
end

function extract_warmup(high_level_windows :list):
    use heuristic to check if warm-up is present (via change point);
    if present create a Warmup object with the first section;
    if it's whole activity then
        call the extract_cool_down func.;
        return the result of the previous call (preceded by Warmup if present);
    else
        if warm-up not present check if whole section can be considered it and
          create the Warmup object if True;
        call extract_standard_block func. on all the sections that are not block
          objects;
        return all the resulting sections (in order);
end
```

Algorithm 3: Block extraction procedure (continue)

```
function extract_cool_down(high_level_windows :list):
    use heuristic to check if cool-down is present (via change point);
    if present create a CoolDown object with the last section;
    if cool-down not present and not whole activity then
        check if whole section can be considered it and create the CoolDown
          object if True;
    end
    call extract_standard_block func. on all the sections that are not block
      objects;
    return all the resulting sections (in order);
end
function extract_standard_block(high_level_windows :list):
    use heuristic to check if there are pace zone changes;
    compute prevalent zone for each resulting section;
    if no changes or all prevalent zones equal then
        return a UniformBlock object;
    else
        return a NonUniformBlock object;
    end
end
```

Base workout classification

As anticipated earlier, once a running activity is divided in the block view we have presented so far, is then possible to utilize this information (along with the other high level aspects of the run) to perform any kind of classification partition needed for the specific task at hand.
For analysis purposes, we came up with a number of workout classes (based on the ones introduced in Section 2.2.5) that, in our opinion, cover the majority of the most commonly adopted workout types among runners. In particular, these mutually exclusive classes can be summarized with the following labels: *race*, *base*, *long*, *recovery*, *threshold*, *anaerobic*, *interval*, *surges/strides*, *progression*, *fartlek*.

Of all these classes, races are the most straightforward ones to discriminate since we simply trust the runner labeling and consequently assign to this class all the runs that are marked as *race*. In any other case, we start checking the blocks distribution to initially divide the uniform effort runs from the others.
If only UniformBlock or Warmup and CoolDown objects are found (so it's uniform), we need to determine if the session was either a base, long, recovery, threshold or anaerobic run. To do this, both the grade adjusted speed and heart rate (if present) labeled high level streams are considered to verify if the majority of time was passed around, below or above the personal lactate threshold. Consequently, to the efforts located around or above this parameter the *threshold* and *anaerobic* labels are assigned respectively.
Conversely, in the case of a distribution mainly concerning the intensity below the anaer-

obic threshold, we differentiate base, long and recovery runs by considering their duration and distance with respect to the average ones of the relatively recent past workouts. In this context, the *recovery* label is associated to the ones significantly shorter than usual (<=0.5 times), while the *long* label is given to the ones that are considerably longer than the norm (>=1.5 times). In any other case the *base* label is assigned by default.
When the running activity instead presents some PeaksSequence objects, we know that some kind of interval efforts were present during the session. In this instance, the total time (w.r.t. whole run) passed inside repetitions intervals is weighted against the rest of the activity (warm-up and cool-down excluded). If the majority was given by the peaks sections we label it as *interval*, otherwise as *surges/strides*.
Lastly, if all the other conditions fail, we end up with a workout that clearly exhibits some intensity changes (not comparable to intervals) that surely result in a NonUniformBlock object. If in this case the changes create an effort of increasing intensity (composed by at least two sections) and appropriate duration (minimum 10% of the run if last section, 20% otherwise), the run is labeled as *progression*. Alternatively, the *fartlek* label is used on all the remaining activities.

3.5 Activities calendar

Up until now, we have only focused our attention in the domain of a single running activity, ignoring its relation with time or eventual other activities of the same athlete. In our analysis, however, we also wanted to understand the runners training habits by considering short and long term periods of time including several workout sessions.
In order to do this, we need to take into account the fact that some physiological parameters of our athletes can change over the time-span considered. This is why, earlier, we introduced the concept of assigning a different Athlete object to each activity, representing the athlete's status in the moment of the actual effort. If in the previous sections we took for granted the correct computation of these objects that are assigned to each high level activity, now we must spend some time to discuss the issues that arise when trying to implement the parameters estimation procedures in the real domain.
For each athlete considered in our study, we typically collect a bunch of activities that are included inside a precise date interval that can span several months. For this reason, aside from storing the actual JSON files of each run, we utilize a database in order to map each activity with the corresponding date in which it was performed. Following this approach allows us to easily create a sort of calendar that contains the history of past workout activities, thus empowering us with the ability to associate a temporal meaning to every run examined during the analysis process. This data, however, is not limited to running workouts alone. As we will see, some minimal information is also collected on the training sessions that make up other kind of sports (cross-training).
In the context of this input data, we developed an architecture to be able to correctly manage the creation of each HighLevelActivity object and, consequently, perform other analysis tasks on the collection of available activities. The main element of this framework is given by the *AthleteCalendar* object, which is responsible for a series of important steps required in this context. Particularly, given a database of activities and the corresponding files for each run, it provides the following features:

- Estimation of the athlete's physiological parameters and their evolution in the time-frame considered (using the available runs)
- Creation of the HighLevelActivity objects for each run in the database (with the relative Athlete objects)
- Possibility to divide the calendar in multiple windows composed of a fixed number of days to compute some statistics
- Calculation of training load balance parameters

With the subsections below, we will now present the details of all the single aspects characterizing the implementation of these capabilities.

3.5.1 Athlete parameters evolution and estimation

As we have stated earlier, some of the physiological parameters used for the assessment of training intensity can change over time, provided that we consider a large enough interval. Because of this, it's important that we track those changes in order to correctly

characterize all the running activities examined in our analysis.
The initialization phase of the *AthleteCalendar* object conveniently includes a step where we make sure to have the correct estimate of a number of these parameters. If we don't have access to this kind of information, a dedicated process is called and the results are then saved in a file for future use. In particular, to be able to create the needed high level activities, we focus on the estimation methods of two variables that we have presented earlier in Section 3.3: HR_{max} (maximum heart rate) and AT (anaerobic threshold).

Maximum heart rate

For the purpose of our study, HR_{max} can be considered as a fixed parameter, since the time period taken into account does not include several years of activities. In fact, we have seen that this value generally decreases as the age of the athlete advances, so treating it like a constant shouldn't make much difference given the relatively short scope of our analysis.
To estimate it, we simply provide as input to the algorithm described in Section 3.3.1 all the available runs with heart rate data that are contained in the calendar database. Since, in our case, we perform all the processing after the entire data collection is completed, we usually have access to a decent number of runs with HR measurements for this part of the estimation. However, this same exact method can be also used in the case of a dynamic analysis (live after every new activity), just by making sure that enough suitable past data is present in the database.
When a final estimation of the HR_{max} values is available, we save it in the dedicated file for the current athlete so that it will be possible to utilize it later in the high level activities creation.

Anaerobic threshold

Contrary to maximum heart rate, we have already seen that the anaerobic threshold value for a single athlete can change in a reasonably short amount of time when following a long-term training plan. In general, as the performance increases due to an appropriate workout regime, it's often possible to observe a gain in threshold speed that reflects the runner's improved preparation. In contrast, decreasing or impairing training effectiveness leads to the deterioration of this parameter over time.
Having said that, in this context it would not make any sense to adopt the same strategy used for HR_{max} where more data is favored over less. As a matter of fact, utilizing a too wide date range of activities could only render more difficult the threshold individuation process (as seen in Section 3.3.2) since the various deflection points would be overlapping. For this reason, we opted for a windowed approach that, instead, computes multiple estimations by considering several groups of consecutive runs over the course of the time period examined.
Ideally, we should perform a new estimate for every running activity in the calendar that includes all the workouts close to it. In practice, however, this could be computationally intensive in some instances and, in reality, it wouldn't make much difference since many runs would basically have the same threshold value. Because of these considerations, we divide the calendar in a series of fixed activity windows, whose estimations are then used when assigning the values to each run in a way that we will see later.

In order to create the slicing of the calendar in windows, we initially considered using the activities dates to generate one month windows with a step of two weeks to better account for the transition periods between two consecutive estimations. However, we quickly found out that this approach presents some problems, mostly related to the fact that it doesn't ensure that a certain amount of suitable data is collected in each window. This often results in threshold estimations that lack sufficient data points due to a low number of activities with heart rate measurements inside some windows.
To solve this issue, we decided to instead base the window slicing on the cumulative hours of heart rate data present among the running activities in the calendar (ordered by date). This way, we make sure that each window contains at least a fixed amount of HR data, thus allowing a better overall estimate in every instance. Moreover, also in this case we use a step (now in hours of HR activity) after which a window is recomputed so that we end up with a more fine grained sequence of estimations.
To achieve this, the calendar is sequentially sliced in n step-sized windows that contain at least the number of HR hours that is required by each step (except for the last one). Since a final window must reach a certain amount of HR hours that is a multiple of the step ones, a sliding window is used to group the steps into a series of window-sized windows to guarantee the minimal final window length (an example of this division can be seen in Fig. 3.12). If this process is performed in a dynamic fashion, is possible to use this same logic by waiting every time for a step amount of new data to be collected before creating a new window that uses this data together with the past steps (to reach a window size).

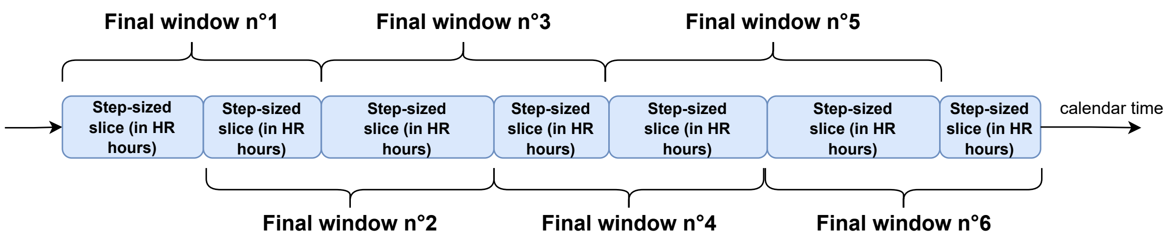


Figure 3.12: Slicing of the calendar w.r.t. cumulative HR hours. In this case a final window is composed by two steps (each step is represented with a different width because it can include a diverse number of activities and dates to reach step hours).

One disadvantage of this approach is that the runs inside a single window span different periods of time depending on how often the athlete trained. This aspect can result in some wide windows (if some intervals with a low number of workouts are present in the calendar) that can still in part exhibit the overlapping threshold issue mentioned before. However, this problem is partially mitigated by imposing a maximum threshold (in terms of days) after which the window is considered too big and consequently discarded. Moreover, as we will se shortly during the values assignments, some additional measures are employed to prevent the influence of a single erroneous value on the other estimates.
When the slicing of activities is completed, we simply use the algorithm for AT estimation introduced in Section 3.3.2 to compute all the thresholds using the runs with HR data in each window. The resulting values are then saved in the appropriate file (like with HR_{max}), only this time we obtain a list of values along with the specification of windows and step size used for the computation.

3.5.2 High level activities creation

In order to perform more advanced tasks on the athlete calendar such as statistics or load balancing calculations, all the running activities in the database need to have a reference to their respective HighLevelActivity objects. As a consequence, when an AthleteCaledar is first initialized, this mapping is established by following a procedure that creates each required object taking into account the athlete's parameters in time.

We already know that for the successful generation of an high level activity two particular elements need to be provided: the ActivityAccessor and Athlete objects. For this reason, in the initialization phase of the AthleteCalendar, both these aspects are taken in consideration and consequently managed by a specific function that, as a result, returns the aforementioned mapping.

For each run in the calendar, the ActivityAccessor is easily retrieved by recovering all the relevant JSON files and providing them as input to each accessor object. This is quite a simple operation, therefore no particular problems need to be discussed in this regards. The creation of each single Athlete object, instead, involves a few more steps, thus we will now spend some words on its relative implementation details.

In the first step of this process, we make sure that the athlete's parameters have already been estimated by checking the relative file where they are usually stored. If this is not the case, the whole procedure presented in Section 3.5.1 is called before moving on with the rest of the operation. Once all parameters are readily available, an empty Athlete object is created for each running activity in the calendar.

At this point, two different types of parameters need to be correctly assigned to each relevant object. First, the time-invariant ones (e.g HR_{max}) along with some basic information on the athlete (e.g. sex) are provided as input to all the objects considered (no one excluded). This makes sense since they should be equal by definition for all the runs examined in the chosen time period. Next, the various values of the time-variant parameters must be attached to the corresponding Athlete objects considering the logic of their computation.

Anaerobic threshold assignments

For the correct assignment of all the threshold estimated on the calendar, both the values and the performed slicing specifications need to be retrieved from the relative parameter file.

Since the estimations were made on a series of windows, each value is referred to a specific group of running activities, that in this case is given by the sequence of step-sized slices composing the entire calendar.

In order to preserve the possibility to perform this computation dynamically (not a posteriori), each step-sized window can only use the values computed in the past. This means that all the Athlete objects of the runs in a step slice receive as threshold estimation the one calculated on the previous final window. However, considering that sometimes the algorithm used for the threshold search can fail, in practice we actually take the median of a number of successful past estimates instead of relying on the single precedent one. This allows us to avoid placing too much weight on a potentially wrong value that might otherwise skew the results.

Having said that, it can still happen that, for some reason, a few Athlete objects are not able to get a threshold value estimation. For example, this is always the case for the first slices in the calendar that do not have a previous final window. In such circumstances, we create the HighLevelActivity anyways by assigning to the relative runs an incomplete athlete representation.

3.5.3 Calendar windows and statistics

After the AthleteCalendar initialization and the consequent creation of all the HighLevelActivity objects, we end up with an handy mapping that links the time-related information of each activity in the calendar to its appropriate high level representation (except for cross-training activities).
This framework allows us to potentially perform all kind of analysis tasks that can exploit all the features we introduced so far when presenting the single running activity architecture. In particular, being interested in the runners training habits over some definite time periods, we implemented the capability to divide the calendar of a given athlete in a collection of fixed length (in days) windows in order to compute some statistics on them. This feature is provided through a dedicated function present inside the AthleteCalendar object which only requires two input parameters (one of them is optional). By specifying the length in days of the desired windows and, optionally, a precise start date, the whole calendar is partitioned into a sequence of windows starting from the given date (if not specified, from the first day of the calendar). All resulting windows refer to their respective date interval and can include both running and non-running activities. In some cases, however, if no workouts were registered in a particular range of dates, the relative windows can also turn out to be empty.
In order to provide a consistent structure every time a calendar slicing is executed, the whole operation returns a list of *CalendarWindow* objects, where each one of them is a container that represents a specific window by giving access to its activities and some information on the slice itself.
Given a single CalendarWindow, we then created a collection of statistics that can be applied to this type of objects with the purpose of delivering some interesting insights on the low and high level characteristics of the workouts contained in it (all these functions only require a calendar window object to work).
In tables 3.6, 3.7, 3.8, 3.9 and 3.10 we list all the functions used during our analysis.

Function Name	Description
n_of_training_days()	Returns the number of days in the window that have at least one type of training (run or cross-training).
n_of_days_with_runs()	Returns the number of days in the window that have at least one running workout.
n_of_rest_days()	Returns the number of days in the window that have no workouts (neither runs nor cross-training).
avg_daily_runs()	Returns the average runs per day. Only days with runs are included.
max_runs_per_day()	Returns the maximum runs observed in a single day considering all days of the window.
n_of_running_ workouts()	Returns the number of runs in the window.
n_of_cross_training_ workouts()	Returns the number of cross-training workouts in the window.
n_of_flat_runs()	Returns the number of runs in the window that can be considered flat.

Table 3.6: Window statistic functions concerned with temporal training aspects.

Function Name	**Description**
running_time()	Returns the total seconds of running activities in the window.
cross_training_time()	Returns the total seconds of cross training activities in the window.
running_volume()	Returns the total meters covered considering all the runs in the window.
elevation_gained()	Returns the total elevation gained (in meters) considering all the runs in the window. Treadmill runs are excluded since have 0 elevation gain by definition (no way to know incline if present).
avg_run_duration()	Returns the average duration (in seconds) of a run considering all the runs in the window.
avg_run_volume()	Returns the average volume (in meters) of a run considering all the runs in the window.
avg_speed()	Returns the average speed (in m/s) considering the time of all the runs in the window.
avg_grade_adjusted_speed()	Returns the average grade adjusted speed (in m/s) considering the time of all the runs in the window.
avg_heart_rate()	Returns the average bpm considering the time of all the runs in the window. Runs without heart rate are excluded.
avg_altitude()	Returns the average altitude (in meters) considering the time of all the runs in the window. Treadmill runs are excluded.
avg_elevation_gain()	Returns the average elevation gained (in meters) considering all the runs in the window. Treadmill runs are excluded.
avg_incline_grade()	Returns the average incline grade considering the time of all the runs in the window. Treadmill runs are excluded because of lack of incline information.

Table 3.7: Window statistic functions concerned with low level runs characteristics.

Function Name	**Description**
n_of_low_intensity_runs()	Returns the number of runs in the window that can be considered of low intensity. Only the runs with a lactate threshold estimation are considered.
n_of_medium_intensity_runs()	Returns the number of runs in the window that can be considered of medium intensity. Only the runs with a lactate threshold estimation are considered.
n_of_high_intensity_runs()	Returns the number of runs in the window that can be considered of high intensity. Only the runs with a lactate threshold estimation are considered.
time_at_low_intensity()	Returns the seconds passed at low intensity considering the time of all the runs in the window that have a lactate threshold estimation.
time_at_medium_ intensity()	Returns the seconds passed at medium intensity considering the time of all the runs in the window that have a lactate threshold estimation.
time_at_high_intensity()	Returns the seconds passed at high intensity considering the time of all the runs in the window that have a lactate threshold estimation.
time_at_low_intensity_pct()	Returns the time in percentage passed at low intensity considering the time of all the runs in the window that have a lactate threshold estimation.
time_at_medium_ intensity_pct()	Returns the time in percentage passed at medium intensity considering the time of all the runs in the window that have a lactate threshold estimation.
time_at_high_intensity_pct()	Returns the time in percentage passed at high intensity considering the time of all the runs in the window that have a lactate threshold estimation.

Table 3.8: Window statistic functions concerned with running intensity distribution.

Function Name	Description
running_load_rtss()	Returns the total training load (rTSS) considering all the runs in the window that have a functional threshold pace estimation.
max_run_load_rtss()	Returns the maximum training load (rTSS) observed considering all the runs in the window that have a functional threshold pace estimation.
avg_run_load_rtss()	Returns the average training load (rTSS) considering all the runs in the window that have a functional threshold pace estimation.
running_load_edwards()	Returns the total training load (Edwards TRIMP) considering all the runs in the window that have heart rate and a max HR estimation.
max_run_load_edwards()	Returns the maximum training load (Edwards TRIMP) observed considering all the runs in the window that have heart rate and a max HR estimation.
avg_run_load_edwards()	Returns the average training load (Edwards TRIMP) considering all the runs in the window that have heart rate and a max HR estimation.
avg_volume_acwr()	Returns the average 'Acute:Chronic Workload Ratio' value of the days contained in the window.
max_volume_acwr()	Returns the maximum 'Acute:Chronic Workload Ratio' value of the days contained in the window.

Table 3.9: Window statistic functions concerned with training load distribution and balancing.

Function Name	Description
n_of_races()	Returns the number of runs labeled as 'Race' in the window.
n_of_base_runs()	Returns the number of base runs (uniform, low intensity and with duration and distance close to the average one) contained in the window. Only the runs with a lactate threshold estimation are considered.
n_of_long_runs()	Returns the number of long runs (uniform, low intensity and with duration and distance significantly greater than the average one) contained in the window. Only the runs with a lactate threshold estimation are considered.
n_of_recovery_runs()	Returns the number of recovery runs (uniform, low intensity and with duration and distance significantly lower than the average one) contained in the window. Only the runs with a lactate threshold estimation are considered.
n_of_threshold_runs()	Returns the number of threshold runs (uniform and mainly performed around lactate threshold) contained in the window. Only the runs with a lactate threshold estimation are considered.
n_of_anaerobic_runs()	Returns the number of anaerobic runs (uniform and mainly performed above lactate threshold) contained in the window. Only the runs with a lactate threshold estimation are considered.
n_of_interval_runs()	Returns the number of interval runs (with one or more sequences of intensity peaks that span over the majority of the session) contained in the window. Only the runs with a lactate threshold estimation are considered.
n_of_surges_strides_runs()	Returns the number of surges/strides runs (with one or more sequences of intensity peaks that do not span over the majority of the session) contained in the window. Only the runs with a lactate threshold estimation are considered.
n_of_progression_runs()	Returns the number of progression runs (that exhibit a significant progression in intensity during the session) contained in the window. Only the runs with a lactate threshold estimation are considered.
n_of_fartlek_runs()	Returns the number of fartlek runs (that are not uniform in intensity but do not follow a progression) contained in the window. Only the runs with a lactate threshold estimation are considered.

Table 3.10: Window statistic functions concerned with classification of running workouts.

3.5.4 Training load balance parameters

Since in the AthleteCalendar each activity is associated with a precise date, inside this object is also possible to analyze the trend of several time-dependent parameters such as the training load balance ones.
In our specific case, we implemented the computation of the *acute: chronic workload ratio* index (as described in Section 2.2.4) by considering the daily distance volumes present in the calendar of runs and consequently estimating fitness, fatigue and ACWR itself in order to assign their appropriate values to each day in the time period considered. In particular, to perform the calculation we used an exponentially weighted moving average with a span of 28 and 7 days for fitness and fatigue respectively.
Fig. 3.13 shows an example of the results obtained in a one-year time frame of activities.

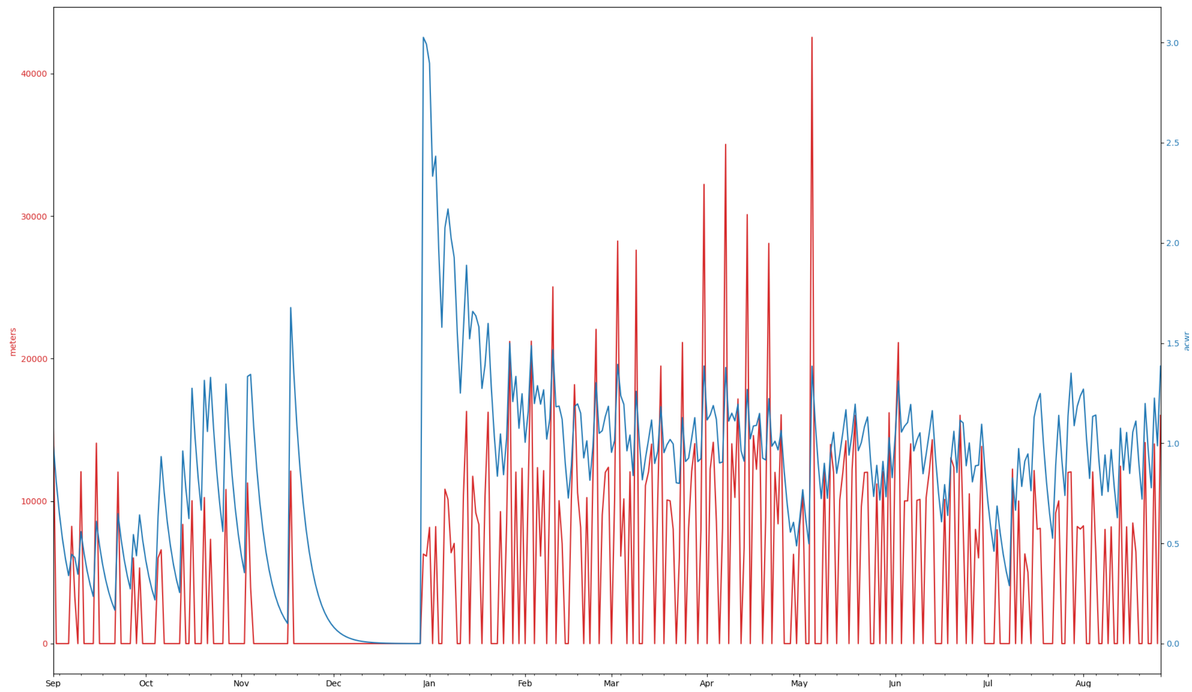


Figure 3.13: Daily training volumes (in red) and ACWR index evolution (in blue).

Once the evolution of these parameters has been computed entirely, the indices and the respective dates associated to them are saved inside the AthleteCalendar for future use. Whenever a CalendarWindow is created, the relevant portion of this database is then passed to each window so that an analysis regarding this aspect can be performed by means of some particular statistics functions.
This feature allows us to potentially gain insights on the training load distribution habits of the athletes considered and, additionally, can give us hint on the best or worst strategies adopted by amateur runners in this sense.

Chapter 4

Experimental Design

In this chapter, we will describe the design steps followed when preparing for the experimental analysis to be performed on the athletes data. First, we will provide an overview of the data collection process and the resulting dataset creation. After that, we will discuss the motivations and preparation stages behind the methodologies adopted in the experiments. Finally, we will give some background information on the statistical tools utilized and the workflow we employed during testing.

4.1 Data collection

Before addressing the methodologies utilized in our experimental data analysis, we first focus our attention on the decisions made during the data collection process. Specifically, regarding the quantity and nature of the information gathered in this crucial phase. In particular, in this section we will delineate the scope taken into account during the acquisition stage (concerning both the athletes and the training horizon considered) along with the actual detailed information saved for each single runner and activity. After that, we will shortly deal with the data cleaning problems confronted at the end of this operation. Finally, a brief description and visualization of the data obtained will be given in order to provide a clear picture of the dataset used in the next steps of the analysis. All the data gathered in this phase was collected from the Strava platform [1].

4.1.1 Athletes and training activity considered

For both athletes and their respective training history an initial selection process was designed by keeping in mind the aim of our various analysis investigations requirements. Because of this, a number of constraints were set for purpose of obtaining only strictly relevant data from the individuals representing adequately our target.

Athletes selection

Since in our work we wanted to focus mainly on amateur runners, to generate our analysis data we decided to extract a random sample of athletes by just requiring a minimum amount of training data consistency to discard extremely casual runners or individuals

[1] `https://www.strava.com`

just approaching the sport (which might not be even considered of amateur level). This was done to ensure that all the athletes involved in the study would, in fact, represent a typical non-professional runner, equipped with some decent training experience and an active workout history that contained at least a fair amount of consecutive running sessions.
Next, given that most of our processing (as seen in the previous chapter) requires heart rate data to function properly, of all these individuals only the ones having the majority of running data with this information available were kept.
Finally, in addition to these general constraints, we further based our selection criterion on the nationality of each subject. In fact, we made sure to create a balanced dataset containing a similar number of athletes for every country taken in consideration (i.e. **Italy**, **Belgium**, **Japan**, **United States**).
The reason why we performed this specific selection was to eventually implement some experiments based on the influence of the nationality aspect on training habits. In any case, the details of this choice will be discussed more in depth inside Section 4.2.

Workout activity selection

For what concerns the time-horizon and category of workout data collected, we chose to adopt the same exact strategy for every athlete selected in the previous step.
In particular, a time period of **12 months** (**from Sept. 2018 to Sept. 2019**) was considered for the extraction of all the workout activities present in the training log of the amateur runners.
Among the reasons of this decision, we were driven by the need to exploit a large enough interval (for medium to long term analysis) that, however, needed to be before the SARS-CoV-2 pandemic outbreak at the end of 2019. Indeed, we all know that this incident profoundly changed the lifestyle of all the people around the world. Consequently, also the training habits of all sort of sports changed after this event (e.g. due to lock-downs and restrictions). For this reason, we decided to avoid this particular period since it was not the intended focus of our analysis.
In regard to the activity type selection, instead, we obviously had to include all running activities contained inside this time period for each athlete. However, also the non-running sessions were additionally collected in the form of "cross-training" workouts. Specifically, this was done to perform further analysis inquiries regarding the impact of multiple sport training in addition to running, as we will see later.

4.1.2 Information gathered for each athlete and activity

So far we mentioned the extent of our research in terms of all the high level information retrieved when considering the available athletes and their workouts. Conversely, we will now proceed towards presenting in more detail the structure of the actual data that was gathered in practice during the entire sampling process.

First off, before the collection of all the activities information, some essential data on each athlete was identified and saved in our database for later use. Particularly, the sex and nationality of all the individuals was extracted at this point and associated to its specific runner. Following, a dataset of all the workout activities inside the time interval

considered was generated by specifying some important summary information for each one of them (the whole collection of attributes is presented in Table 4.1).
In the case of cross-training activities, however, only a limited number of these details were saved, since we were not interested in many of the in-depth aspects of this kind of training sessions. Because of this, just the date and duration of these efforts were kept in order to have at least the possibility to consider them in relation to the runs and to analyze their impact on them.

Name	**Description**
UTC Time	Date and timestamp (in Coordinated Universal Time) in which the activity started.
Local Time	Date and timestamp (in local time) in which the activity started.
Duration	Duration of the whole activity.
Distance	Distance covered in the running activity. (this field is left empty in non-running activities)
Label	Optional label assigned to the activity. It's used to mark special workout sessions as races.
IsRunningActivity Flag	Attribute signaling if this activity was a running or a non-running one.

Table 4.1: Summary attributes saved for each activity in an athlete's training log.

While for the cross-training activities this brief summary was enough from an analysis point of view, for each run in the dataset we also had to save the actual data streams composing the activity in all its sensor measurements. Consequently, every running workout was also associated to a related JSON file containing these exact streams (for a detailed description of the content of each file refer to Section 3.2).

The information gathered with this particular data structure (for athletes and activities) was crucial in order to perform all the processing steps described in the previous chapter, starting from the single low level activities up to the athletes calendars presented earlier. Notably, the combined use of the summary attributes (specifically the timestamps) and the streams included in the files allowed for the implementation of all the AthleteCalendar features illustrated in Section 3.5. Nonetheless, since we designed our processing architecture to be able to deal with as little information as possible, we could exploit most of the input data, given that the requirements were minimal.

4.1.3 Data cleaning operations

Before being able to use the workout data collected, we had to perform some cleaning and preparation steps on the raw observations in order to limit the presence of erroneous samples in our dataset.
Multiple activities presented empty, missing or faulty streams that rendered their use impossible. For this reason, we identified and eliminated these categories of activities:

- Runs with empty or missing values for any essential stream (e.g. speed, distance)

- Runs with duration or distance of a few seconds
- Cross-training activities with a duration lower than 5 minutes
- Immediately evident outlier activities (with obviously wrong values)

At the end of the whole cleaning process a total of **3878** activities were deleted from the initial data.

4.1.4 Collected data

After the data collection and cleaning processes we obtained the dataset that is described in the following subsections.

Athletes

A total of **398** athletes were gathered with a distribution of nationality and sex as shown in Fig. 4.1 and 4.2.

