Master degree in Mathematical Engineering
Laboratory of Statistics, MOX

Master Thesis, Politecnico di Milano

R-package for Interval-Wise Testing Procedure

Author :
Alexandre Podvin
matricola: 854992

Thesis Supervisors:
Simone Vantini
Alessia Pini

Academic Year: 2017-2018

Thanks

I would like to thanks Simone Vantini and Alessia Pini who allowed me to carry out this master thesis in statistics on a particularly interesting subject. To all my years in Milan.

Contents

List of Figures

Introduction

Standard parametric approaches requires some applicable assumptions on the experiment design (random sampling) and the population model (normal distribution or homoscedasticity, for example). When the application conditions of these approaches are not respected, in particular when the data distribution law is not well fitting with the test requirements, the outputs of the parametric tests are less reliable. In that context, non-parametric tests can offer a more relevant alternative since they require less assumptions to be assessed. In that framework, we think "conditionally to the data themselves" . An important class of non-parametric tests are the permutations test. That will be the underlying method used in that document.

Fisher (1935) and Pitman (1937) has presented for the first time this class of non-parametric tests. Back then, when they were created, these kind of tests were not very used, because of the loud amount of computation time they required. But with the computation capacity increasing, these tests became more an more democratized, and used in many different kind of applications. We will study in this work the combination of the permutation tests with the so-called IWTP, the interval-wise testing procedure that provide a way to compute test p-values allowing a global control of the type-I error over the multivariate tests involved.

The goal of this work is to provide to the R package "fdatest" an extension bundling new functions and their documentation using the IWTP. The underlying testing method that drives this procedure in the R package are the permutation tests presented in the abstract. The functions are implemented for both non-functional and functional data ("fda" package required for functional data processing).

Chapter 1

Statistical Hypothesis Testing

Hypothesis testing is a decision-making process for evaluating some claims about a population. Basically, we are looking for evaluating some features or parameters of the population from which an experimental sample is extracted. From this sample we will try to get information about an intrinsic population feature, generally a mean, a standard deviation or a proportion.

1.1 Parametric and Non-parametric frameworks

In a parametric framework, we set a statistical model $(\mathcal{O}^n, (\mathbb{P}_\theta)_{\theta \in \Theta})$ where $\mathcal{O}^n$ is the observations space, $\mathbb{P}_\theta$ the parametric law of the data, Θ the set of all parameters. We denote by Θ_0, Θ_1 two disjoint subsets of Θ.

A test consists in the confrontation of the two following hypotheses on the real parameter θ of the *a priori* data law:

$$H_0 : \theta \in \Theta_0 \quad \text{(null hypothesis)} \quad \text{vs} \quad H_1 : \theta \in \Theta_1 \quad \text{(alternative hypothesis)} \tag{1.1}$$

In non-parametric framework, the reasoning is the same except that no assumption is given on the data law, the generic model can be written as $(\mathcal{O}^n, \mathbb{P})$ where $\mathbb{P}$ is unknown. If we want to perform, for example, a mean comparison, with μ (population real mean) and μ_0 (reference mean), the test can be designed like:

$$H_0 : \mu = \mu_0 \quad \text{(null hypothesis)} \quad \text{vs} \quad H_1 : \mu \neq \mu_0 \quad \text{(alternative hypothesis)} \tag{1.2}$$

But here we cannot use the data law to assess the test, so one idea is to reason conditionally to the data (with the measure $\mathbb{P}(. \mid X \in \mathcal{O}^n)$), and find a strategy to compute, from the data, this probability measure.

One of these strategies is the permutation method, developed in details below in this paper.

1.2 Test assessment

Definition 1.2.1 A test statistic Φ is a decision rule that take 0 or 1 as output and a sampled random vector as input :

$$\Phi(x) = \begin{cases} 0 & H_0 \quad \text{accepted} \\ 1 & H_0 \quad \text{rejected} \end{cases} \tag{1.3}$$

In this case, doing a test Φ consists in determining a region of rejection, which corresponds to the set: $R = \{x \mid \Phi(x) = 1\}$.

If the observed sample x belongs to this region, we reject the null hypothesis H_0. The criteria that define the region of rejection of a test Φ are type I error or (the probability of rejecting H_0 while it is true), and the power of the test (the probability of rejecting H_0 every time it is false).

We always want to control the type I error so that it does not exceed a given significance level α and at the same time maximize the power of the test.