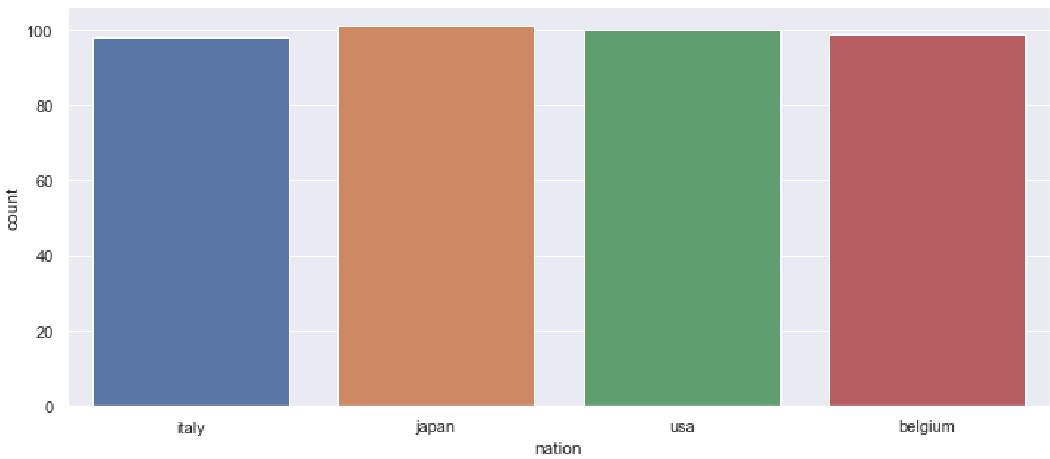


Figure 4.1: Number of athletes collected from Italy (98), Japan (101), USA (100) and Belgium (99).

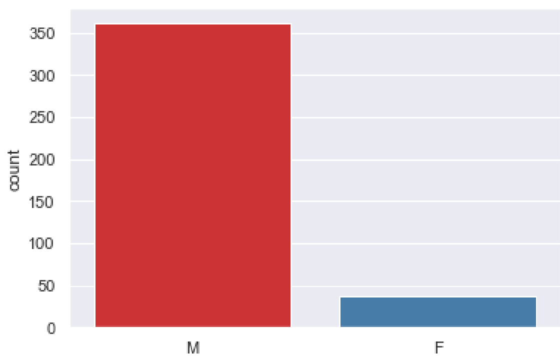


Figure 4.2: Number of female (37) and male (361) athletes collected.

In particular, we can see that the representatives of each nation are almost identical in number (design choice) while there is a substantial majority of male runners compared to the female counterpart (not a design choice).
However it must be noted that, since our athletes selection was not based in any way on the sex attribute, this gap was not intentional but merely the result of a random selection. In any case, this found disparity was not relevant for our specific analysis purposes.

Activities

For each one of these athletes above an entire year of workouts was then collected with a total amount of **79536** activities.
A breakdown of the ratio between runs and cross-training sessions is present in Fig. 4.3, along with the distribution of all these activities inside the time-interval specified previously (in Fig. 4.4).

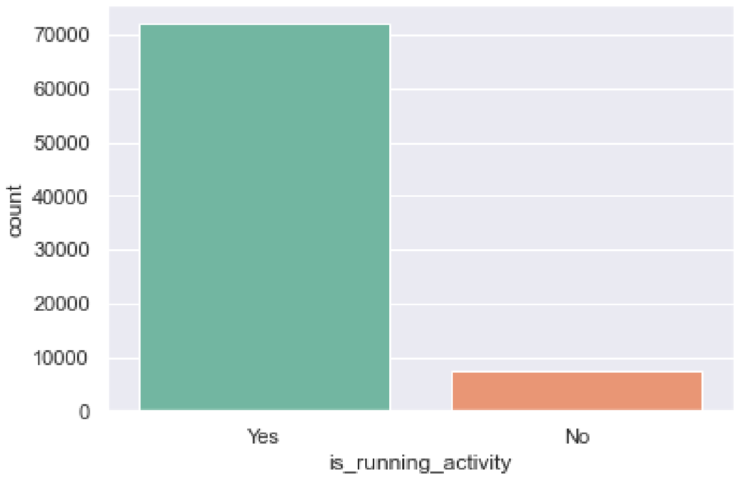


Figure 4.3: Running (72048) vs. non-running (7488) activities collected.

Since we selected amateur athletes whose main sport is running, it does not surprise that the vast majority of workouts gathered was of this particular type (with respect to cross-training). Having said that, there is still a relevant number of non-running sessions that we were able to utilize during our analysis.
A visualization of all the running activities is given in Fig. 4.5 where for each run the distance and duration is shown. Additionally, for all the runs labeled as race by the athletes a different symbol is adopted.

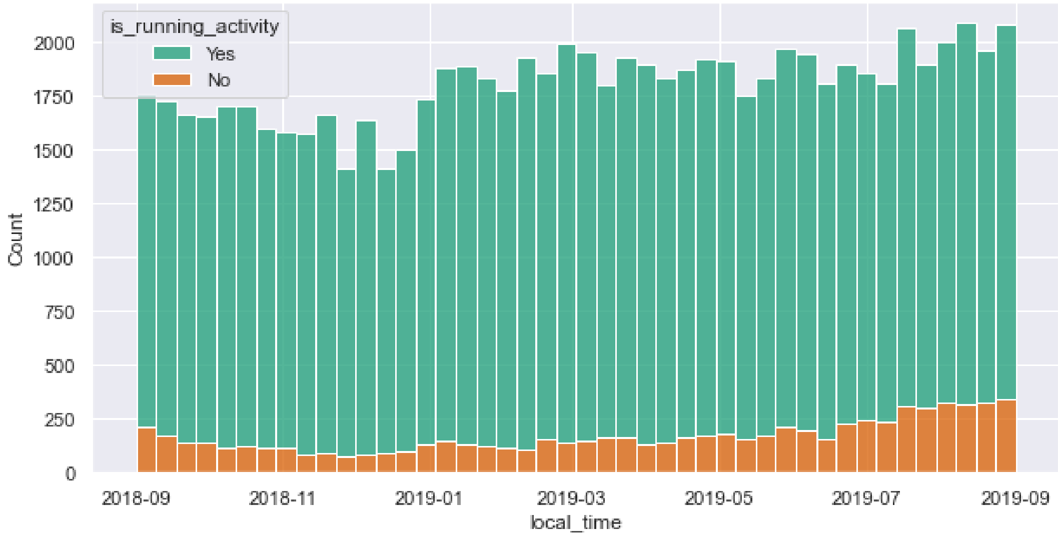


Figure 4.4: Distribution of running and non-running activities inside the time-period where they were collected.

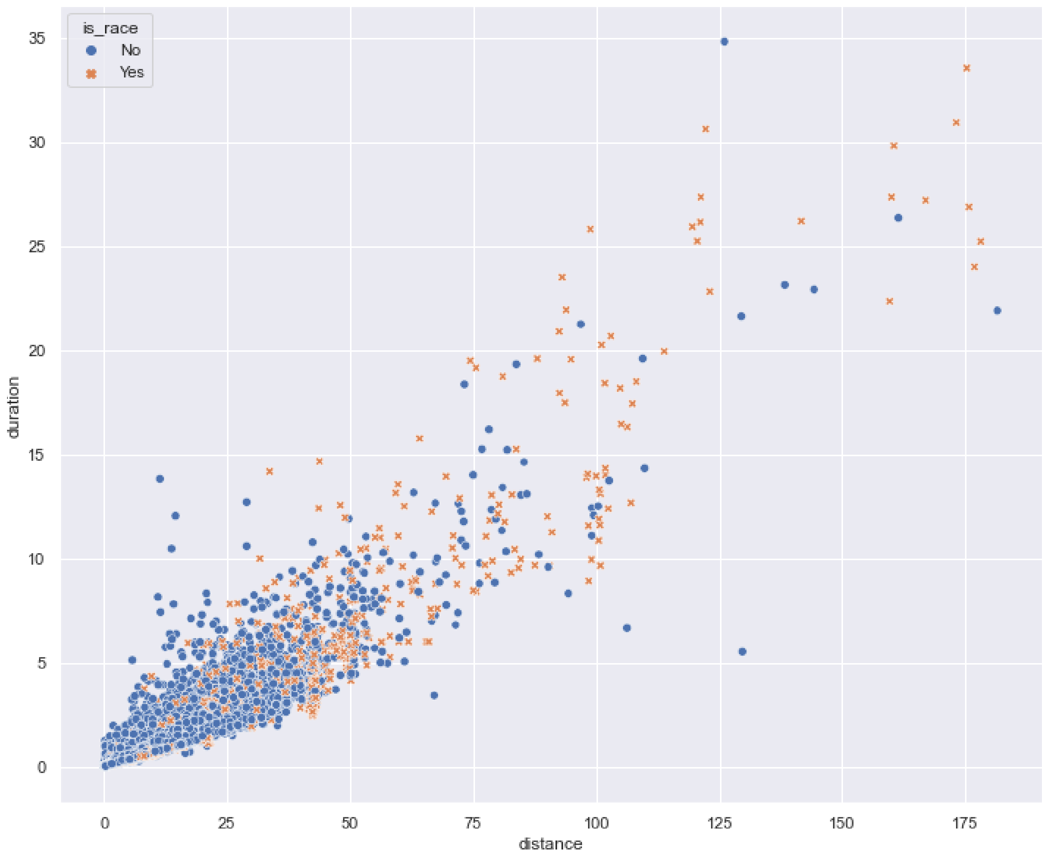


Figure 4.5: Running activities expressed as distance (in Km) and duration (in hours). The runs labeled as races by the athletes are highlighted in the plot.

4.2 Analysis methodologies

With the aforementioned data collection process presented, we now move to the explanation of actual experiment design and all the related methodologies we decided to involve in our investigations.

The main objective of our study was aimed at understanding the training habits of amateur runners around the world. For this reason, in order to gain in-depth insights, we developed the entire processing framework structure to be a able to provide both low and high level information on the workout routines of each single athlete considered in our analysis (given a large enough amount of training data available). Together with a targeted data collection, these tools then gave us the possibility to come up with a series of questions that could be answered by performing some statistic evaluations on a considerable quantity of subjects.
This method ultimately allowed us to carry out the analysis task that are described in detail in the following sections.

4.2.1 Impact of nationality on training habits

The fist study subject we wanted to address was relative the possibly existing geographical difference among amateur runners with regards to the philosophy of training and the diverse workout consequently adopted in these nations.

Motivations

It's undeniable that different schools of thought exist when it comes to coaching or training in general for the sport of running. In particular, these differences are usually more evident when considering various areas around the world and, especially, some nations that have a rooted history in this discipline. Because of these reasons, we decided to set up an experiment to understand if the athletes nationality actually impacted on the training methods used when performing the usual workout activity.
As we anticipated in the data collection section, the nations chosen for this study were Italy, Japan, USA and Belgium. The motive behind this decision was mainly due to the fact that we wanted to consider countries both geographically and traditionally distant. Furthermore, with this choice, we also wished to cover most of the continents in order to extend as much as possible the analysis scope. Unfortunately, however, we were also limited by the data and computational resources available. Therefore we had to exclude some parts of the world and limit our research to these few states.

Experiment setup

Since a training plan is usually defined and carried out on a weekly basis, in order to understand the characterizing aspects regarding each athlete's workout habits we opted for an analysis of the single weeks composing the 12 months of activities present for every runner. This way, we aimed to discover if some considerable differences were present on the methods used to manage this important building block of training among the nations taken into account.

For the execution of this experiment we exploited the data of all the athletes available in our dataset. Specifically, **all the weeks** (Monday to Sunday) **starting from September the 3rd 2018 until August the 25th 2019** were sampled for every individual athlete to subsequently compute some statistics on them.
To achieve this, for each runner involved the relative AthleteCalendar object was initialized generating the entire collection of high level activities (in the 12 months) after the correct estimation of all the personal physiological parameters. Next, using the calendar window functionality described in Section 3.5.3, the workout sessions were divided in 7-days windows (the weeks) starting from the date mentioned above. Finally, all the statistics functions in tables 3.6, 3.7, 3.8, 3.9 and 3.10 were computed on each single generated week to extract a number of interesting features describing them.
The result of this operation was the generation of a collection of weeks samples each representing the respective country (of the athlete that performed it) along with the outcomes of the statistics (features) and some combination of the latter. In addition to this information, to each sample we also associated several flags (listed in Table 4.2) to have the possibility to eventually exclude some of them from different analysis configurations. In fact, we already mentioned when describing each statistic function that some of them only consider the runs that have certain crucial information (e.g. heart rate or lactate threshold estimation) to compute their results. Therefore we signaled the eventual lack of this knowledge in the samples to try different trade-offs using less or more data depending on how precise we wanted our weekly features to be.

Flag Name	**Description**
has_empty_runs	Checks if in the sample week there were no running workouts. Used to eventually exclude empty weeks from the analysis since they might not be considered part of a training plan.
all_runs_with_hr	Checks if all the runs in the sample week contain heart rate information.
all_runs_with_lactate_threshold	Checks if all the runs in the sample week contain a lactate threshold estimation.

Table 4.2: Flags associated to each sample week generated during the analysis of the impact of nationality on training habits.

Being in possess of this data coming from all the different countries considered, the idea of the experiment was then to perform multiple ANOVA tests in order to detect if a statistical significant difference was present between the groups (countries) on at least some of the weekly features observed among the athletes. In particular, a one-way ANOVA test was performed for each statistic in examination, as we will see in the next section. Before executing each test, however, all the available weeks were grouped by athlete to generate a representation their typical weekly routines. Consequently, the runners appearing in any given experiment on nationality were each only represented by a single week averaging all their other individual ones.

4.2.2 Seasonality of training

Another aspect that we considered during our analysis was the one related to training seasonality. Namely, we were interested in how seasons during the year affect the training habits of amateur runners.

Motivations

Intuitively, we know that meteorological conditions can affect outdoor sports such as running. In particular, both extreme temperatures and adverse weather have a negative impact on performance and can consequently influence the outcome of a workout session. Because of these reasons, many amateurs tend to follow a training plan that is usually centered around the main seasons constituting the whole year. For athletes competing in races, however, the entire preparation can also be obviously dependent on the dates of local competitions. Either way, generally speaking the two things are often connected and correlated.
In our analysis we were interested in finding these and potentially other variations among the runners considered, therefore we decided to perform some experiments dividing the activity collected for each athlete in the four main seasons.

Experiment setup

The preparation for this experiment was very similar to the one we presented in the case of nationality. As a matter of fact, we reused exactly the same week samples generated with the previous method, only this time we assigned to all of them a label indicating the corresponding season. In order to determine the correct one for each week, we utilized the meteorological definition of season (for the Northern Hemisphere) which considers as beginning date the first day of the months that include the equinoxes and solstices. However, for convenience, we adapted this rule in our case by using instead the first week starting inside the corresponding months. Consequently, the weeks available were divided as shown in Table 4.3 (the last week of summer is missing since we did not have the relative data).

Season	**Date interval**
Autumn	3/09/2018 - 2/12/2018
Winter	3/12/2018 - 3/03/2019
Spring	4/03/2019 - 2/06/2019
Summer	3/06/2019 - 25/08/2019

Table 4.3: Date interval for each season used in the seasonality of training analysis.

Also in this case, we then executed a number of experiments using the ANOVA framework to detect differences in the weekly statistics between the various seasons considered. Only this time we grouped the weeks by considering both athlete and season in order to generate four samples for each runner (one per season). The practical details regarding the experiment implementation are described in the next section.

4.3 ANOVA tests format

All the experiments executed during our tests follow the same format, which is relative to the functioning of ANOVA. Depending on the task, we used two different configurations in order to obtain the desired results. In particular, we will now define all the steps constituting both one-way and two-way ANOVA as implemented in our analysis.

4.3.1 One-way ANOVA

The ANOVA test [1] is typically used to detect whether a statistical difference exists between multiple groups of samples on a particular outcome variable mean. In this first case, only a single factor (e.g. nationality) is considered as the independent variable on which the grouping is made. By producing a test statistic that considers the ratio of variances among means and within samples, the null hypothesis of equal means among groups can then be rejected if some conditions are satisfied.
The sequence of actions below represents the implementation of this test adapted to the specific needs of our experiments:

1. Selection of the independent (factor) and outcome variable
2. Samples generation and optional filtering
3. One-way ANOVA test
4. Eventual assumptions check
5. Eventual post-hoc analysis

In the first step of the process, both the independent (factor) and dependent variable (outcome) chosen for the single experiment are specified. In this context, the categorical values of the independent variable are used to divide the data into the groups we want to analyze.
After this initial phase, in the second step we decide which samples will be used during the test (in our case the weeks). Depending on the outcome variable we are considering, some of these samples can be discarded to avoid including in the analysis unreliable data that might skew the results (e.g. outliers). At the end of this selection procedure, we group the resulting weeks by athlete and/or season to generate the final data points that will be utilized in the actual test.
In the third step, the One-way ANOVA test itself is executed using the statsmodels library implementation [2] and, consequently, a table with all the results information is generated (see Table 4.4). If we succeed in rejecting the null hypothesis (at least one group is different from the others) the experiment continues with the remaining steps. In any other case, the process stops as we were not able to determine a significant difference among the groups (nothing can be said).
If a positive result is found, the fourth step serves the purpose of checking that all the ANOVA assumptions were respected in order to consider the outcome valid. Particularly, the following elements are tested:

[2] `https://www.statsmodels.org/stable/anova.html`

- The dependent variable must be approximately normally distributed in each group
- The groups must have equal variance
- The observations must be random and the samples taken from the populations must be independent of each other (depends on experiment design)

In our case, normality is verified with both the Shapiro-Wilk test and the probability plot. While homogeneity of variance is checked with either Bartlett's or Levene's tests along with the groups boxplots (all tests use the Scipy library implementation [3] [4] [5] [6]). Since ANOVA is an omnibus test (it checks for a difference overall between all groups), if a statistically significant and valid result is found we are not able to tell which groups are different from the others. For this reason, in the last step of the process post-hoc testing is performed in order to determine the pairwise differences among all groups. To achieve this the following methods are used: Tukey Honestly Significant Difference (HSD), Bonferroni Correction, Dunn-Sidak Correction (as implemented in [7]).

Term	**Meaning**
sum_sq	Sum of squares
df	Degrees of freedom
mean_sq	Mean square
F	F statistic value for significance
PR(>F)	P-value for significance
eta_sq	η^2 effect size
omega_sq	ω^2 effect size

Table 4.4: Results information generated after the execution a One-way ANOVA test.

4.3.2 Two-way ANOVA

In addition to the experiments done to detect the influence of a single factor (i.e. nationality or seasonality) on some dependent variable (e.g. weekly volume), during our analysis we also tested the effect of an interaction between the two factors considered. That is, if the outcome variable depends on the influence of one factor on the other. In order to do this, we used the two-way ANOVA configuration (two factors) by also including the interaction term inside the corresponding model. The steps followed for this type of experiments are exactly the same ones that we already discussed in the one-way case, except for the test itself that this time considers both independent variables simultaneously. In these instances the weeks used as samples were grouped in the same way as in the seasonality case.

[3] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.shapiro.html
[4] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.probplot.html
[5] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.bartlett.html
[6] https://docs.scipy.org/doc/scipy-0.13.0/reference/generated/scipy.stats.levene.html
[7] https://www.statsmodels.org/devel/generated/statsmodels.sandbox.stats.multicomp.MultiComparison.html

Chapter 5

Experimental Results

In this chapter, we will present the results obtained during the experimental analysis. For each category of tests, a table with all the configurations relative to the successful experiments will be provided, along with a brief discussion of the single results and some additional summary comments.

5.1 Weekly training pattern

Week attribute	Weeks excluded	Outliers removed	Grouping operation	Factor
n_of_days_with_runs	without runs	-	mean	nation
avg_daily_runs	without runs	245 weeks	mean	nation
n_of_running_workouts	without runs	125 weeks	mean	nation

Table 5.1: Experiments configurations for weekly training pattern analysis.

Number of days with running activities

These experiments tested for a difference between groups of athletes on the average number of days (in a week) where running activities are performed.
When using nation as factor (Fig. 5.1), a statistically significant difference among the groups was found with an overall medium effect size. In particular, after a post-hoc analysis, it was revealed that the United States mean is significantly greater than both Italy and Belgium, while for the other pairwise comparisons the null hypothesis could not be rejected. We can see, however, that Japan seems to show a reasonably higher mean than the lower pair of countries.
The experiment on seasonality didn't show a statistically significant difference between the group means. No remarkable interaction effect of nationality and seasonality was found either.

Average runs on training day

The aim of these experiments was to test for a variation in average number of runs per training day (in a week) between groups. It must be noted that in this case for each week only the days with a run were considered, hence excluding all the off days. This was done because we wanted to understand which athletes preferred to do multiple runs a day or just a single one in their typical training week. In any case, after the elimination of outlier weeks we obtained as input to our tests just one or two avg. runs per-day samples (reasonable).
Testing nationality (Fig. 5.2) ANOVA found a statistically significant difference among the means, albeit with a small effect size. However, with the help of post-hoc analysis, we were only able to find a distinction between Japan and Belgium, where the former turned out to have a higher value of mean daily runs than the latter.
With season as independent variable, we were not able to reject the null hypothesis of equal means. Furthermore, the two-way ANOVA analysis didn't find an interaction between the factors.

Number of running workouts

These experiments were executed to test for a difference in the average number of running workouts performed during the typical week. Differently from the number of days with runs, this analysis focuses on the actual amount of run sessions regardless of the days in which they were carried out. Nonetheless, we will see that most results were similar since the two things are obviously highly correlated.
Considering the nations (Fig. 5.3), as with the number of days with runs experiment, the ANOVA test returned a low enough p-value with a medium effect size. This time, however, the post-hoc testing revealed two separated groups of countries. A statistically significant difference was found between the Italy/Belgium group and the Japan/USA one with the second presenting higher values. Additionally, this time Japan was separated from the lower two nations as well, which partially confirms the uncertain results obtained in the first test.
Using the seasons as factor instead, ANOVA didn't find a meaningful difference among the groups. The same happened for the interaction test as well.

Comments

At the end of this first set of experiments on the influence of nationality we can say that a quite noticeable difference in terms of runs distribution can be observed between the countries in analysis. Particularly, these can at least be divided in two main groups: Italy/Belgium and Japan/USA. The first one presents overall less training sessions both in terms of total and daily average. The Japanese and American runners, instead, seem to display a similar pattern, except that the first ones utilize a slightly less number of training days in exchange for more daily workouts.
Unfortunately, with respect to seasonality we couldn't obtain interesting results as no immediate difference was found in the means examined. This fact might hint at the possibility that the number of weekly training days of these athletes is kept constant during the year. As a final note, we saw that no particular interaction of the two factors was

identified which further support this speculation. However, it would be interesting to repeat this analysis with a greater amount of data points and see if a distinction appears.

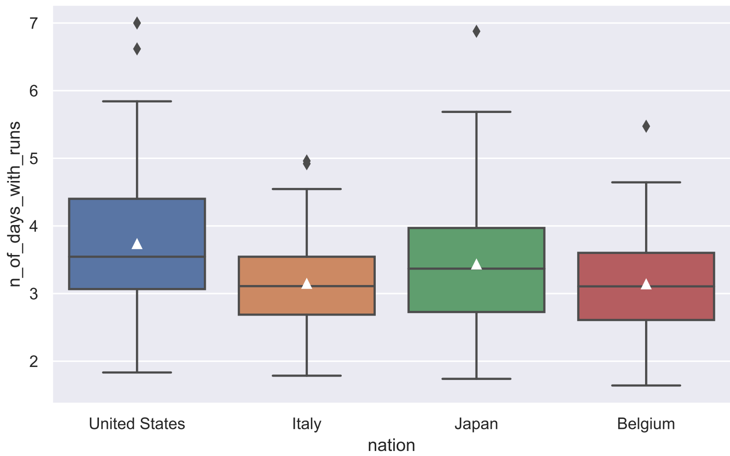


Figure 5.1: Boxplots for n_of_days_with_runs experiment (nationality).

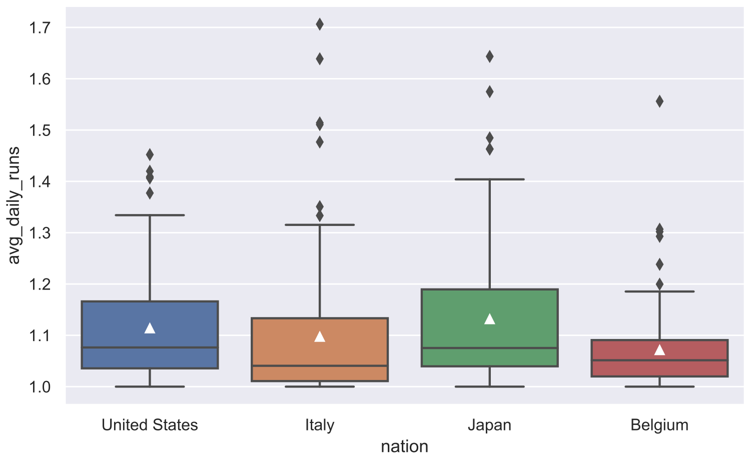


Figure 5.2: Boxplots for avg_daily_runs experiment (nationality).

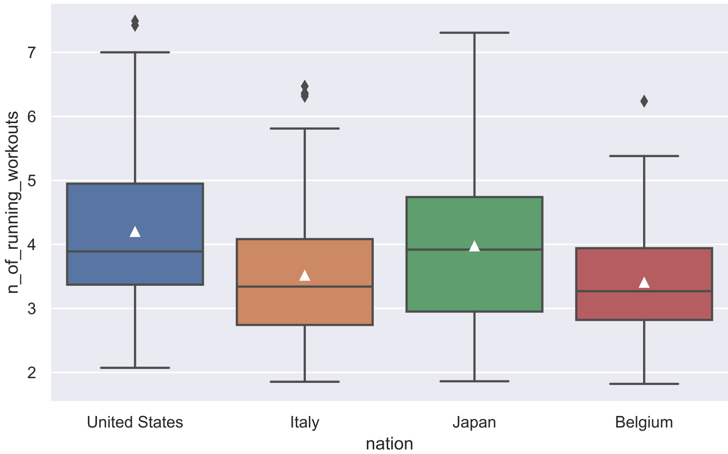


Figure 5.3: Boxplots for n_of_running_workouts experiment (nationality).

5.2 Running duration and volume

Week attribute	Weeks excluded	Outliers removed	Grouping operation	Factor
running_time	without runs	226 weeks	mean	nation
avg_run_duration	without runs	228 weeks	mean	nation
running_volume	without runs	180 weeks	mean	nation
avg_run_volume	without runs	214 weeks	mean	nation
running_volume	without runs	-	median	season
avg_run_volume	without runs	-	median	season

Table 5.2: Experiments configurations for running duration and volume analysis.

Total running time

In these tests we confronted the mean total running time (in seconds) given by the sum of all the running activities present inside a typical week.
In the nationality experiment (Fig. 5.4) the ANOVA results showed a statistically significant difference among groups with a low effect size. In particular, the post-hoc testing showed a difference between the Italy group and both Japan and USA athletes, where the latter two demonstrated an higher overall weekly running time.
The experiment on seasonality and interaction between the two independent variables returned inconclusive results.

Average run duration

Differently from the total running time, with these experiments we wanted to compare the average single run duration (in seconds) inside a typical training week.
With the nationality factor (Fig. 5.5) ANOVA found a statistically significant difference

with a low effect size. Moreover, the post-hoc analysis revealed a that both Japan and Belgium show a considerable higher avg. run duration w.r.t. the USA counterpart.
No statistically significant results were found in the seasonality and two-way tests.

Total running volume

These experiments tested for a difference in total weekly running volume (in meters) given by the sum of all the running activities.
In the nationality case (Fig. 5.6), the ANOVA output showed no significant difference but in the post-hoc analysis Japan turned out to have a significant greater volume w.r.t. Italy.
Concerning seasonality (Fig. 5.8), a statistically significant difference was found, however with a small effect size. After post-hoc, autumn and summer were separated from spring with the latter having more volume.
In this case, the interaction among factors was not considered significant.

Average run volume

For this experiment we focused on testing the average run volume (in meters) inside a typical training week.
In the nationality evaluation (Fig. 5.7) a statistically significant difference was found with a medium effect size. Interestingly, in this case, the only marked difference detected in the post-hoc test saw USA with a lower avg. run volume w.r.t. all the other countries.
Meanwhile, seasonality (Fig. 5.9) showed a difference but with a low effect size. This time the post-hoc analysis revealed a difference between summer (lower run volume) and winter/spring with autumn in the middle (could not be isolated).
Here no remarkable factor interaction was discovered.

Comments

The overall results of the nationality experiment batch highlight some interesting aspects, especially in relation some considerations we previously made. In fact, although inevitably the absolute number of workouts influences the total running time and volume of the week, we can see that some nations show different strategies w.r.t. the single run planning. Specifically, Belgium in some way makes up for the smaller number of sessions with an higher duration and volume per activity, unlike USA which shows the completely opposite behavior. Conversely, Japan appears to maintain quite long sessions despite the higher number of weekly workouts.
Although the seasonality analysis regarding running duration didn't display particular results, with volume we found out consistently lower values for summer in comparison with winter and, especially, the spring season. This gives the idea that for most of the amateur athletes we analyzed, summer seems to be considered as an off season as opposed to the other two we mentioned.
Concerning the interaction between the two independent variables, nothing can be said since we didn't observe any noteworthy phenomenon.

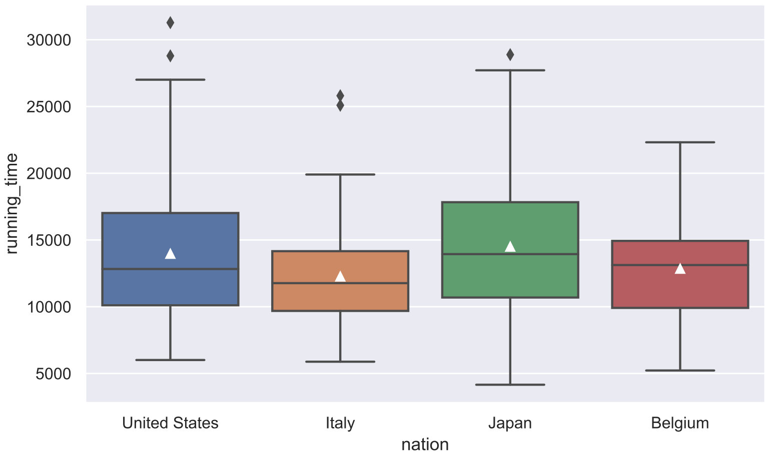


Figure 5.4: Boxplots for running_time experiment (nationality).

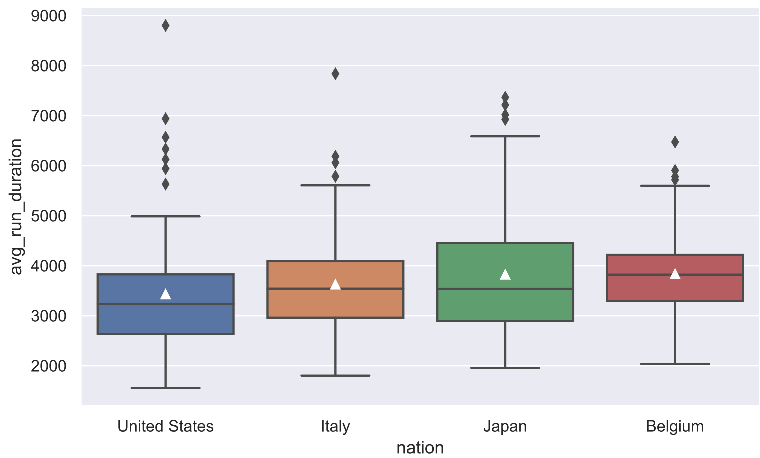


Figure 5.5: Boxplots for avg_run_duration experiment (nationality).

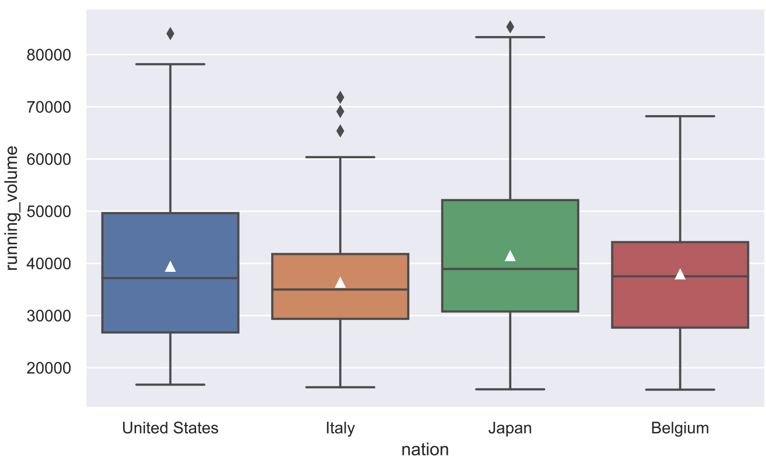


Figure 5.6: Boxplots for running_volume experiment (nationality).

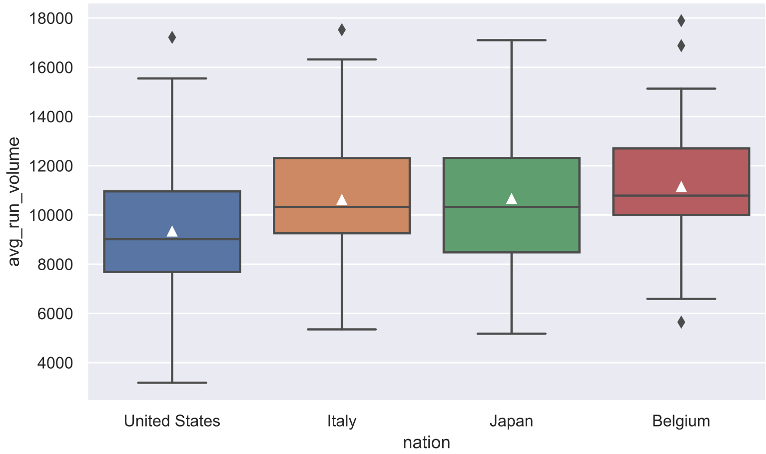


Figure 5.7: Boxplots for avg_run_volume experiment (nationality).

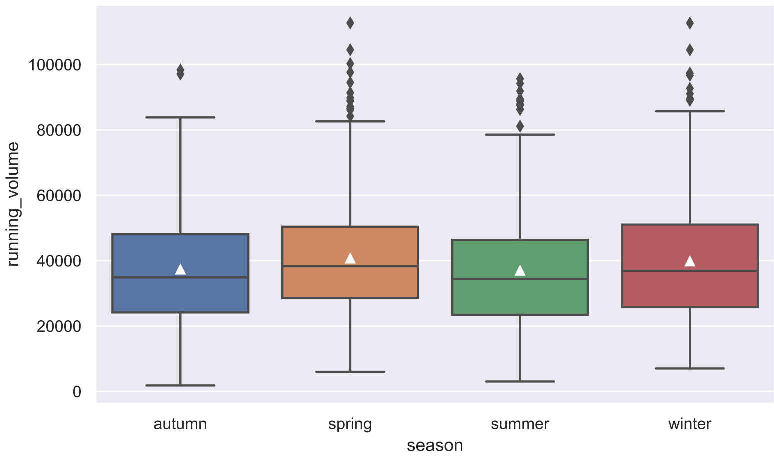


Figure 5.8: Boxplots for running_volume experiment (seasonality).

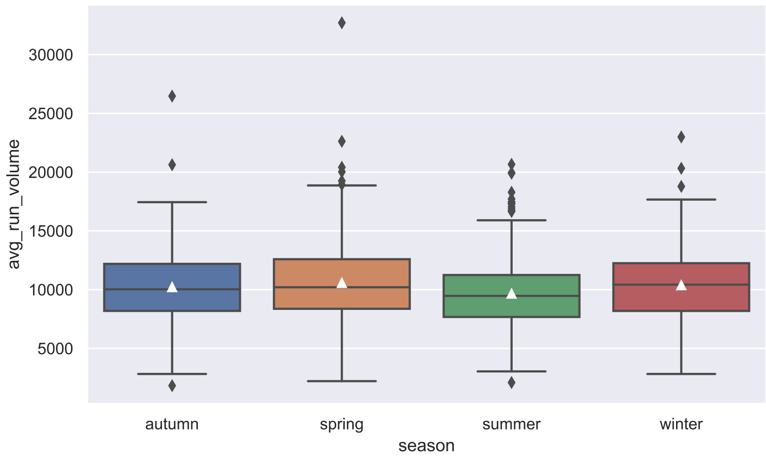


Figure 5.9: Boxplots for avg_run_volume experiment (seasonality).

5.3 Cross-training integration

Week attribute	Weeks excluded	Outliers removed	Grouping operation	Factor
pct_time_running_vs_cross	without runs	-	mean	nation
pct_time_running_vs_cross	without runs	-	mean	season
pct_time_running_vs_cross	without runs	-	mean	two-way

Table 5.3: Experiments configurations for cross-training integration analysis.

Weekly running time vs cross-training time

We prepared these tests in order to understand if a difference in the use of cross-training activities (i.e. workout sessions that are not running) was present among the athletes of the different groups considered. In particular, for every athlete's week we computed a statistic that extracts the percentage of time passed in running activities w.r.t non-running ones.
Regarding the nationality factor (Fig. 5.10), a substantial difference between the groups was found with a total medium effect size. Furthermore, the post-hoc testing revealed Japan as the one integrating less weekly cross-training (compared to running) w.r.t. all the other nations.
In the seasonality experiment (Fig. 5.11) the difference found was also significant but with a low effect size. After the post-hoc analysis summer was separated from the other seasons as it showed a considerable higher cross-training integration.
When testing for the two factors interaction (Fig, 5.12), we found a statistically significant result. Particularly, based on the nationality we noticed that the disparity in cross-training adoption between summer (partially also spring) and winter/autumn changed moderately. In fact, the Belgian athletes presented the highest variation followed by Italy and USA. On the contrary, Japan showed only a minor fluctuation among the seasons.

Comments

Concerning the difference between countries, based on the test presented and given that some other experiments we performed showed very similar results, it appears that Japanese athletes are the ones complementing the least their main running workouts with other sports. Considering that we have seen a big volume and duration of the Japan runs in previous analysis, this lack of cross-training integration could be related to an already packed training week that doesn't allow for the addition of further exercise activity.
On the other hand, with seasonality we discovered that during summer many athletes seem to supplement their main running workouts with other types of activities. Possibly we think that this could be related to a more favorable whether combined with a reduced overall training, as we have seen and will be seeing with other experiments. In any case,

with this test we observed a considerable skew of the values towards no cross-training (many athletes do not per form it) which conflicts with the ANOVA assumptions. Consequently, we believe that in this regard a different kind of statistic test could give us a more reliable output.
Testing the interaction between nationality and seasonality we saw that Japan, in addition to being the country that uses less cross-training overall, is also the one that keeps its usage essentially constant throughout the year. Instead, Belgian athletes in particular tend to change their behavior quite a lot.

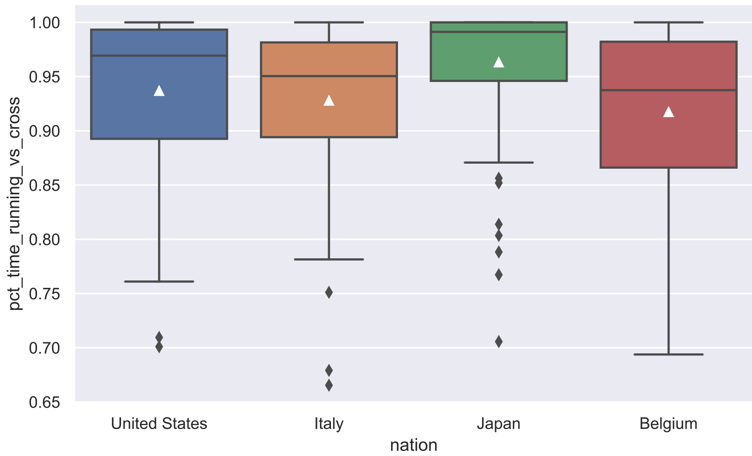


Figure 5.10: Boxplots for pct_time_running_vs_cross experiment (nationality).

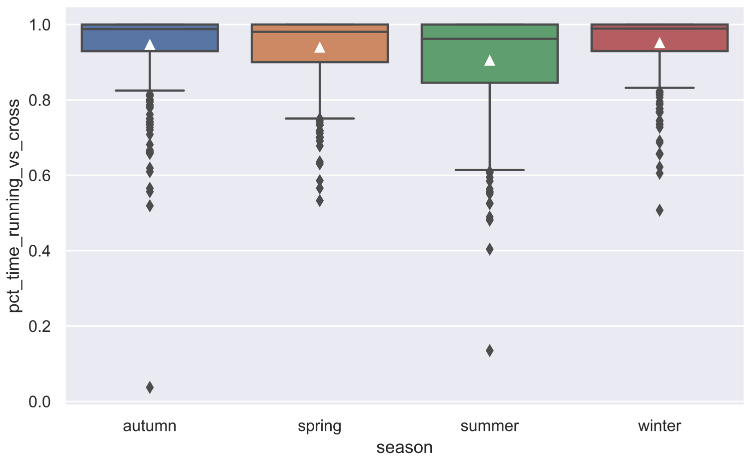


Figure 5.11: Boxplots for pct_time_running_vs_cross experiment (seasonality).

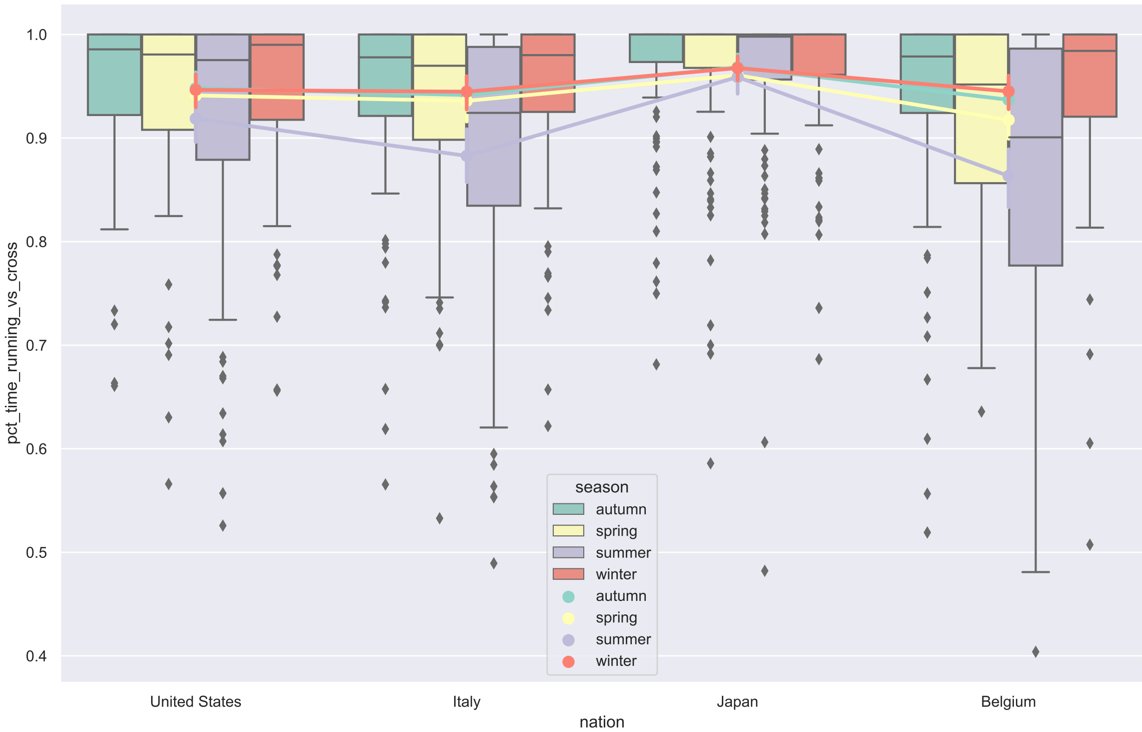


Figure 5.12: Boxplots for pct_time_running_vs_cross experiment (nationality/seasonality).

5.4 Elevation gain and hill running

Week attribute	Weeks excluded	Outliers removed	Grouping operation	Factor
elevation_gained	without runs	-	median	nation
avg_incline_grade	without runs	-	median	nation

Table 5.4: Experiments configurations for elevation gain and hill running analysis.

Total elevation gained

The objective of these experiments was to test for a difference in the total elevation gained (in meters) during a typical training week among the groups of runners considered. Since this result might be highly dependent on the geography, we expected to find some interesting findings confirming and adding to this theory.
The ANOVA test results for the nationality factor (Fig. 5.13) discovered a statistically significant difference between the means of the countries analyzed with a medium to high effect size. Particularly, after the Tukey HSD post-hoc test, Japan showed the highest mean elevation gain followed by Italy, which was separated positively also from Belgium. The USA athletes, instead, only exhibited a meaningful lower value w.r.t. Japan. Using the Bonferroni post-hoc test, however, we were able to divide all the nations in three

distinct groups (of increasing mean elevation gain): Belgium → USA/Italy → Japan.
Concerning the seasonality and two-way analysis, the tests showed no apparent difference that allowed for the null hypothesis rejection.

Average incline grade

With these other experiments we wanted to analyze the eventual difference in the average incline grade observed during the runs of a typical training week for the different groups considered. In comparison to the total elevation gain (that can be influenced by run duration and distance), this measure is focused on the actual inclines present in the running activities and, consequently, considers a different aspect regarding the workouts nature.
In the nationality case (Fig. 5.14), the ANOVA test result showed the presence of a statistically significant difference among the groups, though with a low effect size. After post-hoc analysis, however, the only highly relevant separation placed Italy ahead of Belgium and USA (the latter was not found in Bonferroni and Sidak tests) in terms of mean incline grade.
Using season as factor instead, it was not possible to determine a large enough difference between the groups. Similarly, the interaction test didn't return significant results.

Comments

From the results of both nationality experiments we can see that Belgian runners are the ones that include less inclines overall in their training, which is to be expected given that the country is mostly composed by flat land. For similar reasons, it also appears that the other three nations resort to a good amount of hill running, with Japan leading on the total elevation gained per week. Given that, we know that the reason of this difference depends at least partially on the higher volumes utilized by Japanese athletes.
The inconclusive results concerning elevation gain and seasonality lead us to believe that no particularly big difference in means exists among the seasons of the year. Moreover, the absence of a notable interaction between both factors doesn't hint at some hidden pattern.
However, in these experiments we observed a modest amount of outlier runners in each group which might have influenced the ANOVA score. In any case, the boxplots we obtained during testing (not shown) didn't highlight any evident distinction either.

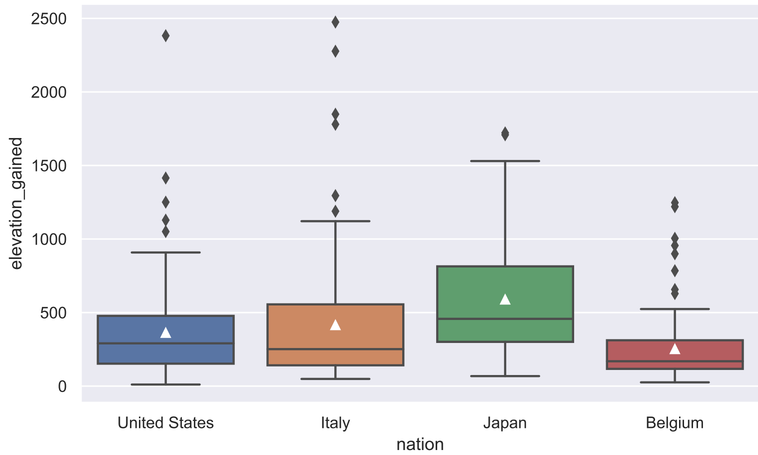


Figure 5.13: Boxplots for elevation_gained experiment (nationality).

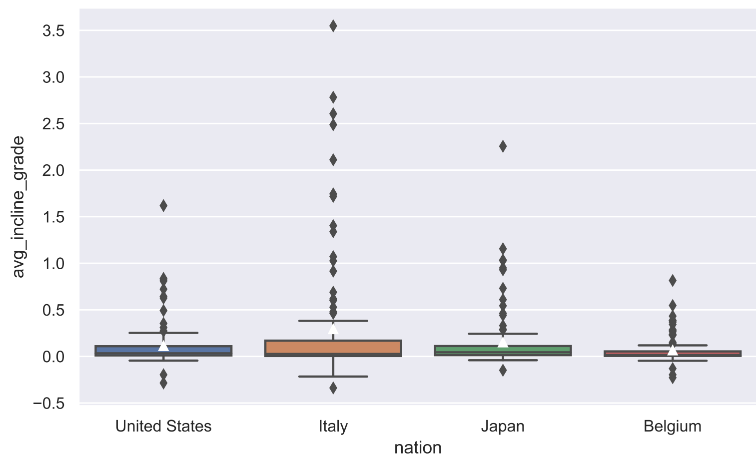


Figure 5.14: Boxplots for avg_incline_grade experiment (nationality).

5.5 Training intensity distribution

Week attribute	**Weeks excluded**	**Outliers removed**	**Grouping operation**	**Factor**
time_at_low_intensity_pct	without runs, without HR, without LT	-	median	nation
time_at_medium_ intensity_pct	without runs, without HR, without LT	-	median	nation
time_at_high_intensity_pct	without runs, without HR, without LT	-	median	nation
time_at_low_intensity_pct	without runs, without HR, without LT	-	median	season
time_at_high_intensity_pct	without runs, without LT	-	median	season

Table 5.5: Experiments configurations for training intensity distribution analysis.

Percentage of running time at different intensities

This set of experiments was aimed at verifying the distribution of running intensity during the entire training time of a typical week. In order to do that, we decided to only consider three main intensity classes: low, medium and high. In particular, all these zones were built around the intensity at lactate threshold, which is represented by the central one. Whereas the first and last zone refer respectively to all efforts reasonably lower or greater than LT.
In all three experiments on nationality (Fig. 5.15, 5.16, 5.17) ANOVA found a statistically significant difference between the groups means (with medium effect size). In the first test, the post-hoc analysis separated all countries in the following three groups of increasing low intensity usage: Italy → USA → Japan/Belgium. With respect to medium intensity Italy resulted the one with highest percentage while the other separations were rejected except for Japan (lower) and USA (higher). Finally, at high intensity Belgium showed an overall lower percentage compared to all the others.
With respect to seasonality (Fig. 5.18, 5.19), the ANOVA tests could only find a significant difference in the low and high intensity case. In both experiments, however, the effect size turned out to be low. After the post-hoc analysis we obtained a separation between summer (with an higher low intensity percentage) and winter/autumn. While the spring mean could no be differentiated from the other ones. Concerning the high intensity test instead, only the means of summer (lower) and autumn (higher) were able to be divided.
The two-way ANOVA tests on the interaction between both independent variables unfortunately didn't find significant results.

Comments

From the nationality outcomes we can see that each country seems to use different ratios when it comes to intensity balancing. Runners in Belgium spend only a minimum

amount of time at high intensities and are among the ones with highest low zone percentage. Japan is slightly more polarized (with a greater high intensity mean), however showing still most of its time at low/medium effort. Italian athletes appear to be largely concentrated around the medium zone and, in particular, are the ones that avoid low intensity the most. Finally, USA appears to devote the majority of training to medium and (partially) low intensity with a mean that is somewhere in between all the other countries for most of the experiments.
Regarding seasonality, the results we found are consistent with the apparent fact that during summer the athletes seem to prefer performing lower intensity runs. Nonetheless, the outcomes obtained for high and medium intensity usage don't really say anything new in this respect. Having said that, we have some doubts on the validity of the autumn results (in all cases), since we know they might be influenced by the weeks without LT that we had to discard in order to compute the intensity zones.
In any case, as no significant interaction was detected between the factors, it looks like the two main separate influences are the only relevant ones present in this instance.
On a final note, it's interesting to notice that the mean percentage of time passed at low intensity in all experiments hardly exceeded the 50%. This particular result emphasizes the belief that most amateur runners indeed lie far from the 80/20 ratio proposed by Fitzgerald in his book ([15]).

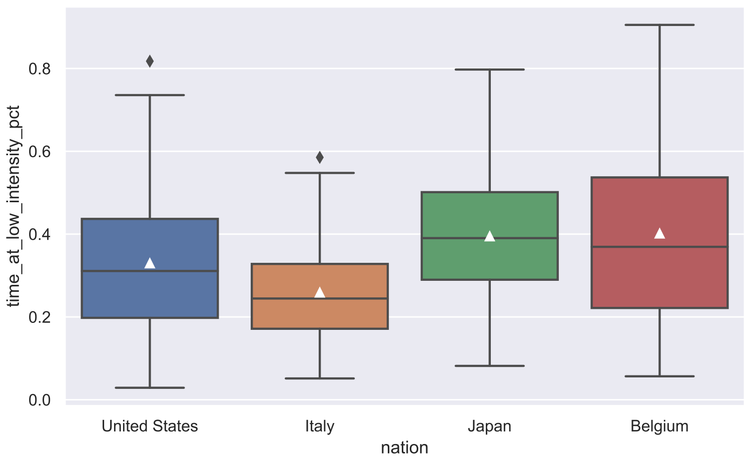


Figure 5.15: Boxplots for time_at_low_intensity_pct experiment (nationality).

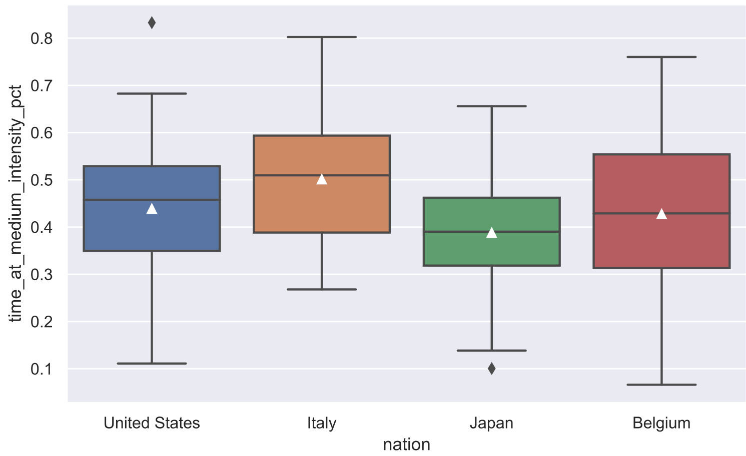


Figure 5.16: Boxplots for time_at_medium_intensity_pct experiment (nationality).

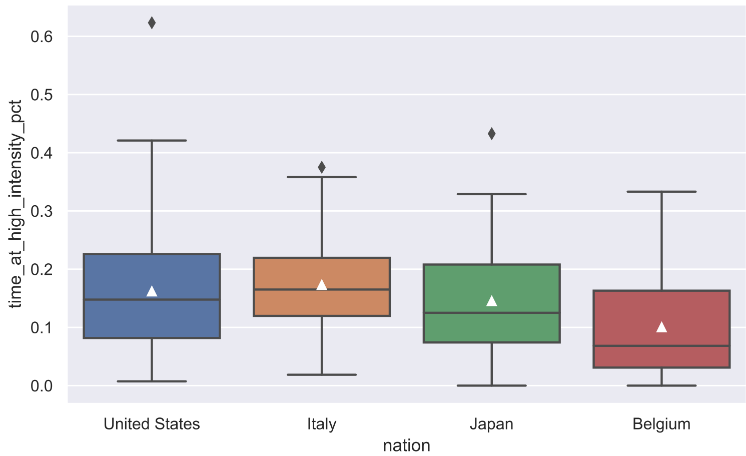


Figure 5.17: Boxplots for time_at_high_intensity_pct experiment (nationality).

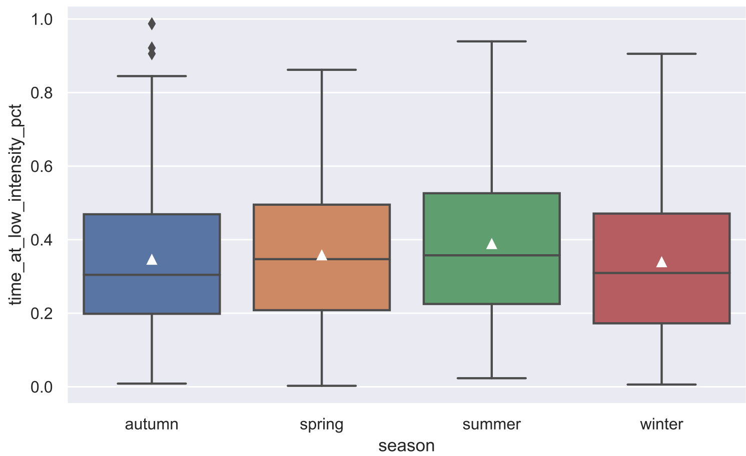


Figure 5.18: Boxplots for time_at_low_intensity_pct experiment (seasonality).

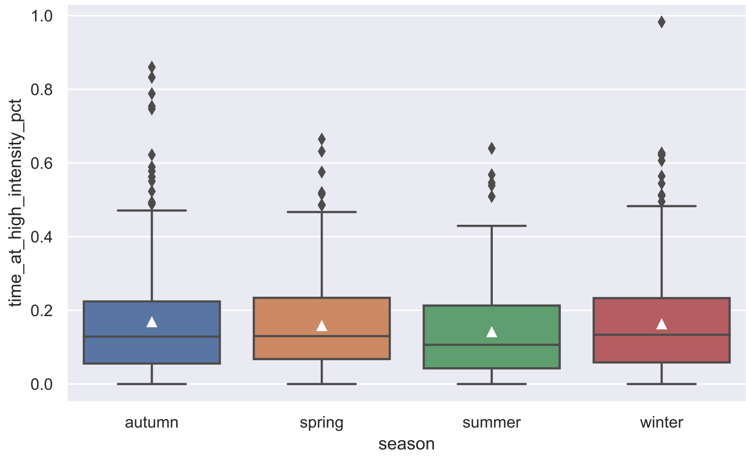


Figure 5.19: Boxplots for time_at_high_intensity_pct experiment (seasonality).

5.6 Training load and balancing

Week attribute	**Weeks excluded**	**Outliers removed**	**Grouping operation**	**Factor**
running_load_rtss	without runs, without HR, without LT	-	median	nation
avg_volume_acwr	without runs, without LT	-	median	nation
running_load_rtss	without runs, without HR, without LT	-	median	season
running_load_edwards	without runs, without HR	-	median	season
avg_volume_acwr	without runs, without LT	-	median	season
avg_volume_acwr	without runs, without LT	181 weeks	median	two-way

Table 5.6: Experiments configurations for training load and balancing analysis.

Total running load

In these experiments we tested for a difference in the mean total weekly training load using Edwards TRIMP and rTSS (TrainingPeaks) formulas for each run.
Using nation as factor (Fig. 5.20), the test with rTSS didn't find a statistically significant difference among the different countries means. Similarly, no possible separation was identified after the post-hoc analysis, although a slight higher mean could be observed for USA and Japan. Using the Edwards approach we obtained the same inconclusive results (not shown).

Concerning seasonality, we performed two separate tests using rTSS and Edwards TRIMP (Fig. 5.22, 5.23) in order to address the problem of weeks without a lactate threshold estimation (Edwards doesn't need it). In both cases ANOVA found a significant difference between the seasons with a low effect size. However, employing rTSS only summer was separated from the other seasons (minor load). In the other case instead, both summer and autumn showed a lower value w.r.t. spring.
When testing for the interaction of the two factors, no significant result was encountered.

Average volume ACWR

These experiments were based on the analysis of mean daily Acute:Chronic Workload Ratio index (see Section 2.2.4) observed in a typical training week. In order to generate this statistic, we assigned to each day of the week an ACWR value (computed using distance volume for training load) and subsequently extracted the mean for every week.
The ANOVA test results for nationality (Fig. 5.21) showed a statistically significant difference with a medium effect size between all the countries in analysis. More specifically, thanks to post-hoc testing, we were able to separate Japan (with a lower mean ACWR) from the rest of the nations.
Repeating the same experiment with season as factor (Fig. 5.24), the results showed a change among the groups with a low effect size. Furthermore, with post-hoc testing we were able to separate them in this sequence of increasing mean value: autumn → spring → winter (summer could not be differentiated from the last two). In this case, including weeks without LT produced no variation with respect to the separations observed previously (although autumn presented a lower mean).
In the two-way analysis we discovered a meaningful interaction between factors on the resulting means. Namely, the observations regarding USA runners suggested an higher total divergence of the ACWR value depending on the season of the year.

Comments

As far as total running load and nationality, only a slight difference between the groups USA/Japan and Italy/Belgium can be observed, although not statistically significant (the same inconclusive results were found using the Edwards TRIMP formula). However, this suggests that maybe with more data we would be able to see a distinction. On the other hand, the ACWR experiment highlights that the Japanese athletes are the most conservative ones when it comes to the fatigue/fitness balance, at least in therms of volumes.
Considering load and seasonality, we saw that the two distinct tests provided two fairly different results. In particular, by working with Edwards, autumn showed a significantly lower mean than with rTSS such that it was separated from spring (reaching values comparable to summer). We think that this was due to the fact that, since with rTSS we had to ignore weeks without a LT estimation, a lot of early autumn weeks were not taken in consideration, thus increasing the overall mean value of the season. Therefore, we presume that the Edwards result represents a more accurate picture of the real situation as it did not cause an imbalance between the seasons. If this was true, it would make sense to some extent, as we have seen that for many other aspects in these two seasons the runners seem to reduce their training overall. Concerning ACWR instead, the results found were rather interesting, as we observed some peculiar separations among the

seasons. Particularly, the reason why athletes in spring are more conservative is perhaps caused by the preparation undertaken for the eventual races, which requires some periods of rest in order to avoid accumulating fatigue. Likewise, the autumn results might be influenced by the difference in load we discovered with the other experiments.
Lastly, it's interesting to see the highly polarized strategy that is highlighted in the USA two-way case that, otherwise, we wouldn't have noticed analyzing the main factors only.

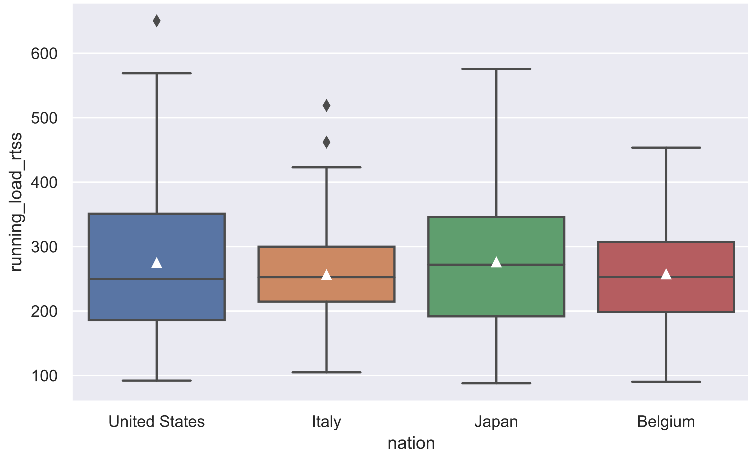


Figure 5.20: Boxplots for running_load_rtss experiment (nationality).

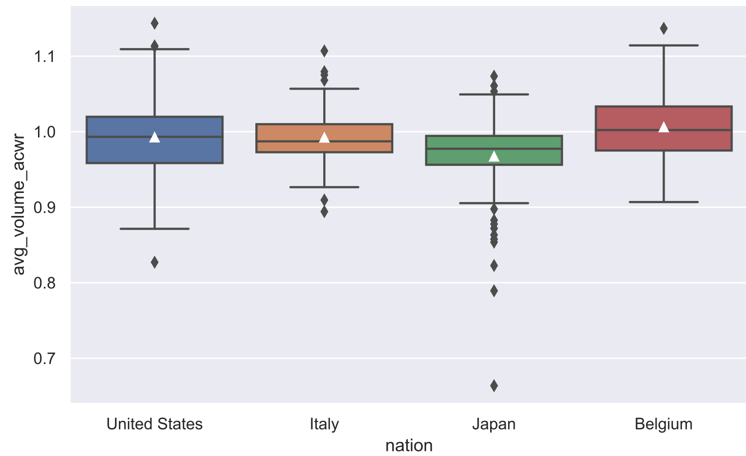


Figure 5.21: Boxplots for avg_volume_acwr experiment (nationality).

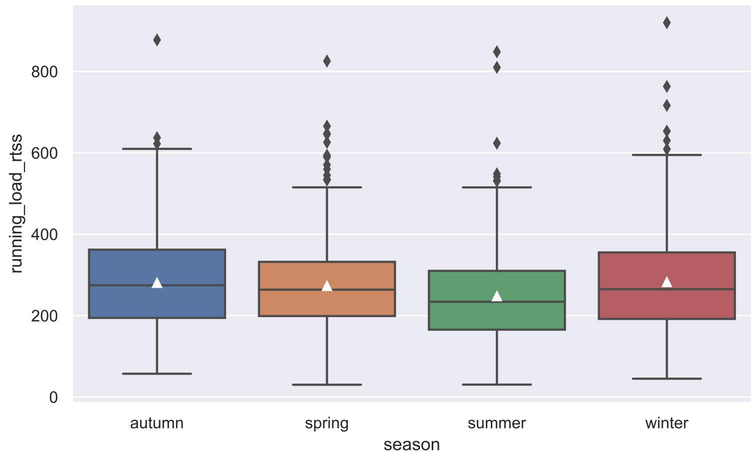


Figure 5.22: Boxplots for running_load_rtss experiment (seasonality).

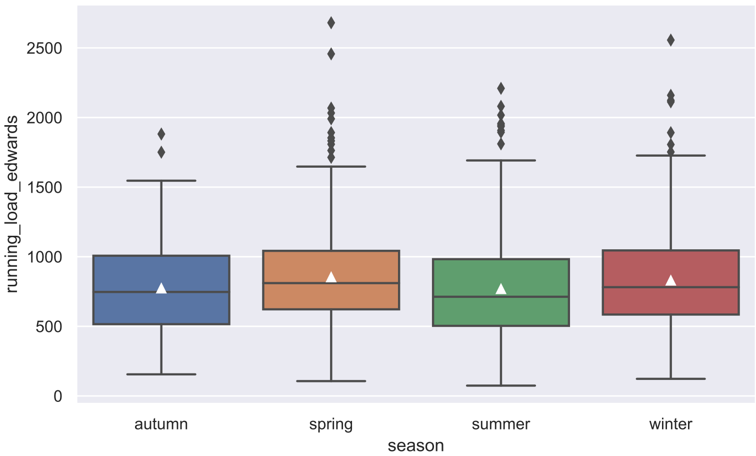


Figure 5.23: Boxplots for running_load_edwards experiment (seasonality).

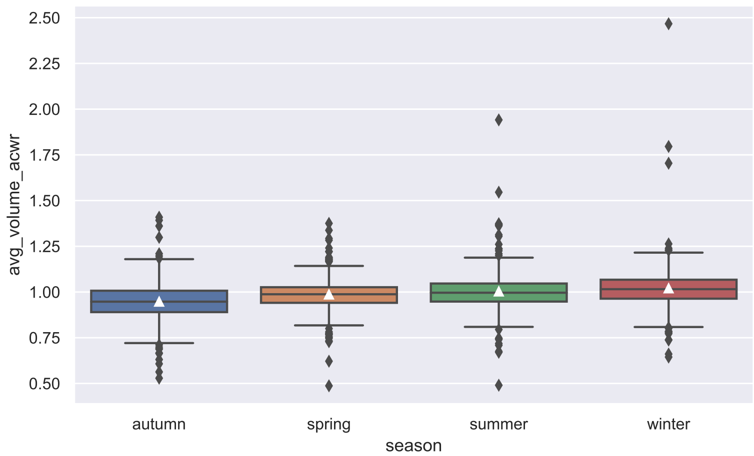


Figure 5.24: Boxplots for avg_volume_acwr experiment (seasonality).

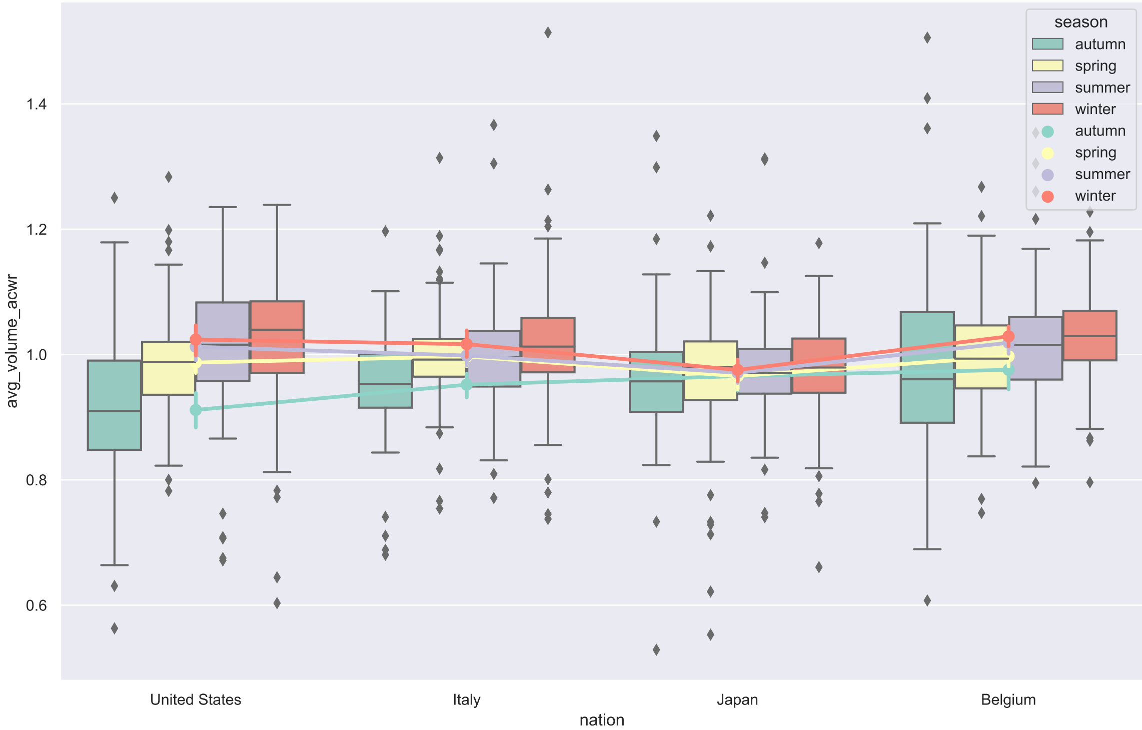


Figure 5.25: Boxplots for avg_volume_acwr experiment (nationality/seasonality).

5.7 Workout classes usage

Week attribute	**Weeks excluded**	**Outliers removed**	**Grouping operation**	**Factor**
pct_of_uniform_runs	without runs, without HR, without LT	-	mean	nation
pct_of_long_runs	without runs, without HR, without LT	-	mean	nation
pct_of_progression_runs	without runs, without HR, without LT	-	mean	nation
pct_of_peaks_runs	without runs, without HR, without LT	-	mean	nation
pct_of_recovery_runs	without runs, without HR, without LT	-	mean	nation
pct_of_uniform_runs	without runs, without LT	-	mean	season
pct_of_uniform_under_lt_runs	without runs, without LT	-	mean	season

Table 5.7: Experiments configurations for workout classes usage analysis.

Percentage of different workout classes

With the help of the following experiments we wanted to test for a difference in the usage of various workout classes inside a typical training week. Below we report the tests that returned interesting results, using nation and then season as factors.

The first test on nationality (Fig. 5.26) found a statistically significant difference (with a low effect size) between the mean percentages of uniform runs executed during a week by the different countries. In particular, the post-hoc tests placed Italy as the one with the lowest mean w.r.t. all others.
Next, when investigating long runs usage (Fig. 5.27), the ANOVA results discovered a significant difference among the groups with a medium effect size. After the post-hoc analysis, the following groups (in increasing order of percentage) were formed: Italy $\rightarrow$ USA $\rightarrow$ Japan. Meanwhile Belgium could not be separated from neither Italy nor USA.
Concerning progression runs (Fig. 5.28), the dedicated experiment successfully found a significant mean difference with a medium to high effect size. Following the post-hoc analysis, these increasing percentage groups were found: Belgium $\rightarrow$ USA $\rightarrow$ Italy. Whereas Japan, being between the highest two, could only be differentiated from Belgium.
The test results for runs containing intervals (Fig. 5.29) revealed a distinction between the nations with a medium effect size. In post-hoc testing, however, only the Belgian runners mean was considered significantly greater than the other countries.
Lastly, for recovery runs (Fig. 5.30) the null hypothesis was rejected as well with a low effect size. Specifically, after post-hoc analysis, USA and Japan runners were found having higher means than Italy and, in case of Japan, also of Belgium.

With seasonality the test involving uniform runs (Fig. 5.31) identified a statistically significant difference among the seasons with a low effect size. However, Post-hoc testing was only able to separate summer (higher percentage) from autumn.
Considering only uniform runs under the lactate threshold (Fig. 5.32), the difference was also significant with a low effect size. Similarly, the post-hoc analysis identified the same separation between the two previously mentioned seasons.

Of all the two-way tests performed no one determined the presence of a noteworthy interaction between nation and season on the usage of different workout classes.

Comments

Looking at the overall results obtained when analyzing the influence of nationality on the types of workout adopted, we can see that they seem to be coherent with most of the findings already discussed in the other categories of tests. Particularly, the Japanese runners appear to make use of a considerable amount of uniform runs that are often low intensity (e.g. recovery) and fairly long in distance. As a consequence, their percentage of weekly interval or progression runs is shown to be lower with respect to some other countries. On the contrary, athletes in Italy can be seen avoiding the most all sort of consistent runs in favor of medium to high intensity efforts such as progressions. In Belgium there seems to be a preference for uniform and interval runs as opposed to progressive exercises. Finally, the results on USA athletes suggest a varied workout classes usage with less homogeneity

than Japan and more time dedicated to other kinds of non-uniform runs.
When considering season as the independent variable, we were not able to find many interesting results. In fact, the only aspect that is somewhat evident confirms the popularity of uniform runs (especially low intensity) during the summer period. However, this difference is not well marked and, in particular, can assume a certain validity only against the autumn results. Maybe these findings suggest that no big variation exists over the course of the year for what concerns the types of workouts performed. Still, nothing much can be said about all the inconclusive outcomes.
In conclusion, the absence of notable interactions between the two factors analyzed doesn't surprise us much given what we just said about seasonality.

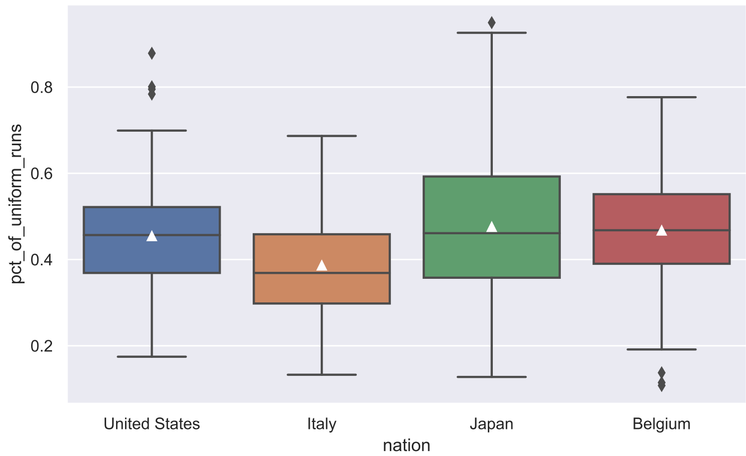


Figure 5.26: Boxplots for pct_of_uniform_runs experiment (nationality).

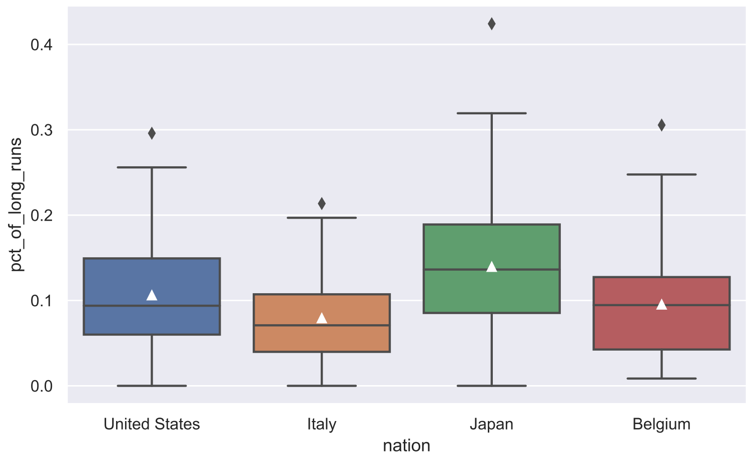


Figure 5.27: Boxplots for pct_of_long_runs experiment (nationality).

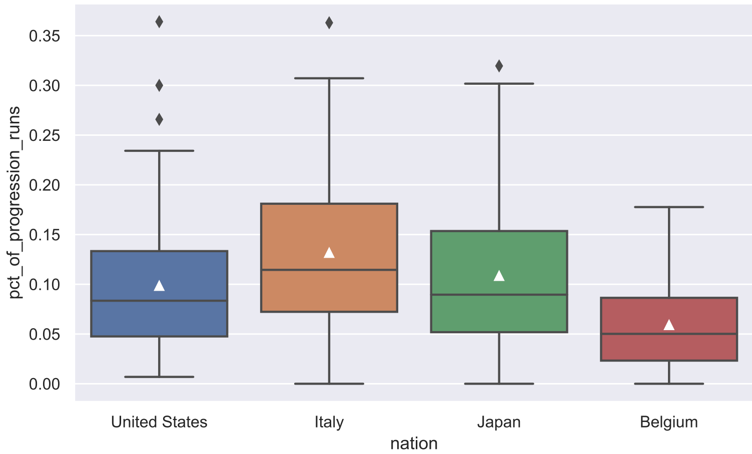


Figure 5.28: Boxplots for pct_of_progression_runs experiment (nationality).

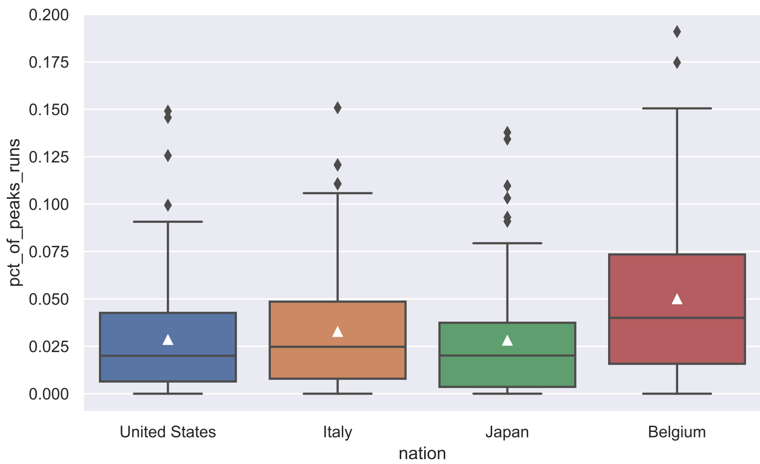


Figure 5.29: Boxplots for pct_of_peaks_runs experiment (nationality).

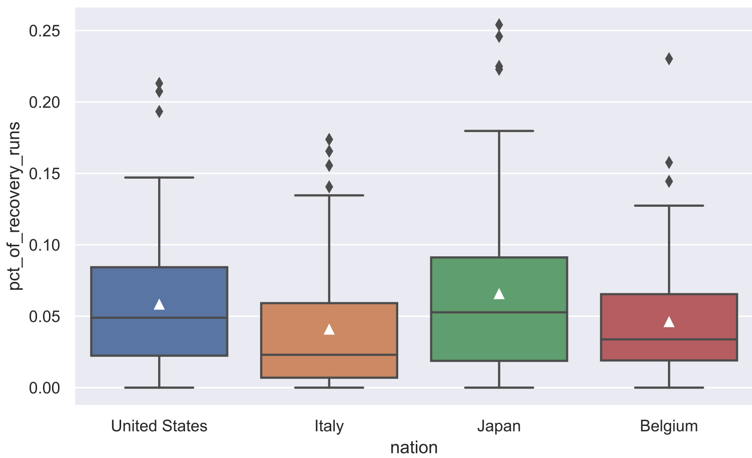


Figure 5.30: Boxplots for pct_of_recovery_runs experiment (nationality).

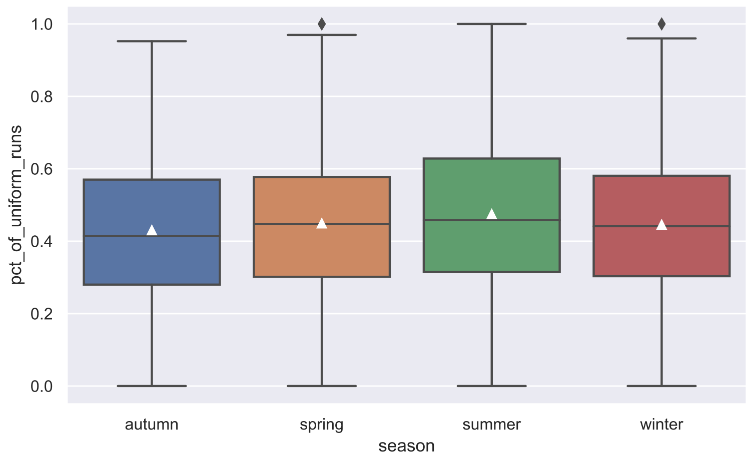


Figure 5.31: Boxplots for pct_of_uniform_runs experiment (seasonality).

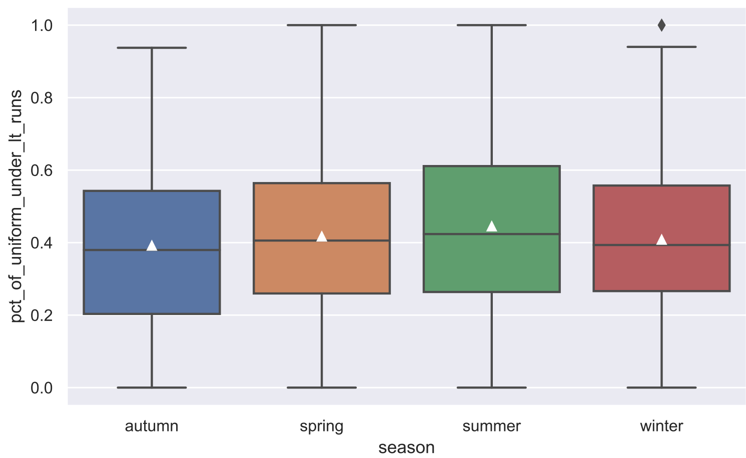


Figure 5.32: Boxplots for pct_of_uniform_under_lt_runs experiment (seasonality).

5.8 Main findings

Considering all the results obtained during the analysis, we can quite confidently say that some patterns exist in the training habits of the different groups of athletes considered. In particular, we have seen that the runner's nationality shows an higher impact compared to the seasonality of training during the year. In fact, for the first one we could clearly identify distinct strategies adopted by the countries that were also mutually coherent between the categories of experiments executed. For example, the Japanese athletes presented a typical routine composed of high weekly volumes spread across multiple sessions with a large amount of time spent at low intensities and, consequently, a considerable number of long and uniform runs (w.r.t. intervals for instance). Moreover, they we the ones with less cross-training and overall more consistency throughout the seasons. Conversely, the results on Italy suggested an higher usage of medium intensity runs (e.g. progressions) coupled with a good amount of hill running. Regarding Belgian runners instead, their training habits indicated high volume runs performed at low intensity but also a significant number of interval efforts, flat running and cross-training integration (especially during summer) compared to the other countries. Finally, considering USA we observed more weekly training sessions (but shorter) fairly balanced across the various categories together with an highly emphasized training load periodization (ACWR).
In terms of seasonality the most consistently relevant difference was related to summer. In fact, we saw that in this period the volumes appear to be reduced with a lower intensity and accumulated fatigue along with an higher integration of cross-training activities. In this context, autumn can be seen as a sort of transition phase towards the main training seasons comprising winter and spring, where the latter is often dedicated to the competitions preparation.

Chapter 6

Conclusions and Future Work

In this thesis, we performed an analysis on the training habits observed in a group of amateur endurance runners. In order to carry out this study, we had to first design an adequate framework to process all the data and consequently extract various features starting from the most basic activity information. In particular, our aim was to generate for each run an high level view showing the quality of the effort exerted in a totally personalized manner (based on the athlete performances). To implement it, we experimented with some strategies in order to estimate the physiological parameters of a given runner from data. Specifically, the assessment of HR_{max} and Anaerobic Threshold were at the center of our approach. For the latter, we designed a method built on the principles of the Conconi Test that, despite some limitations, proved to be rather successful. Having obtained an estimate of these performance parameters, we then moved to the actual creation of the high level activities. Notably, many efforts were spent aiming to achieve a totally autonomous classification of the running workouts based on the data streams of the single sessions. For this purpose, we developed a process that breaks up each activity in a sequence of logic blocks that are subsequently used for the final classes assignment. As the last part of the framework, we eventually tackled the problem concerning the physiological parameters evolution in time and the consequent generation of the high level activities calendar for every athlete.

After the architecture completion, we carried out a data collection process in preparation for the main experimental analysis. At this stage, we selected a number of amateur runners (collecting a year of their training activity) representing the countries of Italy, Japan, Belgium and USA. Following a cleaning and preparation of this data, we then executed a series of experiments designed to study the training habits present among these athletes. Particularly, for each runner we generated a selection of samples that, with the help of some custom functions, were processed in order to represent the various weekly training routines characteristics. Next, considering both nationality and seasonality, we conducted a number of statistical tests (ANOVA) in order to determine the single and combined influence of these factors on the training habits observed. By analyzing the experiments results we could identify some interesting distinctions in the individual categories of tests, especially regarding the nationality aspect. In fact, thanks to these experiments, we were able spot the specific training strategies adopted by each country such as the high-volume/low-intensity one of the Japanese athletes. With seasonality instead, we only managed to detect weak differences except for the summer season. In all

cases, however, the workout classes analysis was the most problematic one.
In conclusion, the fact that for the majority of the results we could find a plausible explanation and a some direct correlation with multiple tests, in our opinion shows that this kind of analysis is indeed feasible, even when using data that is not the outcome of a controlled study.

6.1 Future Work

Despite the encouraging results we obtained, there are many aspects of our approach that could be improved.
Starting from the processing framework itself, the estimation procedures related to the performance parameters highlighted several problems that may be partially solvable by exploring different strategies. For instance, the process for the evaluation of Anaerobic Threshold still struggles in the precise individuation of the breakpoints due to the noisy and inconsistent nature of the data. Therefore, it could probably use the help of more advanced and custom designed outlier detection methods or regression models to increase its robustness. Alternatively, a totally different approach could be focused on detecting the highest speeds consistently observed during training in windows of approximately one hour, thus directly estimating the functional threshold pace.
Concerning the classification of runs, additional work should be done to more precisely compute and utilize the activity blocks information provided. For example, the whole approach could be extended to the use of multiple streams (instead of only speed) together with an improved detection of the characterizing elements contained inside the runs. In particular, the identification of intervals showed an high false positive problem that forced us to be more conservative and maybe discard some real peaks present in the data. On this matter, some approaches based on time series pattern extraction should maybe explored in order to also possibly find more sophisticated features inside the runs. Additionally, the blocks resulting from the classification process could be used to perform a clustering of the activities to aid with the final class labeling.
With respect to the experimental analysis itself, more data and wider time periods should be investigated to support our findings. Moreover, different methodologies could be adopted to search for other interesting patterns in amateur runners such as the ones related to performance or injury prevention.

Bibliography

[1] R. A. Armstrong, S. V. Slade, and F. Eperjesi. "An introduction to analysis of variance (ANOVA) with special reference to data from clinical experiments in optometry". In: *Ophthalmic and Physiological Optics* 20 (3 2000), pp. 235–241.

[2] E. W. Banister and T. W. Calvert. "Planning for future performance: implications for long term training". In: *Canadian journal of applied sport sciences. Journal canadien des sciences appliquees au sport* 5.3 (Sept. 1980), pp. 170–176.

[3] Tudor Bompa and G. Gregory Haff. *Periodization: Theory and Methodology of Training*. 5th ed. Human Kinetics, 2009, pp. 13–15.

[4] Gunnar A.V. Borg. "Psychophysical bases of perceived exertion". In: *Medicine and Science in Sports and Exercise* 14 (5 1982), pp. 377–381.

[5] Thomas W. Calvert et al. "A Systems Model of the Effects of Training on Physical Performance". In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-6.2 (1976), pp. 94–102.

[6] F. Conconi et al. "Determination of the anaerobic threshold by a noninvasive field test in runners". In: *Journal of Applied Physiology* 52.4 (1982), pp. 869–873.

[7] F. Conconi et al. "The Conconi Test: Methodology After 12 Years of Application". In: *International Journal of Sports Medicine* 17 (7 1996), pp. 509–519.

[8] I Cook. "Was the Conconi test validated by sporting success, expert opinion or good science?" In: *South African Journal for Research in Sport, Physical Education and Recreation* 33.1 (Apr. 2011).

[9] Jack Daniels. *Daniels' running formula*. Third edition. Human Kinetics, 2014.

[10] Jack Daniels and Jimmy Gilbert. *Oxygen power: performance tables for distance runners*. J. Daniels, J. Gilbert, 1979.

[11] Talko Bernhard Dijkhuis et al. "Increase in the Acute:Chronic Workload Ratio relates to Injury Risk in Competitive Runners". In: *International Journal of Sports Medicine* (2020).

[12] Drew Robb. *An Improved GAP Model*. 2017. URL: `https://medium.com/strava-engineering/an-improved-gap-model-8b07ae8886c3`.

[13] Sally Edwards. *The heart rate monitor book*. Polar CIC, 1993.

[14] Martin Ester et al. "A density-based algorithm for discovering clusters in large spatial databases with noise". In: (1996), pp. 226–231.

[15] Matt Fitzgerald. *80/20 running: run stronger and race faster by training slower*. Penguin Books, 2015.

[16] Carl Foster et al. “Effects of specific versus cross-training on running performance”. In: *European Journal of Applied Physiology* 70 (4 1995), pp. 367–372.

[17] Tim J. Gabbett. “The training-injury prevention paradox: should athletes be training smarterandharder?” In: *British Journal of Sports Medicine* 50.5 (Jan. 2016), pp. 273–280.

[18] Jan Hnizdil et al. “The Conconi Test – Searching for the Deflection Point”. In: *Physical Activity Review* 7 (2019), pp. 160–167.

[19] Billy T. Hulin et al. “The acute:chronic workload ratio predicts injury: high chronic workload may decrease injury risk in elite rugby league players”. In: *British Journal of Sports Medicine* (2015), bjsports-2015–094817.

[20] Charles F Jekel and Gerhard Venter. “PWLF: a python library for fitting 1D continuous piecewise linear functions”. In: (2019).

[21] D. E. Lieberman et al. “Effects of stride frequency and foot position at landing on braking force, hip torque, impact peak force and the metabolic cost of running in humans”. In: *Journal of Experimental Biology* 218 (21 2015), pp. 3406–3414.

[22] Alejandro Lucia et al. “Tour de France versus Vuelta a Espana: Which Is Harder?” In: *Medicine and Science in Sports and Exercise* 35 (5 2003), pp. 872–878.

[23] Steve Magness. *The science of running: how to find your limit and train to maximize your performance*. Origin Press, 2014.

[24] Vincenzo Manzi et al. “Relation between Individualized Training Impulses and Performance in Distance Runners”. In: *Medicine & Science in Sports & Exercise* 41.11 (Nov. 2009), pp. 2090–2096.

[25] Adrian W. Midgley, Lars R. McNaughton, and Michael Wilkinson. “Is there an Optimal Training Intensity for Enhancing the Maximal Oxygen Uptake of Distance Runners?” In: *Sports Medicine* 36 (2 2006), pp. 117–132.

[26] Alberto E. Minetti et al. “Energy cost of walking and running at extreme uphill and downhill slopes”. In: *Journal of Applied Physiology* 93 (3 2002), pp. 1039–1046.

[27] Philip F. Skiba. *Calculation of Power Output and Quantification of Training Stress in Distance Runners: The Development of the GOVSS Algorithm*. 2006. URL: `http://runscribe.com/wp-content/uploads/power/GOVSS.pdf`.

[28] D. Shookster et al. “Accuracy of Commonly Used Age-Predicted Maximal Heart Rate Equations”. In: *Int J Exerc Sci* 13.7 (2020), pp. 1242–1250.

[29] Karl M. Stagno, Rhys Thatcher, and Ken A. van Someren. “A modified TRIMP to quantify the in-season training load of team sport players”. In: *Journal of Sports Sciences* 25 (6 2007), pp. 629–634.

[30] Stephen McGregor. *Running Training Stress Score (rTSS*) Explained*. URL: `https://www.trainingpeaks.com/learn/articles/running-training-stress-score-rtss-explained/`.

[31] Stephen McGregor. *What is Normalized Graded Pace?* URL: https://www.trainingpeaks.com/learn/articles/what-is-normalized-graded-pace/.

[32] Will Meyer and Chris Spada. *Quantifying Effort through Heart Rate Data*. 2018. URL: https://medium.com/strava-engineering/quantifying-effort-through-heart-rate-data-e6a0e3dd6a52.

[33] Sean Williams et al. "Better way to determine the acute:chronic workload ratio?" In: *British Journal of Sports Medicine* 51 (3 2016), pp. 209–210.

Appendix A

Experiments Additional Output

A.1 Nationality

nation	N	Mean	SD	Interval
Belgium	99	3.1381	0.7251	3.2827
Italy	98	3.1453	0.6815	3.2819
Japan	101	3.4313	0.9405	3.617
United States	100	3.7332	0.9927	3.9301

test	statistic	p-value
Levene's Test:	4.401687	0.004626
Bartlett's Test:	20.070951	0.000164

Figure A.1: Samples and variances assumption check for n_of_days_with_runs experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.977808	9e-06

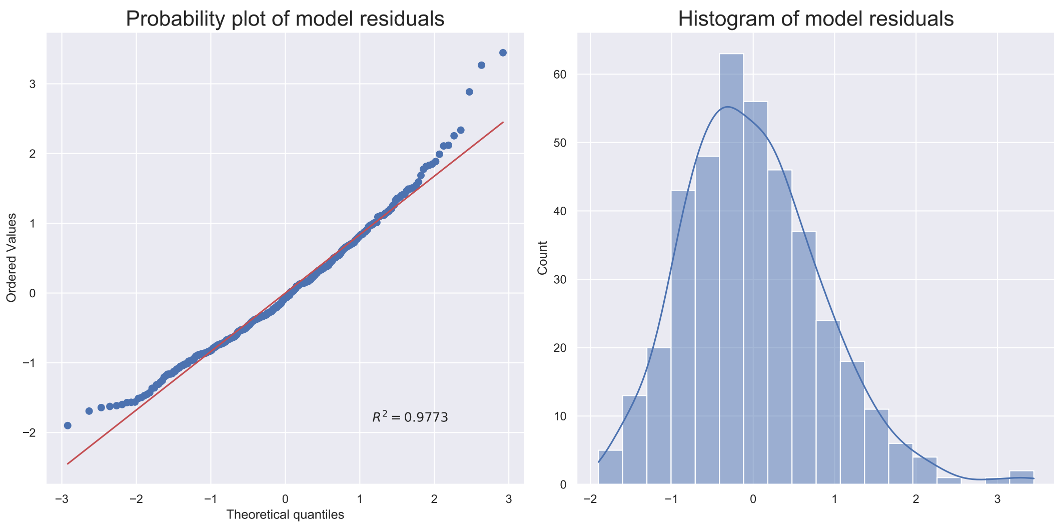


Figure A.2: Normality assumption check for n_of_days_with_runs experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(nation)	23.8279	3.0	7.9426	11.0741	1e-06	0.0778	0.0706
Residual	282.5868	394.0	0.7172	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
Belgium	Italy	0.0072	0.9	-0.3042	0.3185	False
Belgium	Japan	0.2933	0.07	-0.0158	0.6023	False
Belgium	United States	0.5951	0.001	0.2853	0.9049	True
Italy	Japan	0.2861	0.0823	-0.0237	0.5959	False
Italy	United States	0.5879	0.001	0.2773	0.8985	True
Japan	United States	0.3018	0.0575	-0.0064	0.6101	False

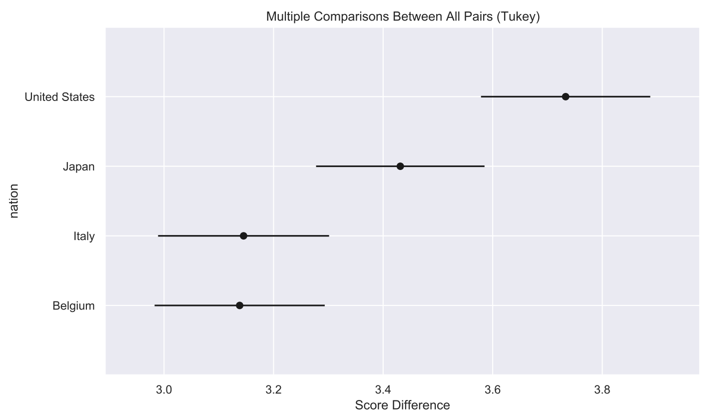


Figure A.3: ANOVA results for n_of_days_with_runs experiment.

nation	N	Mean	SD	Interval
Belgium	99	1.0713	0.0822	1.0877
Italy	98	1.0973	0.1397	1.1253
Japan	100	1.1316	0.1374	1.1589
United States	100	1.1138	0.1057	1.1347

test	statistic	p-value
Levene's Test:	3.755203	0.011088
Bartlett's Test:	33.818405	0.0

Figure A.4: Samples and variances assumption check for avg_daily_runs experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.79985	0.0

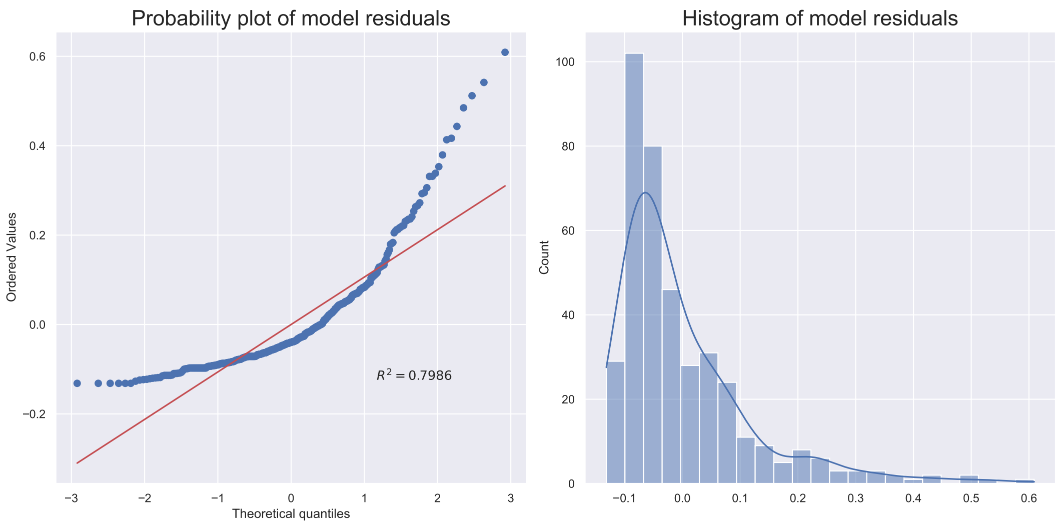


Figure A.5: Normality assumption check for avg_daily_runs experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(nation)	0.1959	3.0	0.0653	4.6401	0.003348	0.0342	0.0268
Residual	5.5301	393.0	0.0141	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
Belgium	Italy	0.026	0.4176	-0.0176	0.0696	False
Belgium	Japan	0.0603	0.0021	0.0169	0.1037	True
Belgium	United States	0.0424	0.0579	-0.001	0.0858	False
Italy	Japan	0.0343	0.1769	-0.0092	0.0778	False
Italy	United States	0.0165	0.7377	-0.027	0.06	False
Japan	United States	-0.0179	0.688	-0.0611	0.0254	False

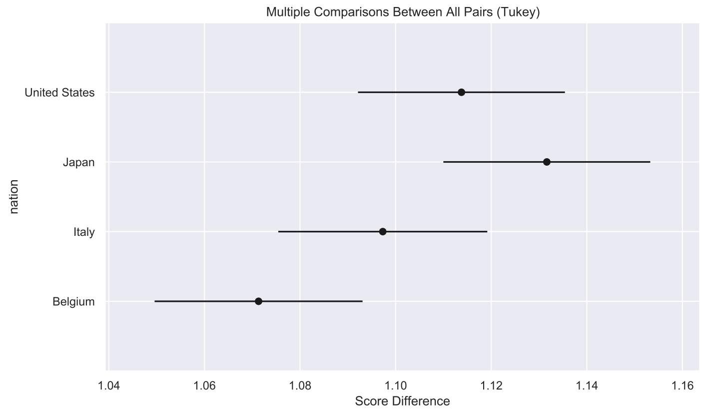


Figure A.6: ANOVA results for avg_daily_runs experiment.

nation	N	Mean	SD	Interval
Belgium	99	3.4019	0.8815	3.5777
Italy	98	3.5125	1.0139	3.7157
Japan	101	3.9703	1.2904	4.225
United States	100	4.1958	1.2307	4.44

test	statistic	p-value
Levene’s Test:	4.743096	0.002911
Bartlett's Test:	17.65232	0.000519

Figure A.7: Samples and variances assumption check for n_of_running_workouts experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.961952	0.0

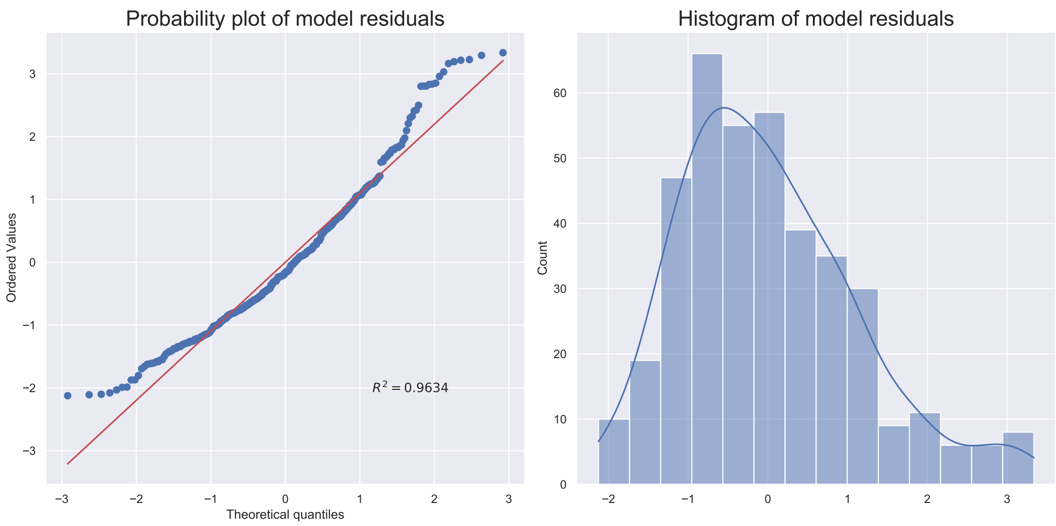


Figure A.8: Normality assumption check for n_of_running_workouts experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(nation)	42.0904	3.0	14.0301	11.2283	4e-07	0.0788	0.0716
Residual	492.3152	394.0	1.2495	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
Belgium	Italy	0.1105	0.8957	-0.3004	0.5215	False
Belgium	Japan	0.5684	0.0021	0.1605	0.9763	True
Belgium	United States	0.7939	0.001	0.385	1.2028	True
Italy	Japan	0.4578	0.0212	0.0489	0.8668	True
Italy	United States	0.6833	0.001	0.2734	1.0933	True
Japan	United States	0.2255	0.4821	-0.1814	0.6323	False

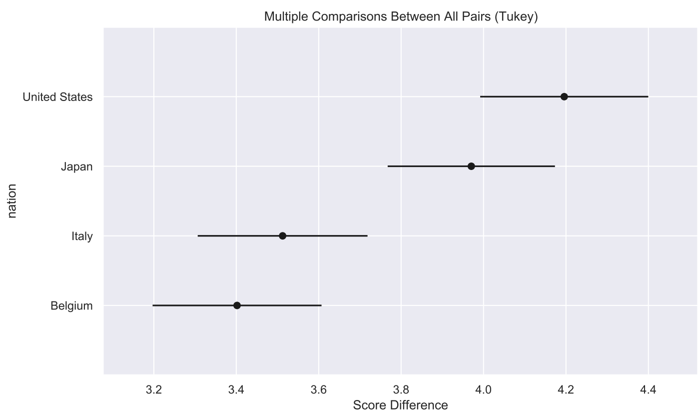


Figure A.9: ANOVA results for n_of_running_workouts experiment.

nation	N	Mean	SD	Interval
Belgium	99	12849.741	3666.2205	13580.9557
Italy	98	12266.4598	3783.5516	13025.0139
Japan	101	14502.3293	5404.9071	15569.3257
United States	100	13968.8101	5138.2289	14988.3462

test	statistic	p-value
Levene’s Test:	6.749505	0.000189
Bartlett's Test:	23.546762	3.1e-05

Figure A.10: Samples and variances assumption check for running_time experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.973358	1e-06

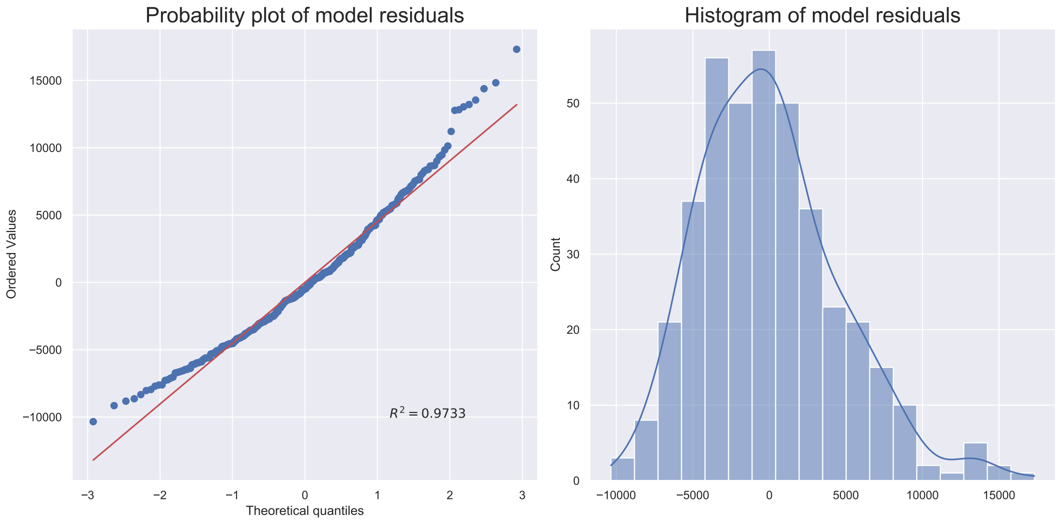


Figure A.11: Normality assumption check for running_time experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(nation)	310962145.3	3.0	103654048.4	4.9558	0.00218	0.0364	0.029
Residual	8240855642.4	394.0	20915877.3	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
Belgium	Italy	-583.2812	0.783	-2264.7067	1098.1443	False
Belgium	Japan	1652.5883	0.0534	-16.2408	3321.4174	False
Belgium	United States	1119.0691	0.3117	-553.8852	2792.0234	False
Italy	Japan	2235.8695	0.0035	562.7462	3908.9928	True
Italy	United States	1702.3503	0.0452	25.1123	3379.5883	True
Japan	United States	-533.5192	0.8212	-2198.1291	1131.0908	False

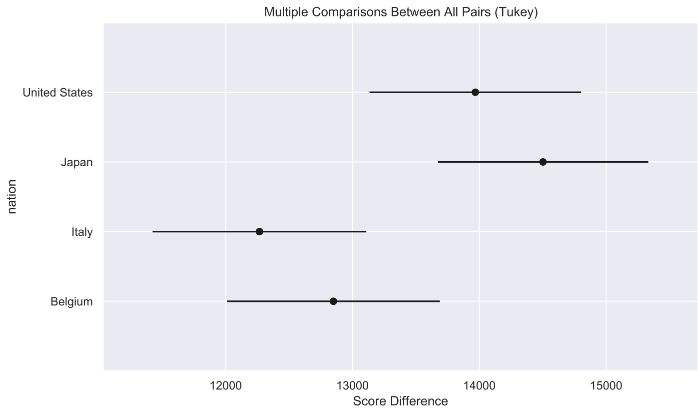


Figure A.12: ANOVA results for running_time experiment.

nation	N	Mean	SD	Interval
Belgium	99	3835.3002	857.5887	4006.3432
Italy	98	3623.695	962.9424	3816.7527
Japan	101	3822.8262	1248.2466	4069.2458
United States	100	3428.8531	1165.6237	3660.1381

test	statistic	p-value
Levene’s Test:	4.06793	0.007268
Bartlett's Test:	17.07727	0.000681

Figure A.13: Samples and variances assumption check for avg_run_duration experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.928481	0.0

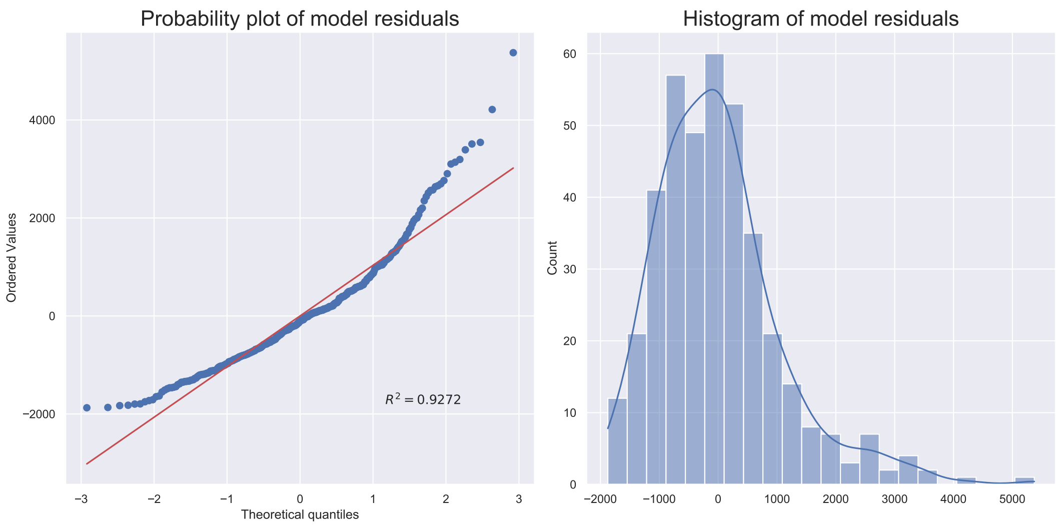


Figure A.14: Normality assumption check for avg_run_duration experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(nation)	11064448.2	3.0	3688149.4	3.2125	0.0229778	0.0239	0.0164
Residual	452340102.5	394.0	1148071.3	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
Belgium	Italy	-211.6052	0.5078	-605.5397	182.3293	False
Belgium	Japan	-12.4739	0.9	-403.4573	378.5094	False
Belgium	United States	-406.4471	0.0388	-798.3969	-14.4972	True
Italy	Japan	199.1312	0.55	-192.8582	591.1207	False
Italy	United States	-194.8419	0.5676	-587.7953	198.1116	False
Japan	United States	-393.9731	0.0467	-783.968	-3.9783	True

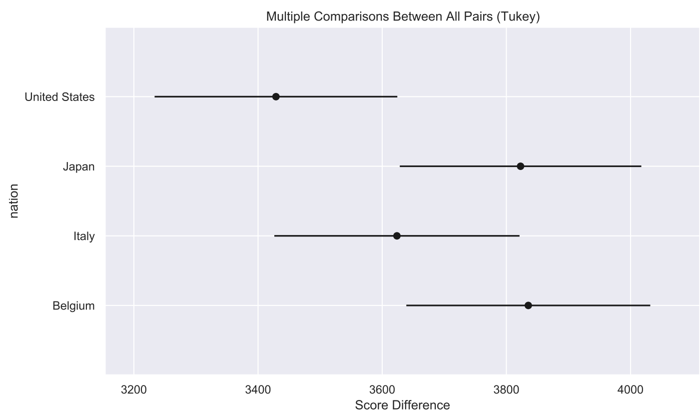


Figure A.15: ANOVA results for avg_run_duration experiment.

nation	N	Mean	SD	Interval
Belgium	99	37903.8855	11900.7983	40277.4579
Italy	98	36306.1892	11281.3211	38567.9511
Japan	101	41417.4008	15379.6128	44453.5293
United States	100	39369.5195	15741.2341	42492.9219

test	statistic	p-value
Levene's Test:	4.932163	0.002251
Bartlett's Test:	17.015218	0.000702

Figure A.16: Samples and variances assumption check for running_volume experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.968769	0.0

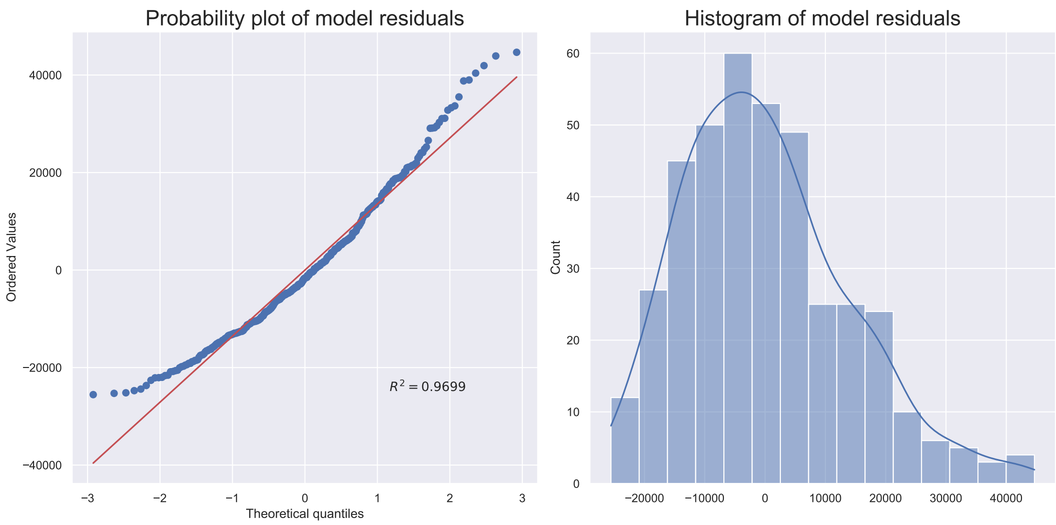


Figure A.17: Normality assumption check for running_volume experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(nation)	1412985056.6	3.0	470995018.9	2.494	0.059621	0.0186	0.0111
Residual	74408765535.8	394.0	188854734.9	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
Belgium	Italy	-1597.6963	0.8274	-6650.1637	3454.7711	False
Belgium	Japan	3513.5153	0.2713	-1501.1015	8528.1321	False
Belgium	United States	1465.634	0.8631	-3561.3787	6492.6467	False
Italy	Japan	5111.2116	0.0446	83.6911	10138.7321	True
Italy	United States	3063.3303	0.3993	-1976.5544	8103.215	False
Japan	United States	-2047.8813	0.6926	-7049.8202	2954.0576	False

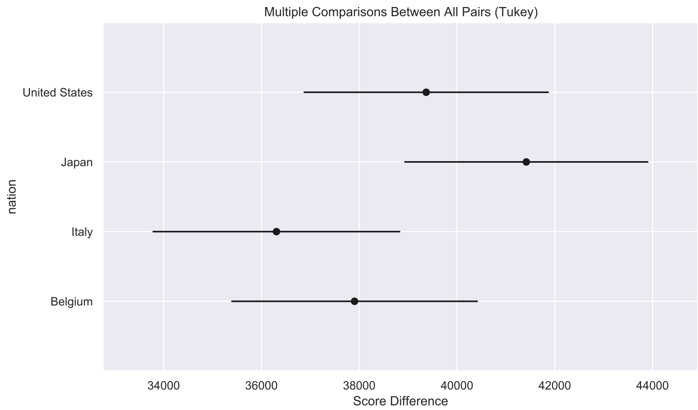


Figure A.18: ANOVA results for running_volume experiment.

nation	N	Mean	SD	Interval
Belgium	99	11145.7147	2163.401	11577.1974
Italy	98	10618.0276	2432.4528	11105.7035
Japan	101	10656.196	2705.245	11190.2453
United States	100	9331.3768	2462.7766	9820.0451

test	statistic	p-value
Levene's Test:	1.681086	0.170513
Bartlett's Test:	4.892268	0.179858

Figure A.19: Samples and variances assumption check for avg_run_volume experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.988307	0.002799

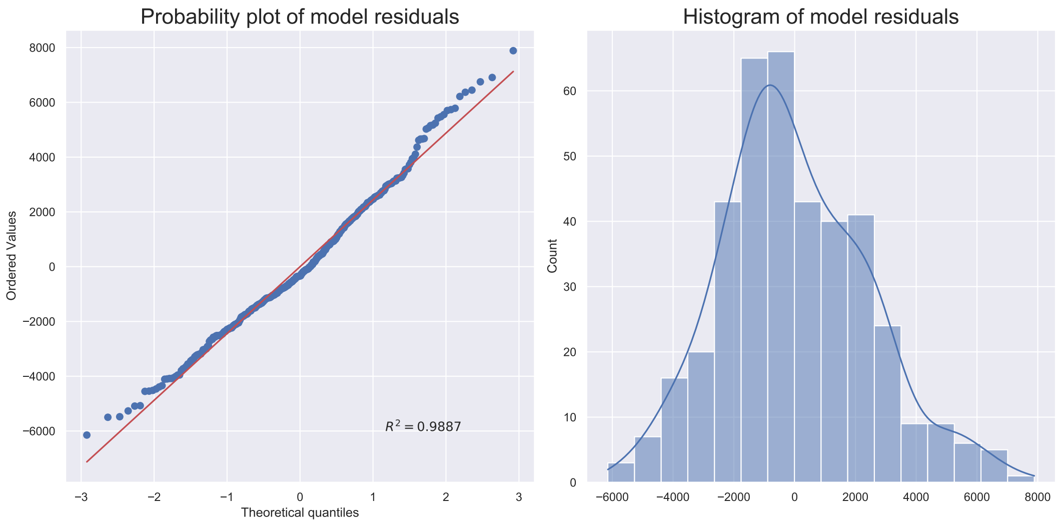


Figure A.20: Normality assumption check for avg_run_volume experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(nation)	180029243.6	3.0	60009747.9	9.9978	2.3e-06	0.0707	0.0635
Residual	2364898602.3	394.0	6002280.7	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
Belgium	Italy	-527.6871	0.433	-1428.4236	373.0493	False
Belgium	Japan	-489.5187	0.4923	-1383.5072	404.4699	False
Belgium	United States	-1814.3378	0.001	-2710.5363	-918.1394	True
Italy	Japan	38.1685	0.9	-858.1205	934.4575	False
Italy	United States	-1286.6507	0.0014	-2185.144	-388.1575	True
Japan	United States	-1324.8192	0.001	-2216.5476	-433.0908	True

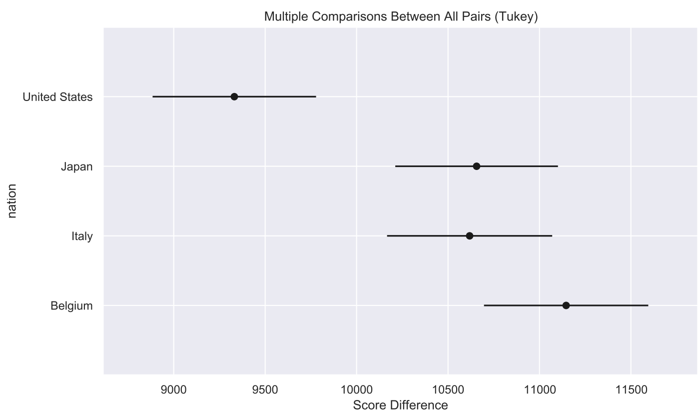


Figure A.21: ANOVA results for avg_run_volume experiment.

nation	N	Mean	SD	Interval
Belgium	99	0.9174	0.0722	0.9318
Italy	98	0.9279	0.0719	0.9423
Japan	101	0.9632	0.0574	0.9745
United States	100	0.9369	0.0706	0.9509

test	statistic	p-value
Levene's Test:	4.367814	0.004843
Bartlett's Test:	6.736469	0.080789

Figure A.22: Samples and variances assumption check for pct_time_running_vs_cross experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.878272	0.0

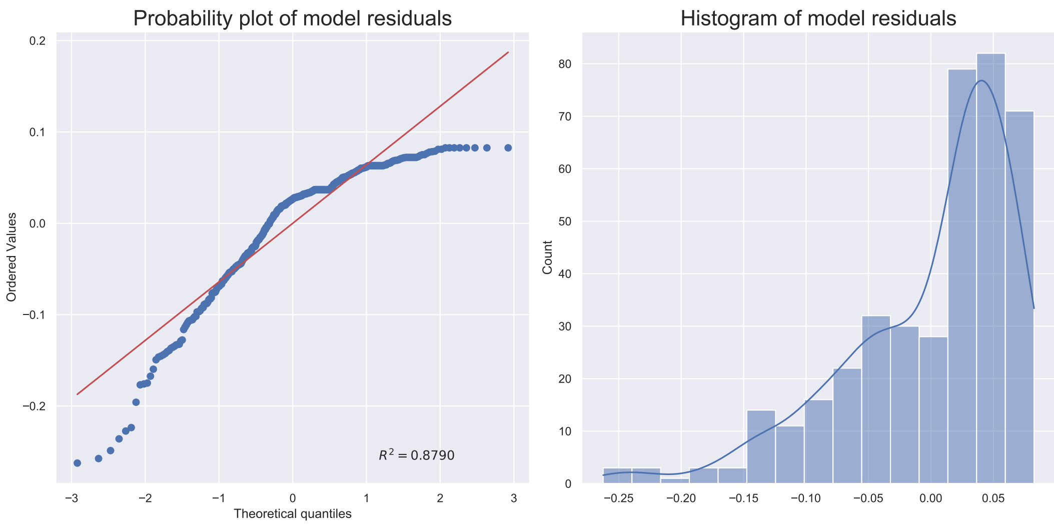


Figure A.23: Normality assumption check for pct_time_running_vs_cross experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(nation)	0.1153	3.0	0.0384	8.2492	2.46e-05	0.0591	0.0518
Residual	1.8353	394.0	0.0047	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
Belgium	Italy	0.0105	0.6799	-0.0146	0.0356	False
Belgium	Japan	0.0458	0.001	0.0209	0.0707	True
Belgium	United States	0.0195	0.1849	-0.0055	0.0444	False
Italy	Japan	0.0353	0.0017	0.0103	0.0603	True
Italy	United States	0.009	0.7656	-0.016	0.034	False
Japan	United States	-0.0263	0.0332	-0.0511	-0.0015	True

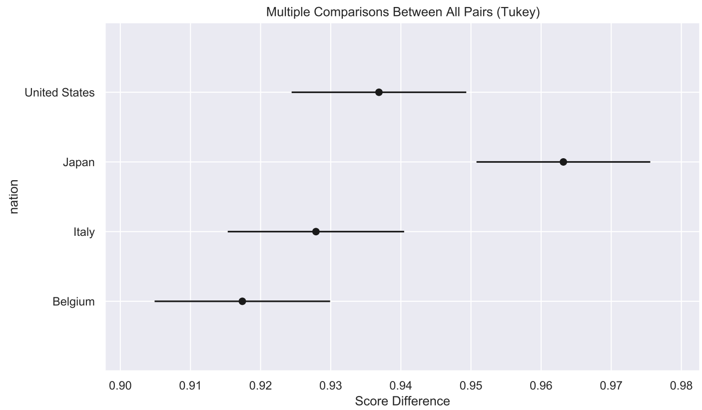


Figure A.24: ANOVA results for pct_time_running_vs_cross experiment.

nation	N	Mean	SD	Interval
Belgium	99	252.1313	237.1742	299.4349
Italy	98	415.0378	450.4669	505.3507
Japan	101	589.0327	403.0957	668.6088
United States	100	362.591	333.4454	428.7538

test	statistic	p-value
Levene’s Test:	5.678528	0.000815
Bartlett's Test:	41.263288	0.0

Figure A.25: Samples and variances assumption check for elevation_gained experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.802818	0.0

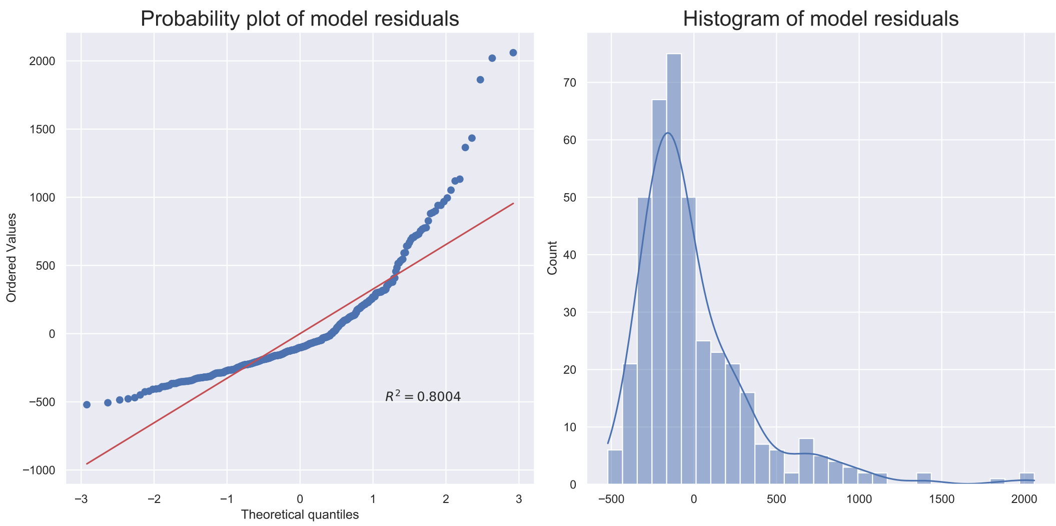


Figure A.26: Normality assumption check for elevation_gained experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(nation)	5923815.06	3.0	1974605.02	14.8325	4e-09	0.1015	0.0944
Residual	52451951.7	394.0	133126.78	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
Belgium	Italy	162.9064	0.01	28.7621	297.0508	True
Belgium	Japan	336.9014	0.001	203.762	470.0407	True
Belgium	United States	110.4597	0.1437	-23.0088	243.9282	False
Italy	Japan	173.9949	0.0047	40.5129	307.4769	True
Italy	United States	-52.4468	0.7179	-186.257	81.3635	False
Japan	United States	-226.4417	0.001	-359.2445	-93.6389	True

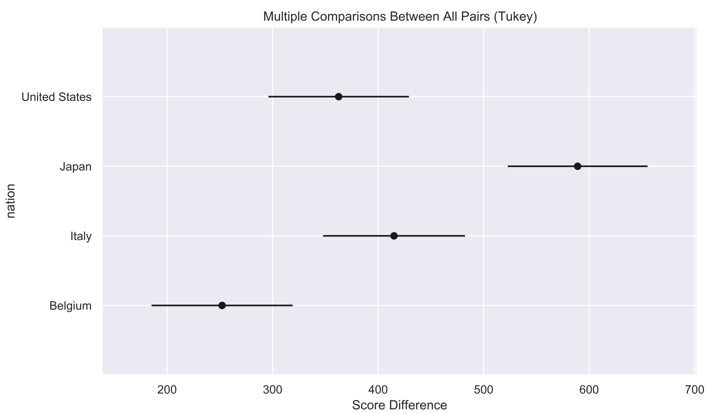


Figure A.27: ANOVA results for elevation_gained experiment.

nation	N	Mean	SD	Interval
Belgium	99	0.066	0.1449	0.0949
Italy	98	0.2937	0.6798	0.43
Japan	101	0.1515	0.3223	0.2152
United States	100	0.1123	0.2447	0.1608

test	statistic	p-value
Levene's Test:	6.90004	0.000154
Bartlett's Test:	236.603177	0.0

Figure A.28: Samples and variances assumption check for avg_incline_grade experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.568682	0.0

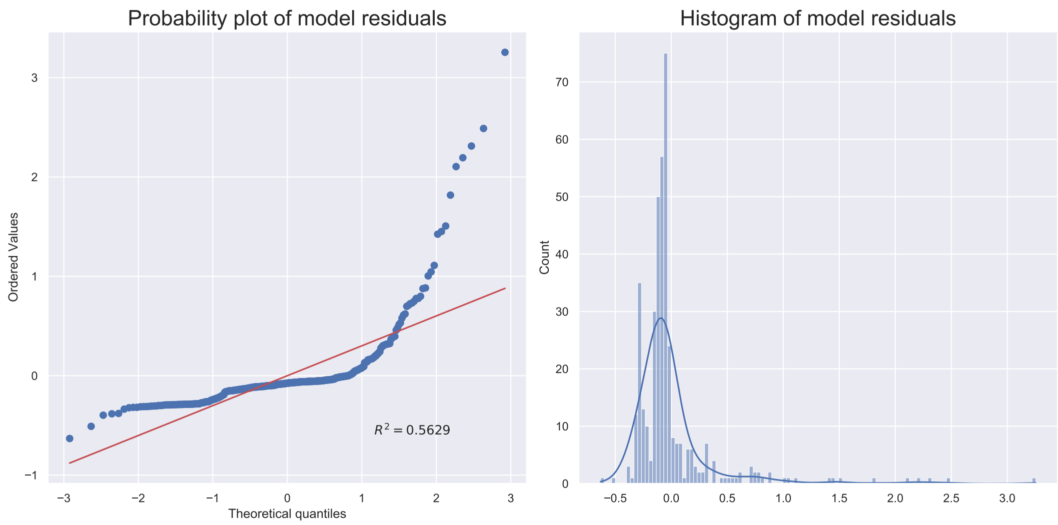


Figure A.29: Normality assumption check for avg_incline_grade experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(nation)	2.86	3.0	0.95	5.9327	0.0005766	0.0432	0.0358
Residual	63.21	394.0	0.16	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
Belgium	Italy	0.2278	0.001	0.0805	0.375	True
Belgium	Japan	0.0856	0.4333	-0.0606	0.2317	False
Belgium	United States	0.0463	0.8277	-0.1002	0.1928	False
Italy	Japan	-0.1422	0.0609	-0.2887	0.0043	False
Italy	United States	-0.1815	0.0084	-0.3284	-0.0346	True
Japan	United States	-0.0393	0.895	-0.1851	0.1065	False

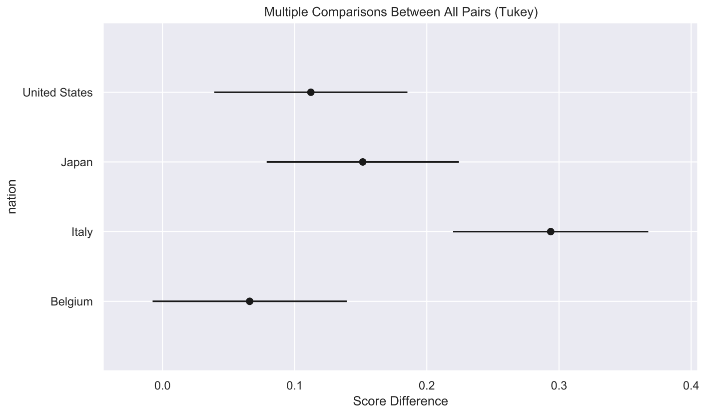


Figure A.30: ANOVA results for avg_incline_grade experiment.

nation	N	Mean	SD	Interval
Belgium	95	0.4016	0.2061	0.4436
Italy	94	0.2591	0.1208	0.2838
Japan	98	0.3948	0.1571	0.4263
United States	99	0.3296	0.1606	0.3617

test	statistic	p-value
Levene's Test:	8.744133	1.3e-05
Bartlett's Test:	26.110171	9e-06

Figure A.31: Samples and variances assumption check for time_at_low_intensity_pct experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.983248	0.000189

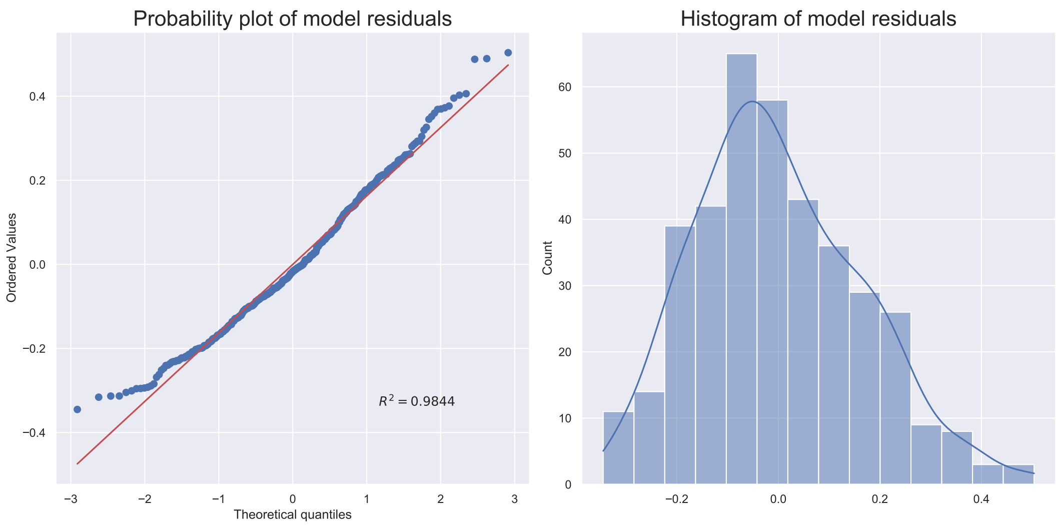


Figure A.32: Normality assumption check for time_at_low_intensity_pct experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(nation)	1.264	3.0	0.4213	15.6689	1e-09	0.1096	0.1023
Residual	10.2722	382.0	0.0269	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
Belgium	Italy	-0.1425	0.001	-0.2041	-0.081	True
Belgium	Japan	-0.0068	0.9	-0.0677	0.0541	False
Belgium	United States	-0.072	0.0128	-0.1328	-0.0112	True
Italy	Japan	0.1357	0.001	0.0747	0.1968	True
Italy	United States	0.0706	0.0158	0.0096	0.1315	True
Japan	United States	-0.0652	0.0282	-0.1255	-0.0049	True

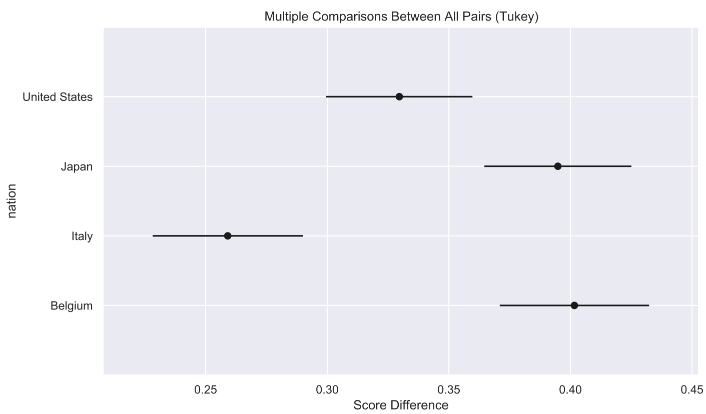


Figure A.33: ANOVA results for time_at_low_intensity_pct experiment.

nation	N	Mean	SD	Interval
Belgium	95	0.4275	0.1552	0.4591
Italy	94	0.5011	0.129	0.5275
Japan	98	0.3881	0.1105	0.4102
United States	99	0.4388	0.1266	0.4641

test	statistic	p-value
Levene's Test:	4.805902	0.002682
Bartlett's Test:	11.290758	0.010253

Figure A.34: Samples and variances assumption check for time_at_medium_intensity_pct experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.995218	0.281751

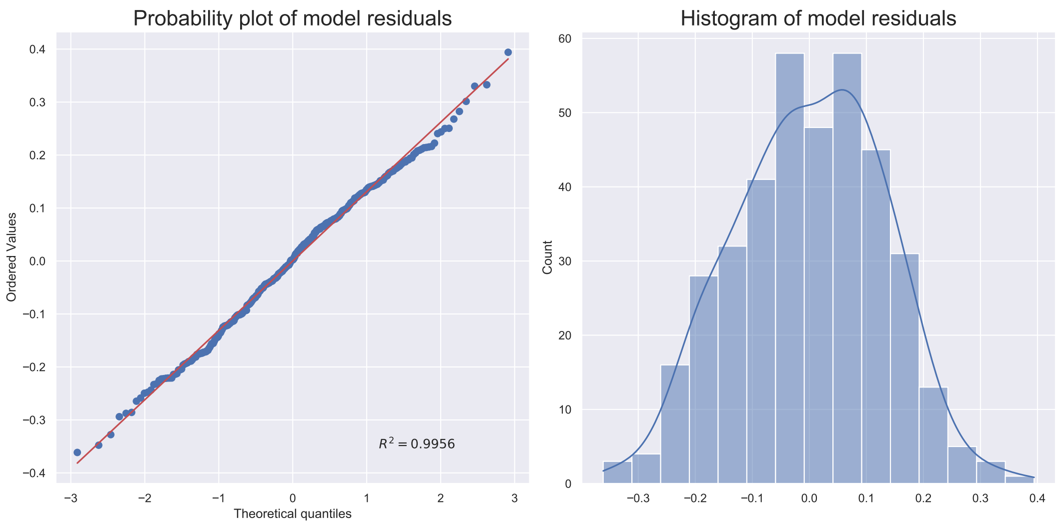


Figure A.35: Normality assumption check for time_at_medium_intensity_pct experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(nation)	0.6291	3.0	0.2097	12.1988	1.22e-07	0.0874	0.0801
Residual	6.5663	382.0	0.0172	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
Belgium	Italy	0.0736	0.001	0.0244	0.1228	True
Belgium	Japan	-0.0395	0.1583	-0.0882	0.0093	False
Belgium	United States	0.0113	0.9	-0.0373	0.0599	False
Italy	Japan	-0.113	0.001	-0.1619	-0.0642	True
Italy	United States	-0.0623	0.0059	-0.111	-0.0135	True
Japan	United States	0.0508	0.0346	0.0026	0.099	True

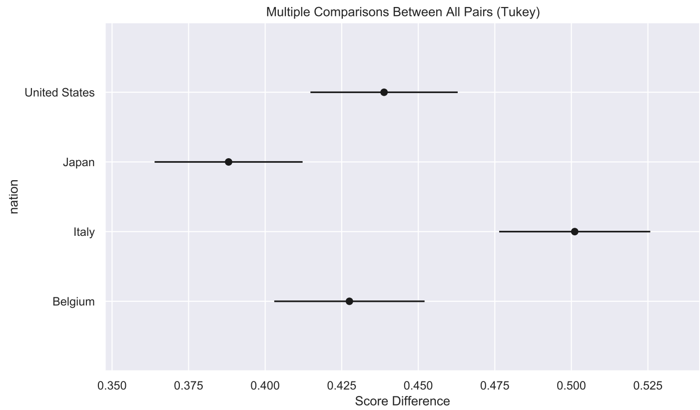


Figure A.36: ANOVA results for time_at_medium_intensity_pct experiment.

nation	N	Mean	SD	Interval
Belgium	95	0.1003	0.0864	0.1179
Italy	94	0.173	0.08	0.1894
Japan	98	0.1453	0.0879	0.1629
United States	99	0.162	0.1056	0.183

test	statistic	p-value
Levene's Test:	1.580729	0.193556
Bartlett's Test:	8.300295	0.040197

Figure A.37: Samples and variances assumption check for time_at_high_intensity_pct experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.95126	0.0

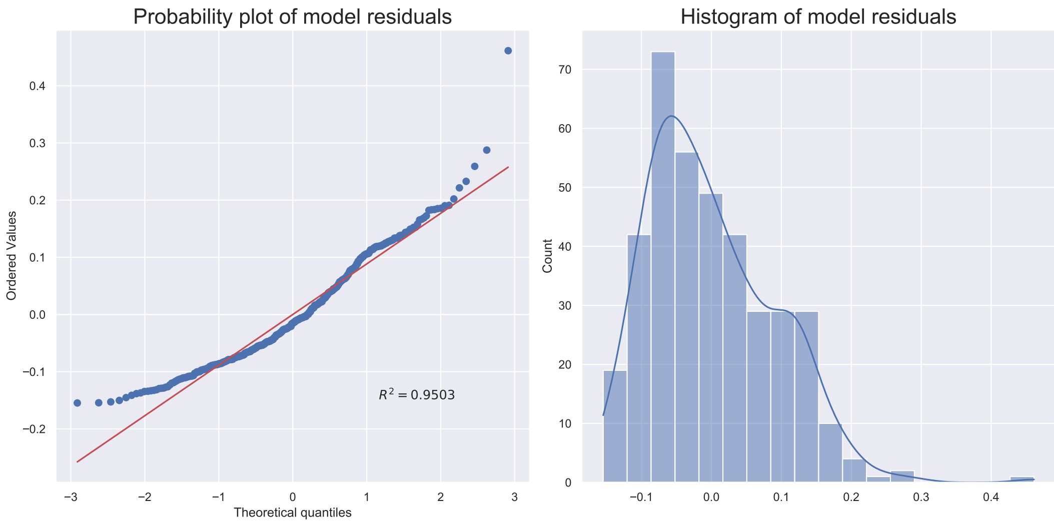


Figure A.38: Normality assumption check for time_at_high_intensity_pct experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(nation)	0.292	3.0	0.0973	11.8439	1.96e-07	0.0851	0.0777
Residual	3.1388	382.0	0.0082	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
Belgium	Italy	0.0727	0.001	0.0387	0.1067	True
Belgium	Japan	0.045	0.0035	0.0113	0.0786	True
Belgium	United States	0.0616	0.001	0.028	0.0952	True
Italy	Japan	-0.0278	0.1484	-0.0615	0.006	False
Italy	United States	-0.0111	0.8093	-0.0448	0.0226	False
Japan	United States	0.0167	0.5609	-0.0167	0.05	False

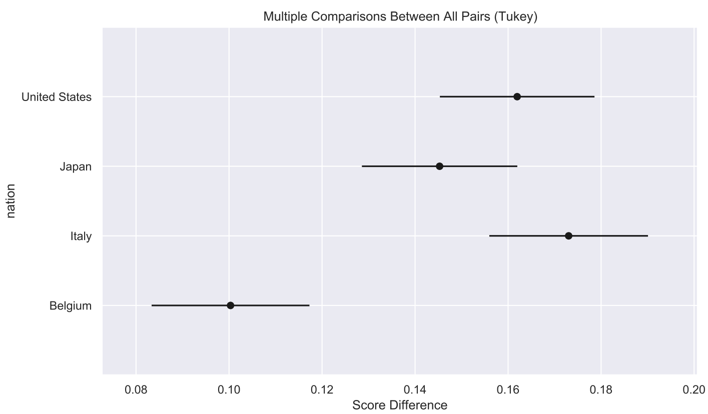


Figure A.39: ANOVA results for time_at_high_intensity_pct experiment.

nation	N	Mean	SD	Interval
Belgium	95	257.118	76.349	272.6711
Italy	94	255.8758	81.0771	272.482
Japan	98	275.6773	109.3238	297.5954
United States	99	274.514	117.3325	297.9155

test	statistic	p-value
Levene’s Test:	6.432752	0.000293
Bartlett's Test:	25.478995	1.2e-05

Figure A.40: Samples and variances assumption check for running_load_rtss experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.977456	1e-05

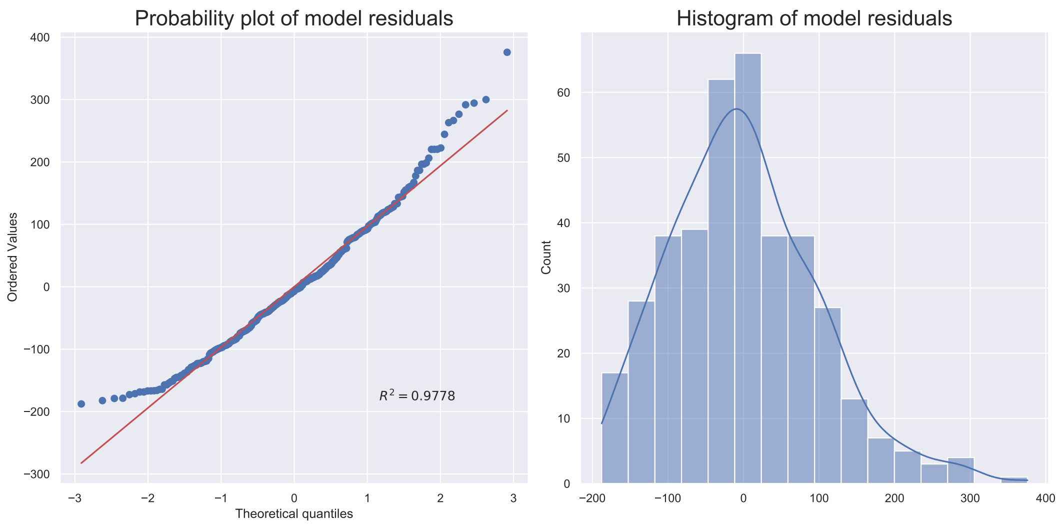


Figure A.41: Normality assumption check for running_load_rtss experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(nation)	33483.53	3.0	11161.18	1.1624	0.3238805	0.009	0.0013
Residual	3667748.6	382.0	9601.44	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
Belgium	Italy	-1.2423	0.9	-38.0271	35.5426	False
Belgium	Japan	18.5593	0.5473	-17.8462	54.9649	False
Belgium	United States	17.3959	0.5918	-18.919	53.7109	False
Italy	Japan	19.8016	0.5001	-16.7022	56.3053	False
Italy	United States	18.6382	0.5444	-17.7752	55.0516	False
Japan	United States	-1.1634	0.9	-37.1936	34.8668	False

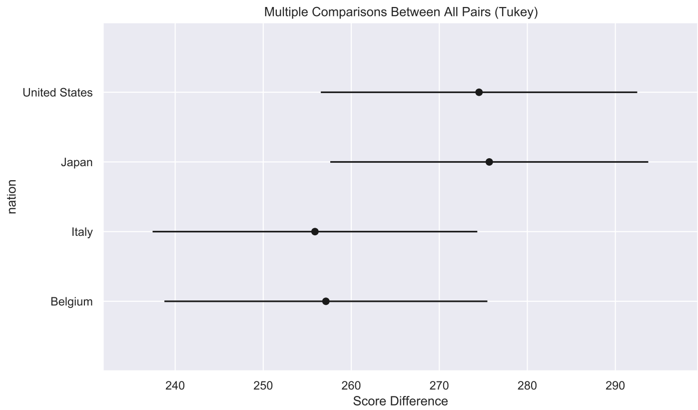


Figure A.42: ANOVA results for running_load_rtss experiment.

nation	N	Mean	SD	Interval
Belgium	96	1.0059	0.0438	1.0148
Italy	96	0.9922	0.0348	0.9993
Japan	99	0.9671	0.0569	0.9785
United States	99	0.9925	0.0533	1.0031

test	statistic	p-value
Levene's Test:	3.292777	0.020658
Bartlett's Test:	26.122953	9e-06

Figure A.43: Samples and variances assumption check for avg_volume_acwr experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.95094	0.0

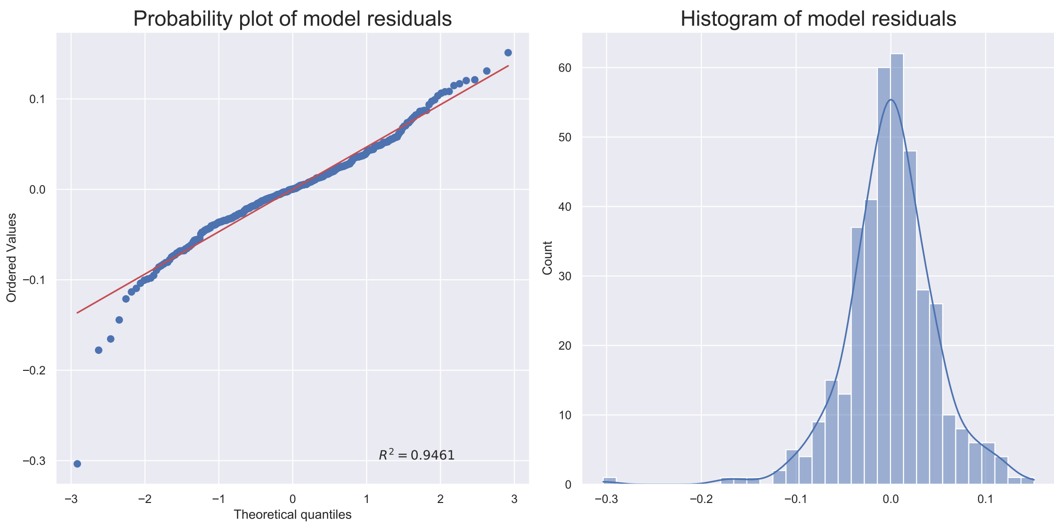


Figure A.44: Normality assumption check for avg_volume_acwr experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(nation)	0.077	3.0	0.0257	11.0995	5.28e-07	0.0794	0.0721
Residual	0.893	386.0	0.0023	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
Belgium	Italy	-0.0137	0.1991	-0.0316	0.0042	False
Belgium	Japan	-0.0388	0.001	-0.0566	-0.021	True
Belgium	United States	-0.0134	0.2093	-0.0312	0.0043	False
Italy	Japan	-0.0251	0.0017	-0.0429	-0.0073	True
Italy	United States	0.0003	0.9	-0.0175	0.0181	False
Japan	United States	0.0254	0.0014	0.0077	0.043	True

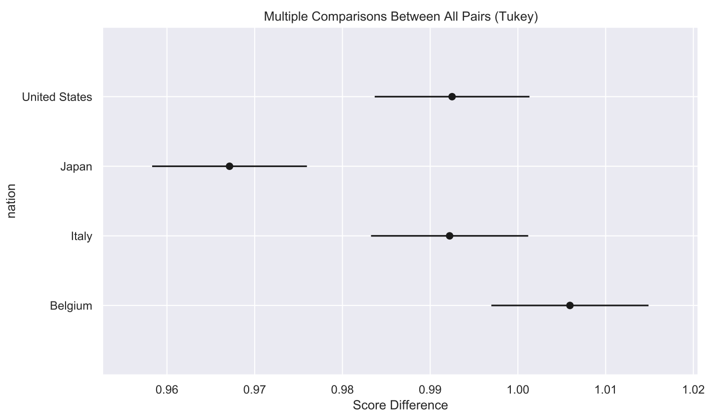


Figure A.45: ANOVA results for avg_volume_acwr experiment.

nation	N	Mean	SD	Interval
Belgium	95	0.4674	0.1294	0.4937
Italy	94	0.3863	0.1366	0.4143
Japan	98	0.4758	0.1686	0.5096
United States	99	0.4544	0.1366	0.4817

test	statistic	p-value
Levene's Test:	2.876639	0.035994
Bartlett's Test:	8.367247	0.039001

Figure A.46: Samples and variances assumption check for pct_of_uniform_runs experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.99292	0.065556

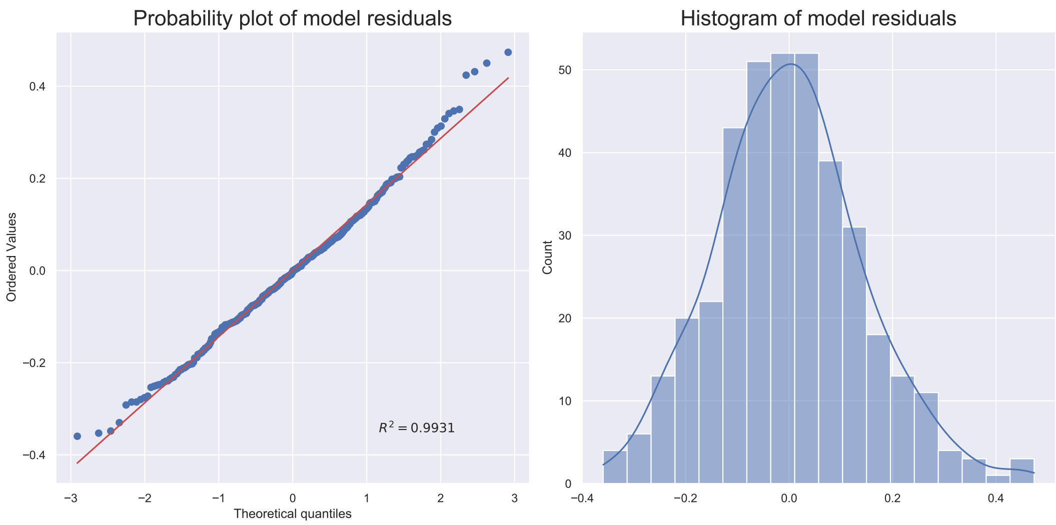


Figure A.47: Normality assumption check for pct_of_uniform_runs experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(nation)	0.4727	3.0	0.1576	7.6224	5.82e-05	0.0565	0.049
Residual	7.8964	382.0	0.0207	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
Belgium	Italy	-0.0811	0.001	-0.1351	-0.0271	True
Belgium	Japan	0.0084	0.9	-0.045	0.0619	False
Belgium	United States	-0.013	0.9	-0.0663	0.0403	False
Italy	Japan	0.0895	0.001	0.036	0.1431	True
Italy	United States	0.0681	0.006	0.0147	0.1215	True
Japan	United States	-0.0214	0.6986	-0.0743	0.0314	False

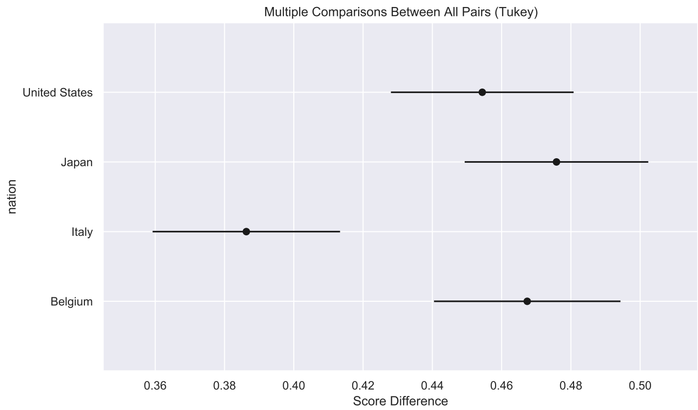


Figure A.48: ANOVA results for pct_of_uniform_runs experiment.

nation	N	Mean	SD	Interval
Belgium	95	0.0953	0.0612	0.1078
Italy	94	0.0793	0.0517	0.0899
Japan	98	0.1395	0.0797	0.1554
United States	99	0.1061	0.0598	0.118

test	statistic	p-value
Levene’s Test:	5.02415	0.001994
Bartlett's Test:	19.387571	0.000227

Figure A.49: Samples and variances assumption check for pct_of_long_runs experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.972381	1e-06

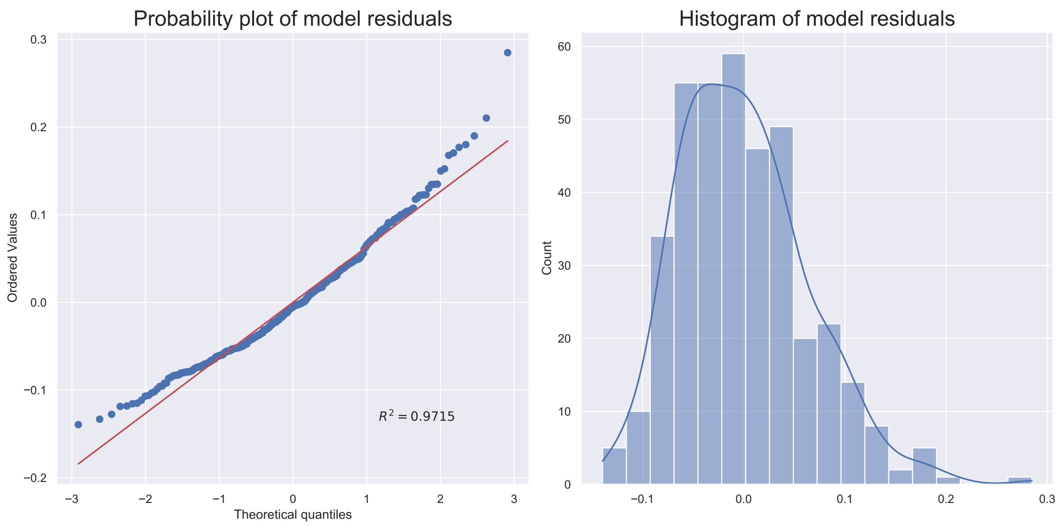


Figure A.50: Normality assumption check for pct_of_long_runs experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(nation)	0.1876	3.0	0.0625	15.2269	2e-09	0.1068	0.0996
Residual	1.5689	382.0	0.0041	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
Belgium	Italy	-0.016	0.3152	-0.0401	0.008	False
Belgium	Japan	0.0442	0.001	0.0204	0.068	True
Belgium	United States	0.0108	0.6269	-0.0129	0.0346	False
Italy	Japan	0.0602	0.001	0.0363	0.0841	True
Italy	United States	0.0268	0.0201	0.003	0.0506	True
Japan	United States	-0.0334	0.0017	-0.0569	-0.0098	True

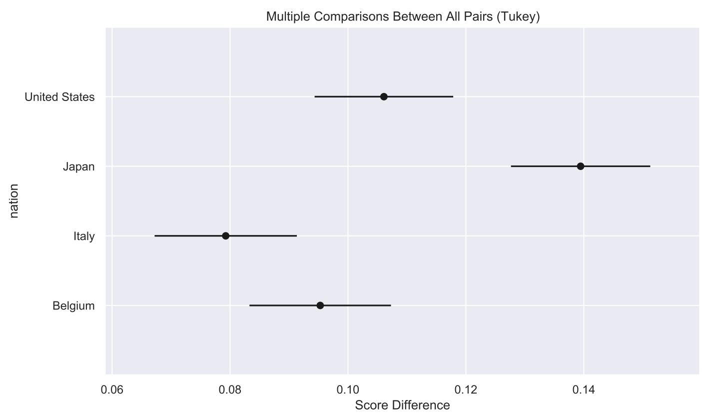


Figure A.51: ANOVA results for pct_of_long_runs experiment.

nation	N	Mean	SD	Interval
Belgium	95	0.0591	0.0437	0.068
Italy	94	0.1317	0.0767	0.1474
Japan	98	0.1085	0.0769	0.1239
United States	99	0.0985	0.0687	0.1122

test	statistic	p-value
Levene's Test:	7.869344	4.2e-05
Bartlett's Test:	33.850392	0.0

Figure A.52: Samples and variances assumption check for pct_of_progression_runs experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.954101	0.0

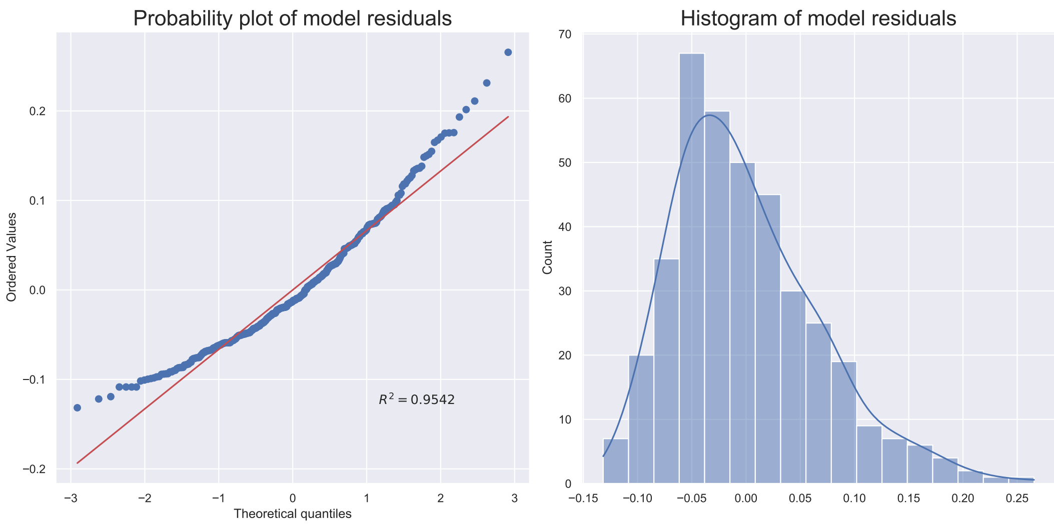


Figure A.53: Normality assumption check for pct_of_progression_runs experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(nation)	0.2606	3.0	0.0869	18.8162	2e-11	0.1287	0.1216
Residual	1.7633	382.0	0.0046	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
Belgium	Italy	0.0726	0.001	0.0471	0.0981	True
Belgium	Japan	0.0494	0.001	0.0242	0.0747	True
Belgium	United States	0.0394	0.001	0.0142	0.0646	True
Italy	Japan	-0.0232	0.0861	-0.0485	0.0021	False
Italy	United States	-0.0332	0.0043	-0.0584	-0.0079	True
Japan	United States	-0.01	0.7058	-0.035	0.015	False

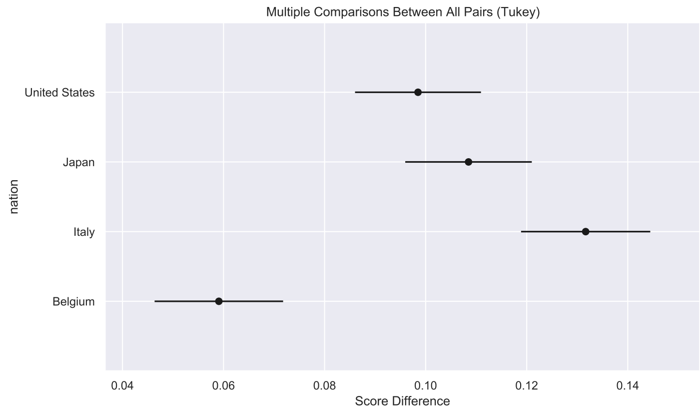


Figure A.54: ANOVA results for pct_of_progression_runs experiment.

nation	N	Mean	SD	Interval
Belgium	95	0.0499	0.0449	0.059
Italy	94	0.0326	0.0321	0.0392
Japan	98	0.028	0.0309	0.0342
United States	99	0.0284	0.0309	0.0346

test	statistic	p-value
Levene's Test:	5.056822	0.001907
Bartlett's Test:	20.611934	0.000127

Figure A.55: Samples and variances assumption check for pct_of_peaks_runs experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.889489	0.0

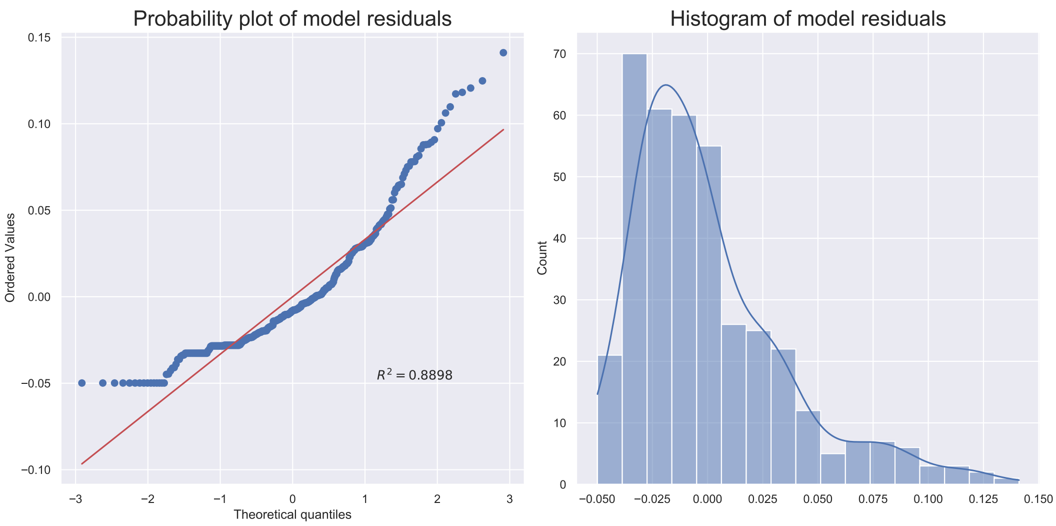


Figure A.56: Normality assumption check for pct_of_peaks_runs experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(nation)	0.0305	3.0	0.0102	8.2483	2.49e-05	0.0608	0.0533
Residual	0.4714	382.0	0.0012	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
Belgium	Italy	-0.0172	0.0045	-0.0304	-0.004	True
Belgium	Japan	-0.0218	0.001	-0.0349	-0.0088	True
Belgium	United States	-0.0215	0.001	-0.0345	-0.0084	True
Italy	Japan	-0.0046	0.7759	-0.0177	0.0085	False
Italy	United States	-0.0042	0.8166	-0.0173	0.0088	False
Japan	United States	0.0004	0.9	-0.0125	0.0133	False

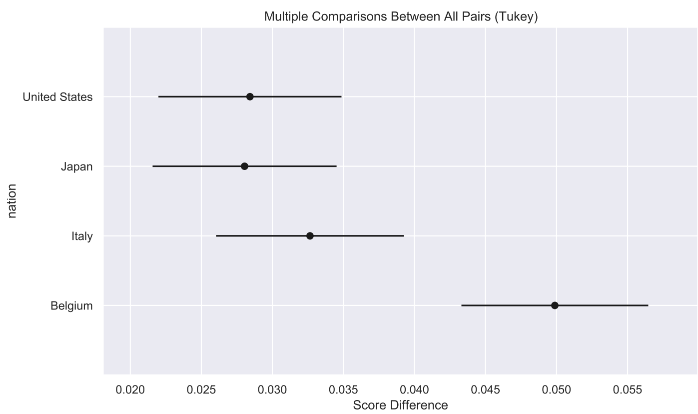


Figure A.57: ANOVA results for pct_of_peaks_runs experiment.

nation	N	Mean	SD	Interval
Belgium	95	0.0459	0.0403	0.0542
Italy	94	0.0407	0.0437	0.0496
Japan	98	0.0655	0.0562	0.0768
United States	99	0.0582	0.0454	0.0672

test	statistic	p-value
Levene's Test:	2.514449	0.058085
Bartlett's Test:	12.13941	0.006921

Figure A.58: Samples and variances assumption check for pct_of_recovery_runs experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.896302	0.0

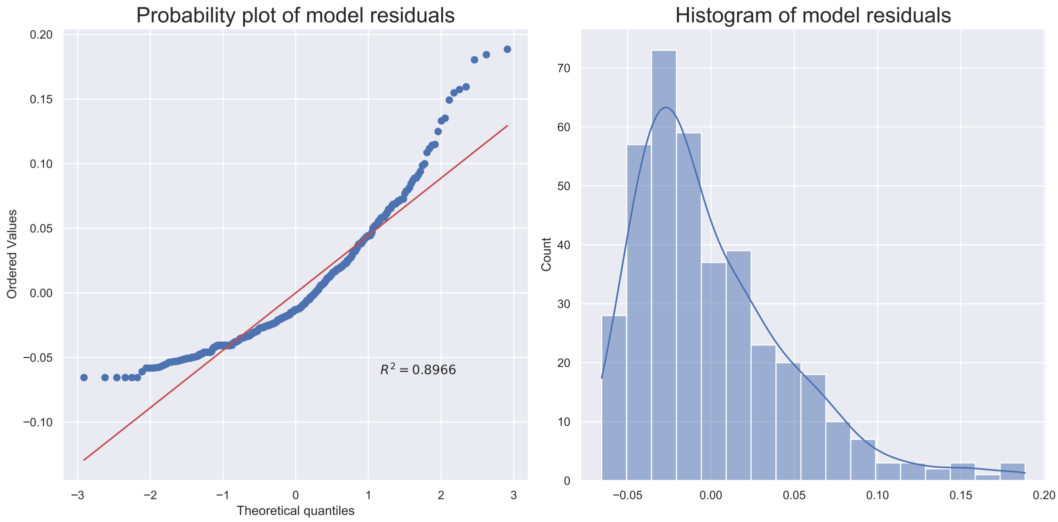


Figure A.59: Normality assumption check for pct_of_recovery_runs experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(nation)	0.037	3.0	0.0123	5.6225	0.000884	0.0423	0.0347
Residual	0.8389	382.0	0.0022	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
Belgium	Italy	-0.0053	0.8527	-0.0228	0.0123	False
Belgium	Japan	0.0196	0.0202	0.0022	0.037	True
Belgium	United States	0.0122	0.2663	-0.0051	0.0296	False
Italy	Japan	0.0249	0.0016	0.0074	0.0423	True
Italy	United States	0.0175	0.0485	0.0001	0.0349	True
Japan	United States	-0.0074	0.6666	-0.0246	0.0099	False

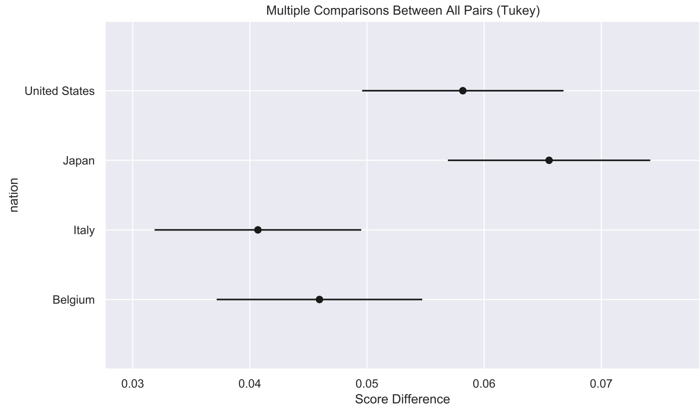


Figure A.60: ANOVA results for pct_of_recovery_runs experiment.

A.2 Seasonality

season	N	Mean	SD	Interval
autumn	381	37337.5364	16937.6292	39043.7117
spring	398	40719.2137	17997.0458	42492.7243
summer	395	36997.4246	18028.9108	38780.8508
winter	393	39821.7272	18807.7988	41686.9587

test	statistic	p-value
Levene's Test:	0.680599	0.563917
Bartlett's Test:	4.24234	0.236457

Figure A.61: Samples and variances assumption check for running_volume experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.96328	0.0

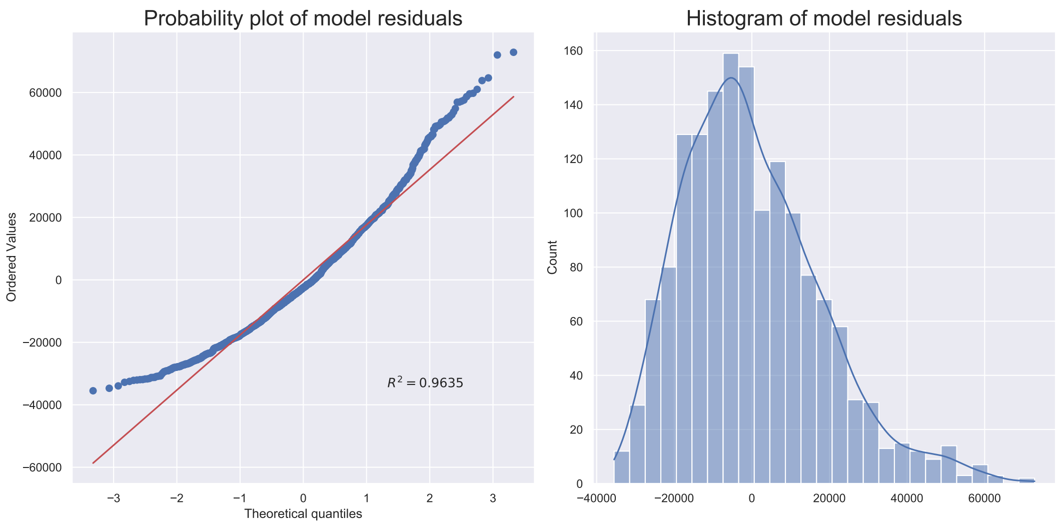


Figure A.62: Normality assumption check for running_volume experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(season)	3967716685.5	3.0	1322572228.5	4.0989	0.006567	0.0078	0.0059
Residual	504331282570.9	1563.0	322668766.8	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
autumn	spring	3381.6773	0.0432	70.5072	6692.8475	True
autumn	summer	-340.1118	0.9	-3657.4261	2977.2025	False
autumn	winter	2484.1909	0.2186	-837.2652	5805.6469	False
spring	summer	-3721.7891	0.0188	-7002.8426	-440.7357	True
spring	winter	-897.4865	0.8908	-4182.7274	2387.7544	False
summer	winter	2824.3027	0.122	-467.1308	6115.7361	False

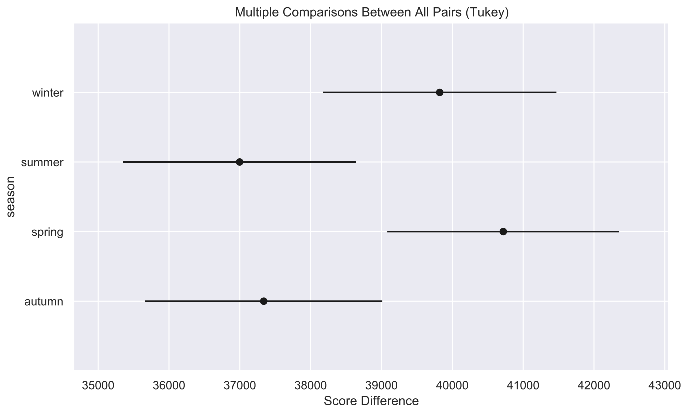


Figure A.63: ANOVA results for running_volume experiment.

season	N	Mean	SD	Interval
autumn	381	10220.6566	3093.8842	10532.3123
spring	398	10573.9499	3384.7422	10907.4977
summer	395	9666.0358	3005.0761	9963.299
winter	393	10379.2144	2997.4581	10676.4822

test	statistic	p-value
Levene’s Test:	0.977466	0.402452
Bartlett's Test:	7.909451	0.047921

Figure A.64: Samples and variances assumption check for avg_run_volume experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.976715	0.0

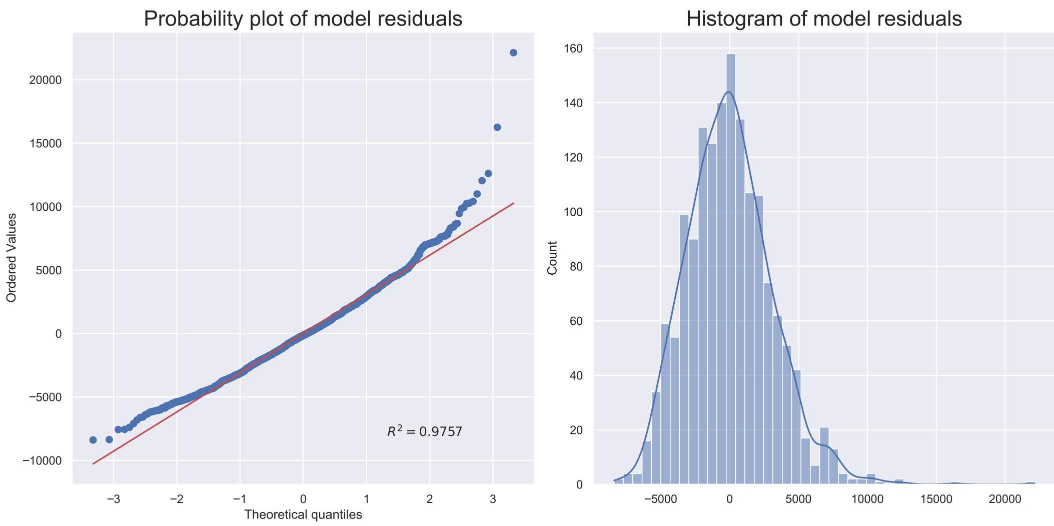


Figure A.65: Normality assumption check for avg_run_volume experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(season)	180894268.5	3.0	60298089.5	6.1737	0.000361	0.0117	0.0098
Residual	15265661875.2	1563.0	9766898.2	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
autumn	spring	353.2933	0.3932	-222.7848	929.3714	False
autumn	summer	-554.6208	0.0649	-1131.7679	22.5262	False
autumn	winter	158.5578	0.8891	-419.3099	736.4254	False
spring	summer	-907.9141	0.001	-1478.7525	-337.0757	True
spring	winter	-194.7355	0.7934	-766.3024	376.8314	False
summer	winter	713.1786	0.0076	140.5343	1285.8229	True

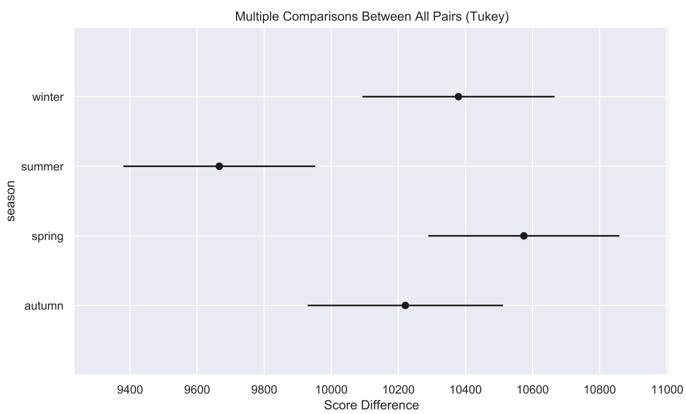


Figure A.66: ANOVA results for avg_run_volume experiment.

season	N	Mean	SD	Interval
autumn	381	0.9461	0.0957	0.9557
spring	398	0.9389	0.0881	0.9476
summer	395	0.9042	0.1286	0.9169
winter	393	0.951	0.0773	0.9587

test	statistic	p-value
Levene’s Test:	17.21094	0.0
Bartlett's Test:	116.2445	0.0

Figure A.67: Samples and variances assumption check for pct_time_running_vs_cross experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.759499669	0.0

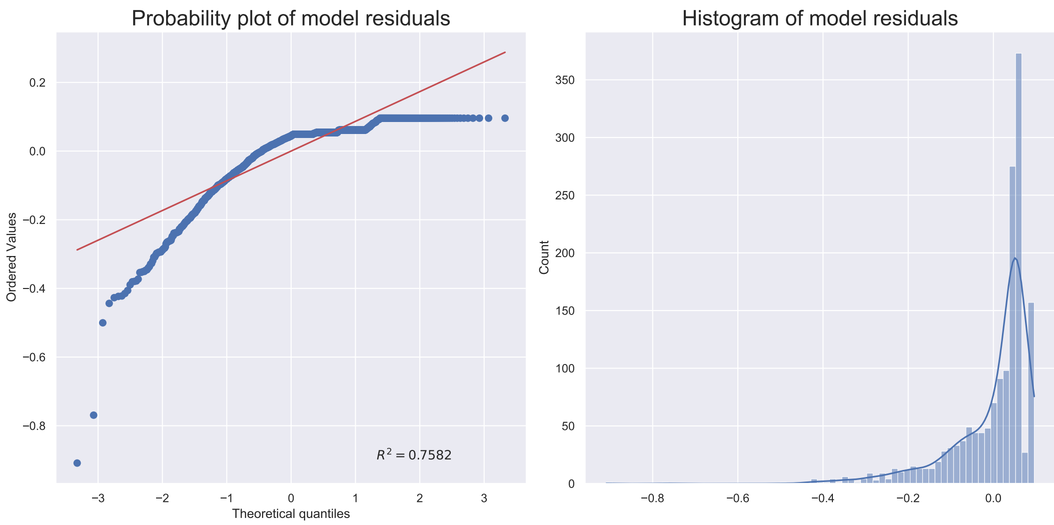


Figure A.68: Normality assumption check for pct_time_running_vs_cross experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(season)	0.5293	3.0	0.1764	17.88	2e-11	0.0332	0.0313
Residual	15.4225	1563.0	0.0099	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
autumn	spring	-0.0072	0.7164	-0.0255	0.0111	False
autumn	summer	-0.0419	0.001	-0.0603	-0.0236	True
autumn	winter	0.0049	0.8978	-0.0134	0.0233	False
spring	summer	-0.0347	0.001	-0.0529	-0.0166	True
spring	winter	0.0121	0.3141	-0.006	0.0303	False
summer	winter	0.0469	0.001	0.0287	0.0651	True

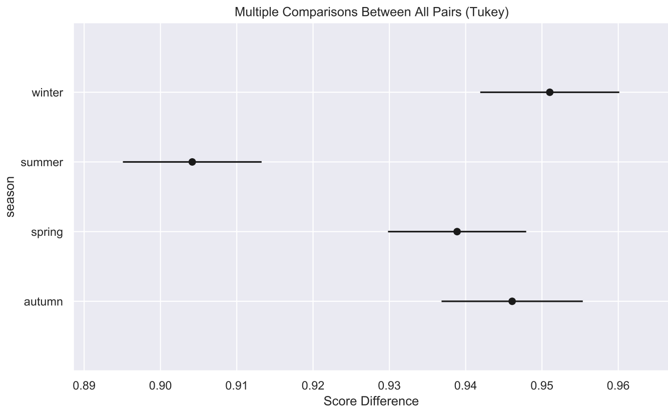


Figure A.69: ANOVA results for pct_time_running_vs_cross experiment.

season	N	Mean	SD	Interval
autumn	308	0.3457	0.1978	0.3679
spring	384	0.3576	0.1934	0.377
summer	375	0.3883	0.2055	0.4092
winter	362	0.3389	0.2019	0.3598

test	statistic	p-value
Levene's Test:	0.633349	0.593537
Bartlett's Test:	1.527536	0.67593

Figure A.70: Samples and variances assumption check for time_at_low_intensity_pct experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.965259969	0.0

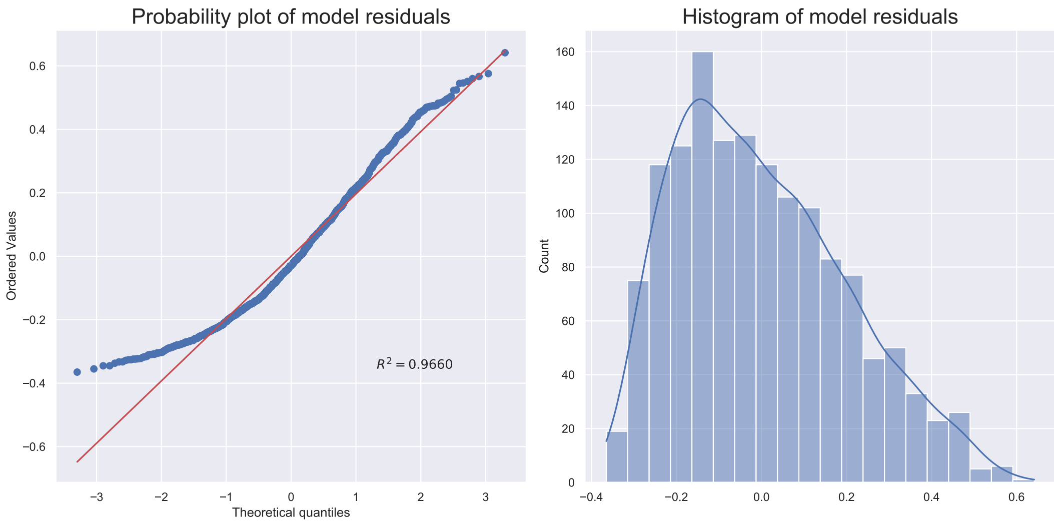


Figure A.71: Normality assumption check for time_at_low_intensity_pct experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(season)	0.5225	3.0	0.1742	4.3671	0.004538	0.0091	0.007
Residual	56.8292	1425.0	0.0399	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
autumn	spring	0.0119	0.8476	-0.0274	0.0512	False
autumn	summer	0.0426	0.0286	0.0031	0.0821	True
autumn	winter	-0.0068	0.9	-0.0466	0.033	False
spring	summer	0.0307	0.1481	-0.0066	0.068	False
spring	winter	-0.0187	0.5686	-0.0563	0.0189	False
summer	winter	-0.0494	0.0045	-0.0872	-0.0115	True

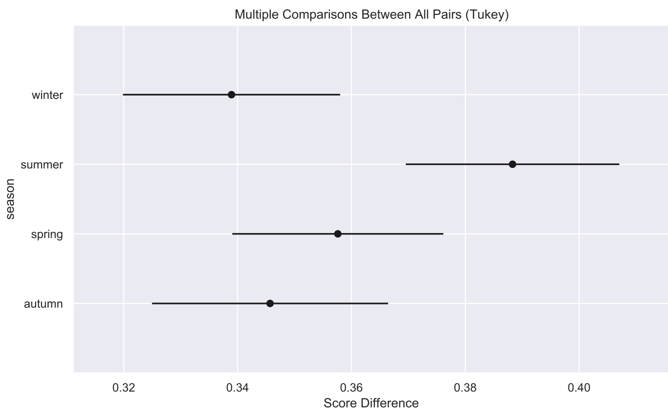


Figure A.72: ANOVA results for time_at_low_intensity_pct experiment.

season	N	Mean	SD	Interval
autumn	310	0.1681	0.1554	0.1854
spring	389	0.1576	0.1215	0.1697
summer	379	0.1409	0.1212	0.1532
winter	364	0.1624	0.1356	0.1763

test	statistic	p-value
Levene's Test:	2.747387	0.041649
Bartlett's Test:	29.057454	2e-06

Figure A.73: Samples and variances assumption check for time_at_high_intensity_pct experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.895866454	0.0

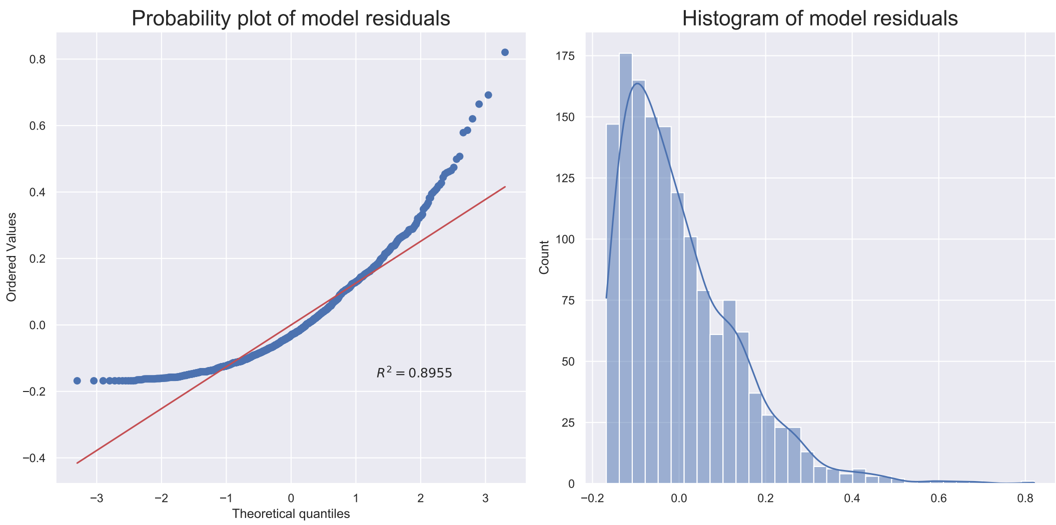


Figure A.74: Normality assumption check for time_at_high_intensity_pct experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(season)	0.1463	3.0	0.0488	2.7578	0.041068	0.0057	0.0036
Residual	25.4211	1438.0	0.0177	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
autumn	spring	-0.0104	0.7061	-0.0365	0.0156	False
autumn	summer	-0.0271	0.039	-0.0533	-0.0009	True
autumn	winter	-0.0057	0.9	-0.0321	0.0207	False
spring	summer	-0.0167	0.3041	-0.0414	0.008	False
spring	winter	0.0047	0.9	-0.0202	0.0297	False
summer	winter	0.0214	0.1248	-0.0037	0.0465	False

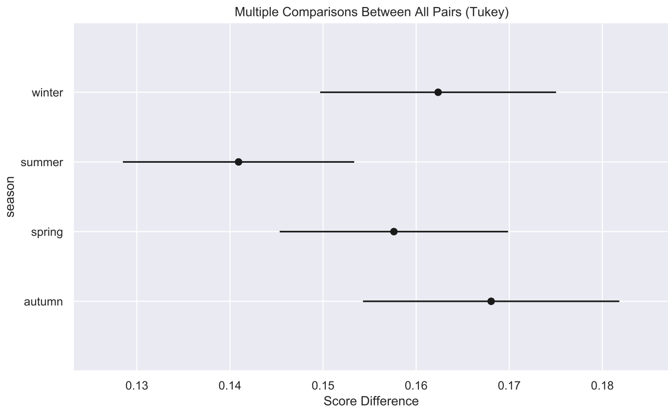


Figure A.75: ANOVA results for time_at_high_intensity_pct experiment.

season	N	Mean	SD	Interval
autumn	308	280.6776	116.0334	293.6875
spring	384	273.1433	116.0886	284.7912
summer	375	247.8873	119.4713	260.0185
winter	362	282.5134	129.3266	295.8806

test	statistic	p-value
Levene’s Test:	1.510977	0.209866
Bartlett's Test:	5.72534	0.125765

Figure A.76: Samples and variances assumption check for running_load_rtss experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.963372469	0.0

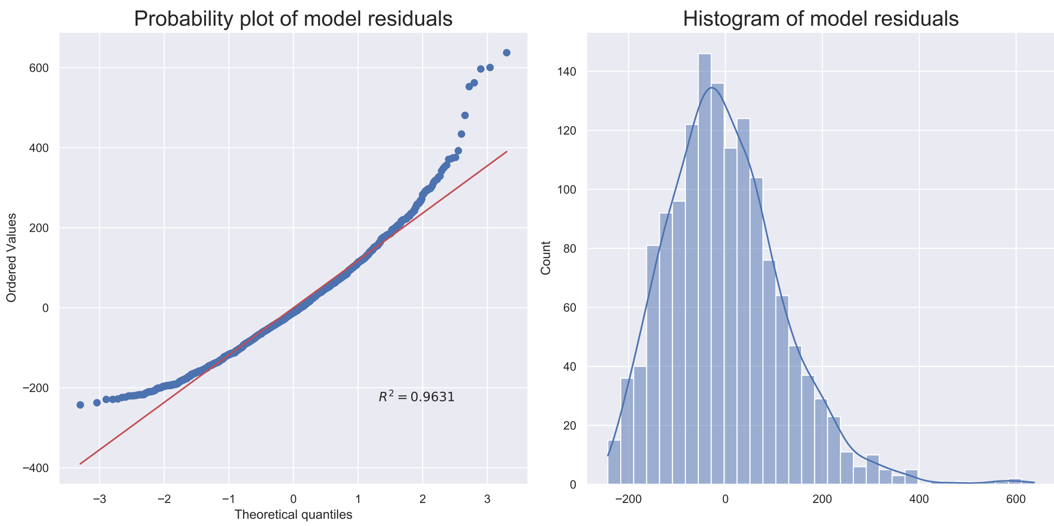


Figure A.77: Normality assumption check for running_load_rtss experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(season)	278581.25	3.0	92860.42	6.4015	0.000263	0.0133	0.0112
Residual	20670999.01	1425.0	14505.96	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
autumn	spring	-7.5343	0.8262	-31.23	16.1614	False
autumn	summer	-32.7903	0.0023	-56.6122	-8.9684	True
autumn	winter	1.8357	0.9	-22.1783	25.8498	False
spring	summer	-25.256	0.0205	-47.7464	-2.7656	True
spring	winter	9.3701	0.6892	-13.3237	32.0638	False
summer	winter	34.6261	0.001	11.8005	57.4516	True

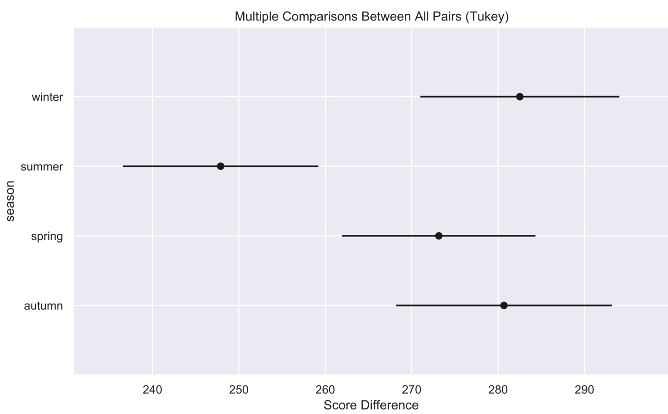


Figure A.78: ANOVA results for running_load_rtss experiment.

season	N	Mean	SD	Interval
autumn	356	772.9247	324.4084	806.7388
spring	392	853.0748	365.3601	889.3553
summer	390	768.4619	378.4902	806.143
winter	387	829.3722	368.11	866.1626

test	statistic	p-value
Levene’s Test:	0.910355	0.435235
Bartlett's Test:	9.751986	0.020796

Figure A.79: Samples and variances assumption check for running_load_edwards experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.965186238	0.0

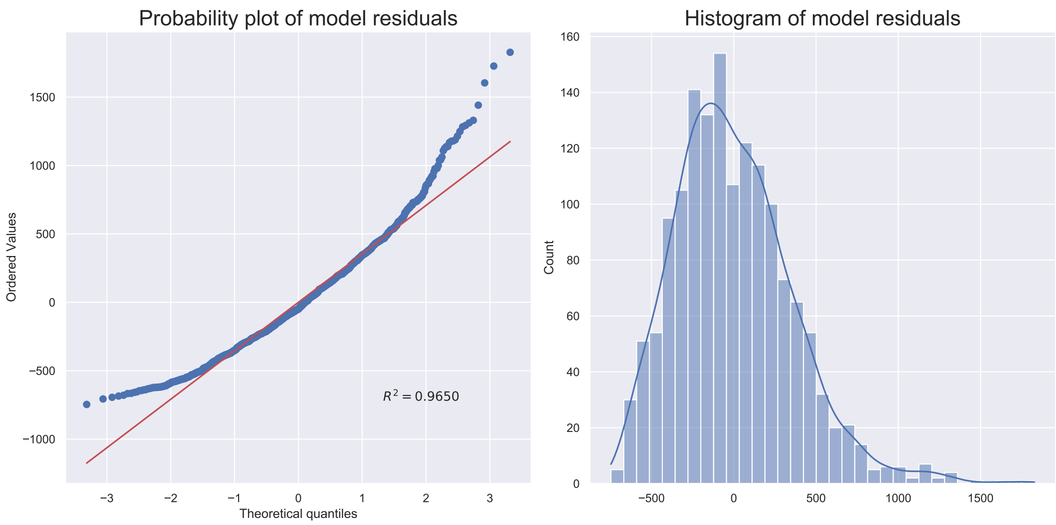


Figure A.80: Normality assumption check for running_load_edwards experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(season)	2018329.15	3.0	672776.38	5.179	0.001461	0.0101	0.0082
Residual	197585338.97	1521.0	129904.89	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
autumn	spring	80.1501	0.013	12.285	148.0153	True
autumn	summer	-4.4628	0.9	-72.4107	63.4851	False
autumn	winter	56.4475	0.1432	-11.6259	124.521	False
spring	summer	-84.6129	0.0058	-150.9097	-18.3162	True
spring	winter	-23.7026	0.7701	-90.128	42.7228	False
summer	winter	60.9104	0.0865	-5.5996	127.4203	False

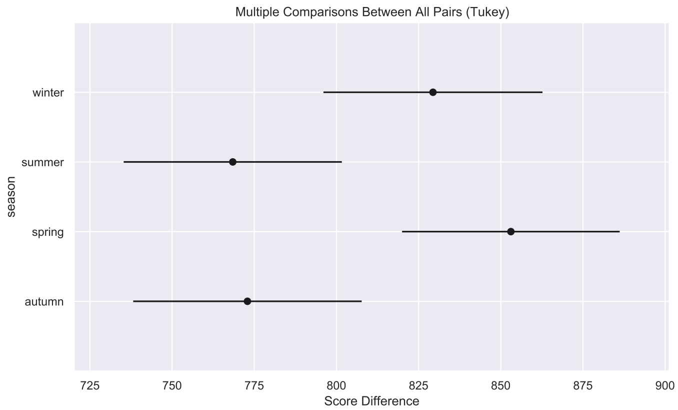


Figure A.81: ANOVA results for running_load_edwards experiment.

season	N	Mean	SD	Interval
autumn	310	0.9475	0.1155	0.9604
spring	389	0.9867	0.093	0.996
summer	379	1.0033	0.1152	1.0149
winter	364	1.0204	0.1301	1.0338

test	statistic	p-value
Levene's Test:	2.948846	0.031746
Bartlett's Test:	42.383997	0.0

Figure A.82: Samples and variances assumption check for avg_volume_acwr experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.851985455	0.0

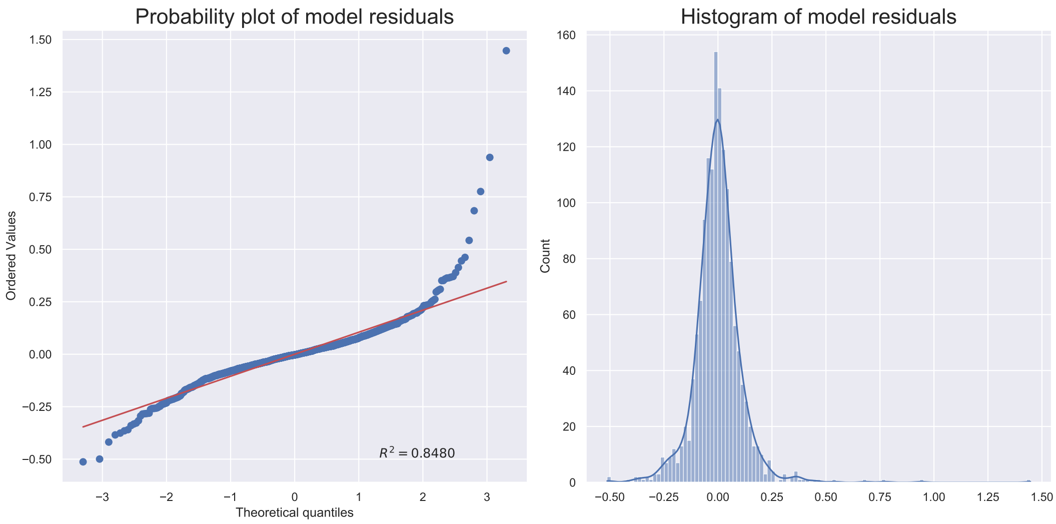


Figure A.83: Normality assumption check for avg_volume_acwr experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(season)	0.96	3.0	0.32	24.8115	1e-15	0.0492	0.0472
Residual	18.64	1438.0	0.01	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
autumn	spring	0.0392	0.001	0.0169	0.0615	True
autumn	summer	0.0557	0.001	0.0333	0.0782	True
autumn	winter	0.0729	0.001	0.0503	0.0955	True
spring	summer	0.0165	0.1836	-0.0046	0.0377	False
spring	winter	0.0337	0.001	0.0123	0.055	True
summer	winter	0.0171	0.1696	-0.0043	0.0386	False

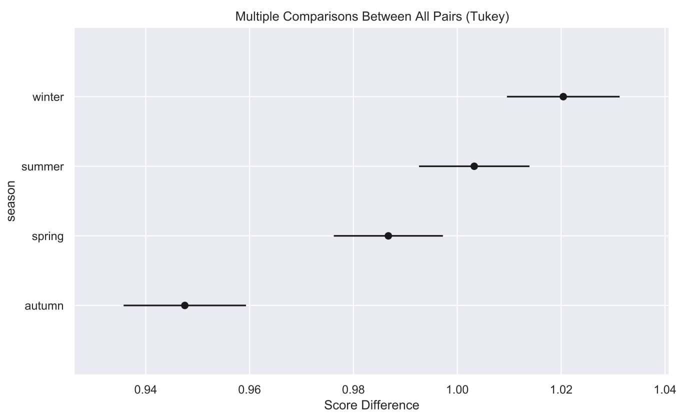


Figure A.84: ANOVA results for avg_volume_acwr experiment.

season	N	Mean	SD	Interval
autumn	310	0.4304	0.2148	0.4544
spring	389	0.4493	0.1916	0.4684
summer	379	0.4747	0.2109	0.496
winter	364	0.4457	0.1926	0.4655

test	statistic	p-value
Levene’s Test:	2.584975	0.051777
Bartlett's Test:	7.542018	0.056489

Figure A.85: Samples and variances assumption check for pct_of_uniform_runs experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.991644502	2.66e-07

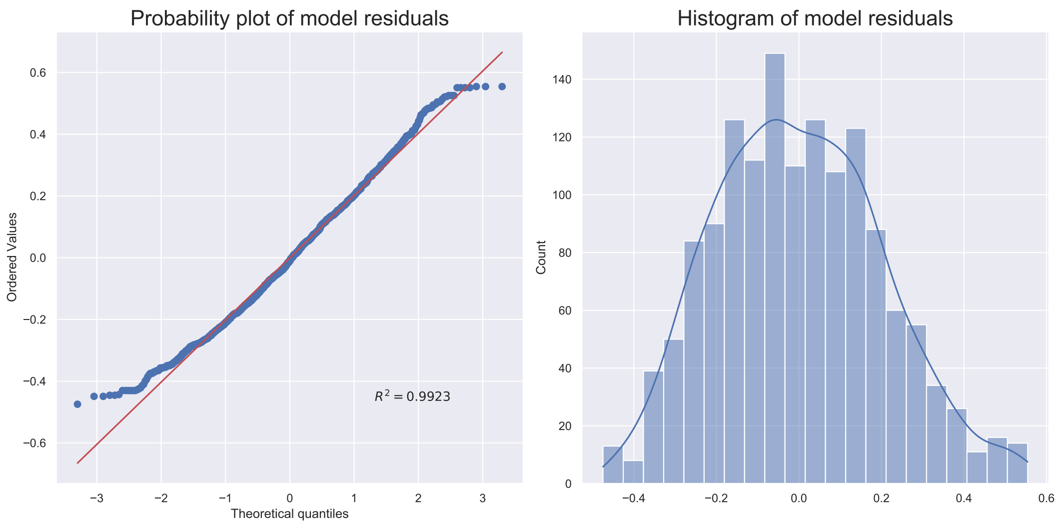


Figure A.86: Normality assumption check for pct_of_uniform_runs experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(season)	0.3563	3.0	0.1188	2.9055	0.03366	0.006	0.0039
Residual	58.7785	1438.0	0.0409	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
autumn	spring	0.0189	0.5977	-0.0207	0.0584	False
autumn	summer	0.0443	0.0221	0.0045	0.0841	True
autumn	winter	0.0153	0.7364	-0.0249	0.0555	False
spring	summer	0.0255	0.3009	-0.0121	0.063	False
spring	winter	-0.0036	0.9	-0.0415	0.0343	False
summer	winter	-0.029	0.2049	-0.0672	0.0091	False

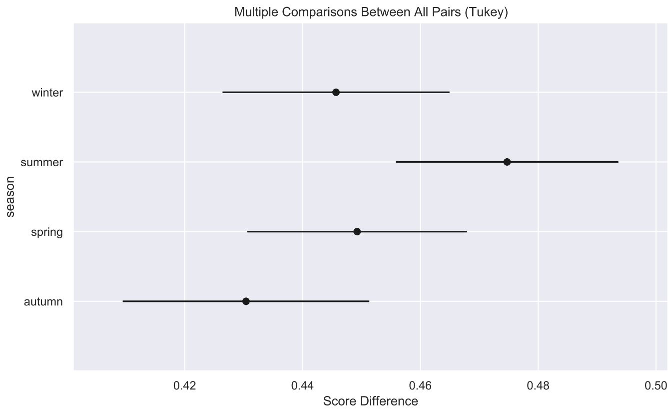


Figure A.87: ANOVA results for pct_of_uniform_runs experiment.

season	N	Mean	SD	Interval
autumn	310	0.3917	0.2269	0.4171
spring	389	0.417	0.2038	0.4373
summer	379	0.445	0.2273	0.468
winter	364	0.4082	0.1999	0.4288

test	statistic	p-value
Levene's Test:	4.194676	0.005762
Bartlett's Test:	10.079789	0.0179

Figure A.88: Samples and variances assumption check for pct_of_uniform_under_lt_runs experiment.

test	statistic	p-value
Shapiro-Wilk Test:	0.986717641	0.0

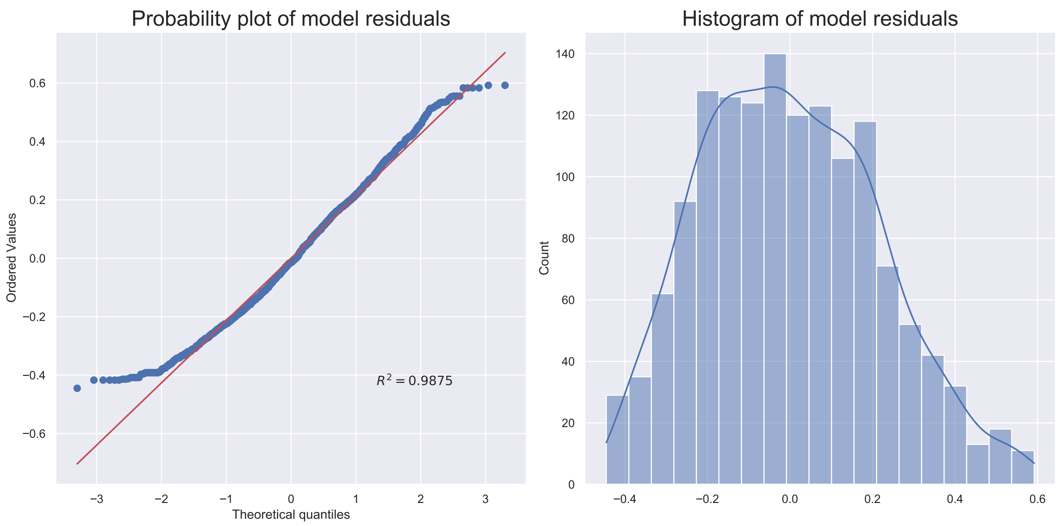


Figure A.89: Normality assumption check for pct_of_uniform_under_lt_runs experiment.

index	sum_sq	df	mean_sq	F	PR(>F)	eta_sq	omega_sq
C(season)	0.5245	3.0	0.1748	3.8063	0.00985	0.0079	0.0058
Residual	66.0473	1438.0	0.0459	nan	nan	nan	nan

group1	group2	meandiff	p-adj	lower	upper	reject
autumn	spring	0.0253	0.4087	-0.0167	0.0673	False
autumn	summer	0.0533	0.0065	0.0111	0.0955	True
autumn	winter	0.0165	0.7265	-0.0261	0.0591	False
spring	summer	0.028	0.2681	-0.0118	0.0678	False
spring	winter	-0.0088	0.9	-0.049	0.0314	False
summer	winter	-0.0368	0.0889	-0.0773	0.0036	False

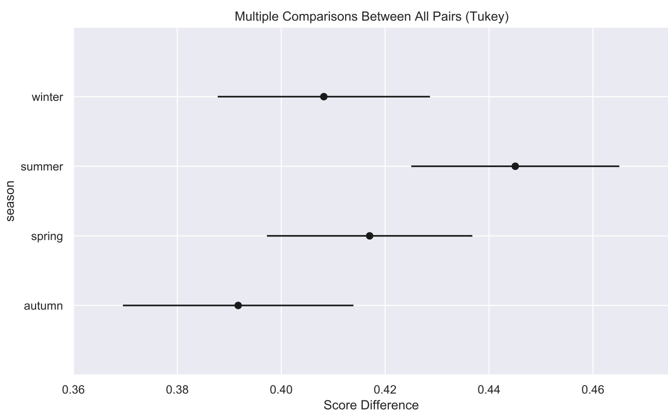


Figure A.90: ANOVA results for pct_of_uniform_under_lt_runs experiment.

A.3 Nation/Season interaction

nation	season	N	Mean	SD	Interval
Belgium	autumn	98	0.9369	0.099	0.9568
Belgium	spring	99	0.9175	0.0926	0.9359
Belgium	summer	98	0.8634	0.1371	0.8909
Belgium	winter	99	0.9451	0.0839	0.9618
Italy	autumn	96	0.9416	0.0885	0.9595
Italy	spring	98	0.9359	0.0916	0.9542
Italy	summer	98	0.8828	0.1285	0.9086
Italy	winter	98	0.9448	0.0807	0.9609
Japan	autumn	93	0.9683	0.0643	0.9815
Japan	spring	101	0.9606	0.076	0.9756
Japan	summer	99	0.9589	0.081	0.9751
Japan	winter	99	0.9676	0.0611	0.9798
United States	autumn	93	0.948	0.0753	0.9635
United States	spring	100	0.9411	0.0874	0.9585
United States	summer	99	0.9187	0.1166	0.942
United States	winter	97	0.9465	0.0804	0.9627

Figure A.91: Samples for pct_time_running_vs_cross experiment.

index	sum_sq	df	F	PR(>F)
Intercept	1370.71	1.0	159728.96	0.0
nation	0.51	3.0	19.619	2e-12
season	0.51	3.0	19.7398	1e-12
nation:season	0.2	9.0	2.6073	0.005470997952
Residual	13.29	1549.0	nan	nan

Figure A.92: ANOVA results for pct_time_running_vs_cross experiment.

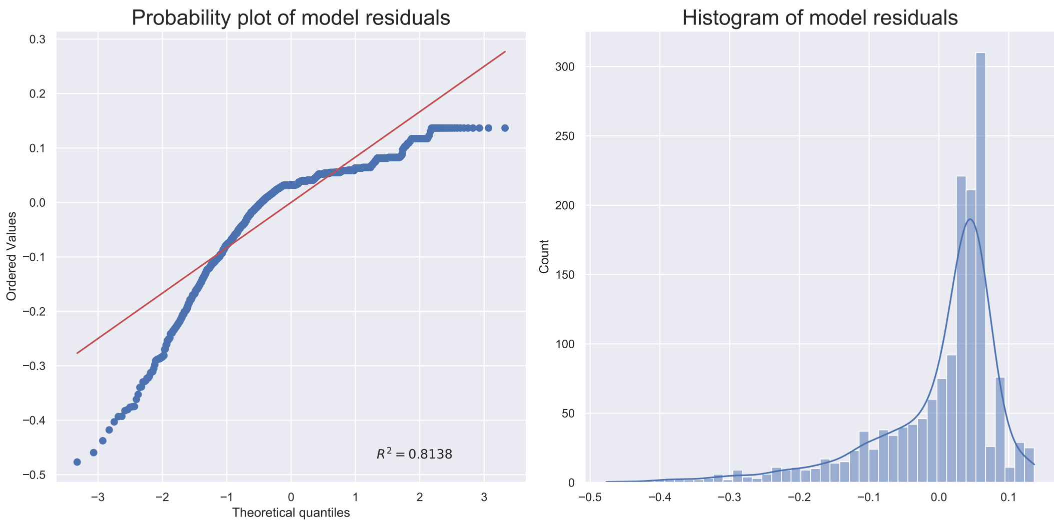


Figure A.93: Normality assumption check for pct_time_running_vs_cross experiment.

nation	season	N	Mean	SD	Interval
Belgium	autumn	79	0.9756	0.1474	1.0086
Belgium	spring	96	0.9972	0.084	1.0142
Belgium	summer	94	1.0188	0.0947	1.0382
Belgium	winter	93	1.029	0.0748	1.0444
Italy	autumn	73	0.9519	0.0867	0.9721
Italy	spring	96	0.9969	0.0802	1.0132
Italy	summer	93	0.9984	0.0855	1.016
Italy	winter	89	1.0166	0.1043	1.0386
Japan	autumn	79	0.9656	0.1036	0.9888
Japan	spring	97	0.9656	0.1017	0.9861
Japan	summer	94	0.9709	0.0843	0.9881
Japan	winter	89	0.9754	0.0828	0.9929
United States	autumn	79	0.9115	0.1244	0.9394
United States	spring	98	0.9872	0.0828	1.0038
United States	summer	96	1.0124	0.1099	1.0347
United States	winter	92	1.0239	0.1135	1.0474

Figure A.94: Samples for avg_volume_acwr experiment.

index	sum_sq	df	F	PR(>F)
Intercept	1389.73	1.0	143273.1426	0.0
nation	0.24	3.0	8.1924	2.085244159e-05
season	0.67	3.0	23.0855	1e-14
nation:season	0.29	9.0	3.3451	0.00046255495638
Residual	13.78	1421.0	nan	nan

Figure A.95: ANOVA results for avg_volume_acwr experiment.

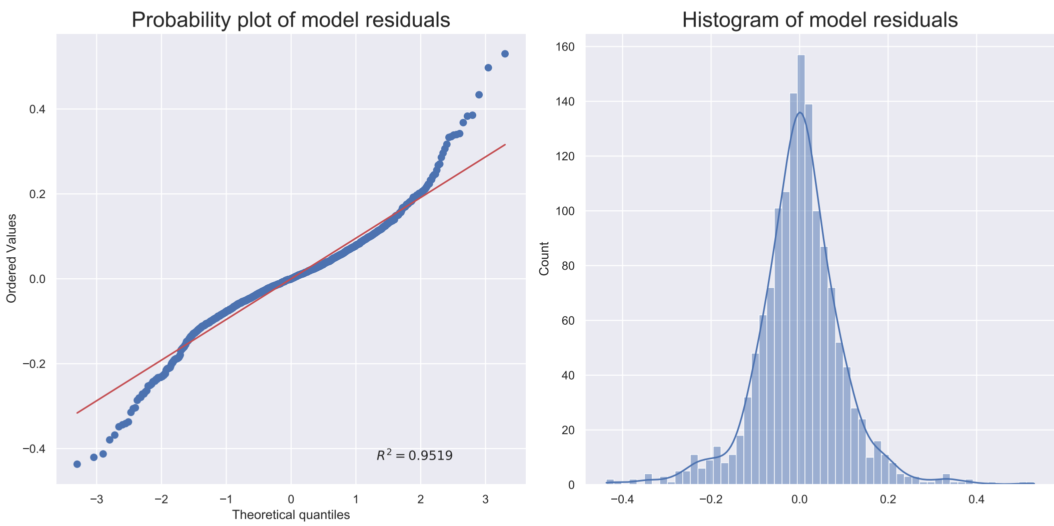


Figure A.96: Normality assumption check for avg_volume_acwr experiment.