The following graph, for a given example of hypothesis testing H_0 vs H_A, allows the visualization of the latter quantities, the type-I error and the power of the test:

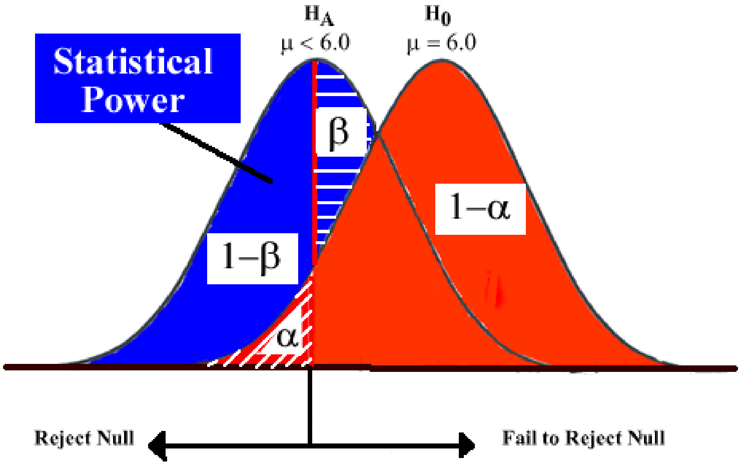


Figure 1.1: Example of power graph representing type I error (α) and the power of the test $(1 - \beta)$, (β) is the type II error.

The main goal is therefore, given a significance level α_0, to minimize the type-I error denoted by α s.t :

$$\alpha \triangleq \mathbb{P}(x \in R \mid H_0) \tag{1.4}$$

under the constraint :

$$\alpha \leq \alpha_0$$

And as a second step to maximize the power of the test defined by, more formally by a function of the rejection zone R, denoted by π:

$$\pi(R) \triangleq \mathbb{P}(x \in R \mid H_1) = 1 - \beta(R) \tag{1.5}$$

Remark This definition of the test power implicitly gives the definition of the type-II error β for a given rejection region R :

$$\beta = \beta(R) \triangleq \mathbb{P}(x \notin R \mid H_1) \tag{1.6}$$

Generally, a test Φ is associated with a random variable called the statistic of the test T. And we define in an equivalent way, the rejection region by:

$$R = \{x \mid \Phi(x) = 1\} = \{x \mid T(x) \in S\}.$$

with S a certain set of 'forbidden' values of T that leads to the rejection of the null.

Let us fix f_0 the density of x under H_0. With these constraints we can define the rejecting region R.

Definition 1.2.2
With a fixed significance level α and given set S of rejection values of the statistic T, the rejection region can be defined as follows:

$$\mathbb{P}(x \in R \mid H_0) = \mathbb{P}(T(x) \in S \mid H_0) = \int \Phi(x) f_0(x) dx \tag{1.7}$$

From these notations, we can retrieve the two following definitions:

Definition 1.2.3
if

$$\text{if} \int \Phi(x) f_0(x) dx \leq \alpha. \tag{1.8}$$

then such a test is called *conservative*

Definition 1.2.4
if

$$\int \Phi(x) f_0(x) dx = \alpha \tag{1.9}$$

then such a test is called *exact*

1.3 P-value approach

In applications, a common practice is to refer to a critical threshold α^*, usually denoted by p called the **p-value** of the test. This involves giving, for a fixed sample and a fixed test, the higher level value α, for which the H_0 is not rejected by the test. The given critical threshold allows to find the set of all α such that the H_0 is rejected (or not rejected) at the α-level, without recalculating for each particular α.

Definition 1.3.1
For a given sampled vector valued x, the value : $p \triangleq \alpha^* = \alpha^*(x)$
is called p-value if :

$$\begin{cases} p \leq \alpha & H_0 \quad \text{rejected} \\ \alpha < p & H_0 \quad \text{accepted} \end{cases} \tag{1.10}$$

In general, we have a statistic $T_0(x)$ computed with the observed vector x, and a reference value depending on α, called the *critical value*, and denoted by T_α.
T_α is the quantile such that $\mathbb{P}(T \geq T_\alpha) = \alpha$ (or $\mathbb{P}(T$ 'worst than' $T_\alpha) = \alpha$, more generally). It corresponds to the α-value s.t a 'worst' value of $T_0(x)$ (i.e greater/lowest value than $T_0(x)$ or than $|T_0(x)|$, depends on the test design) leads to the rejection of H_0 at level α.

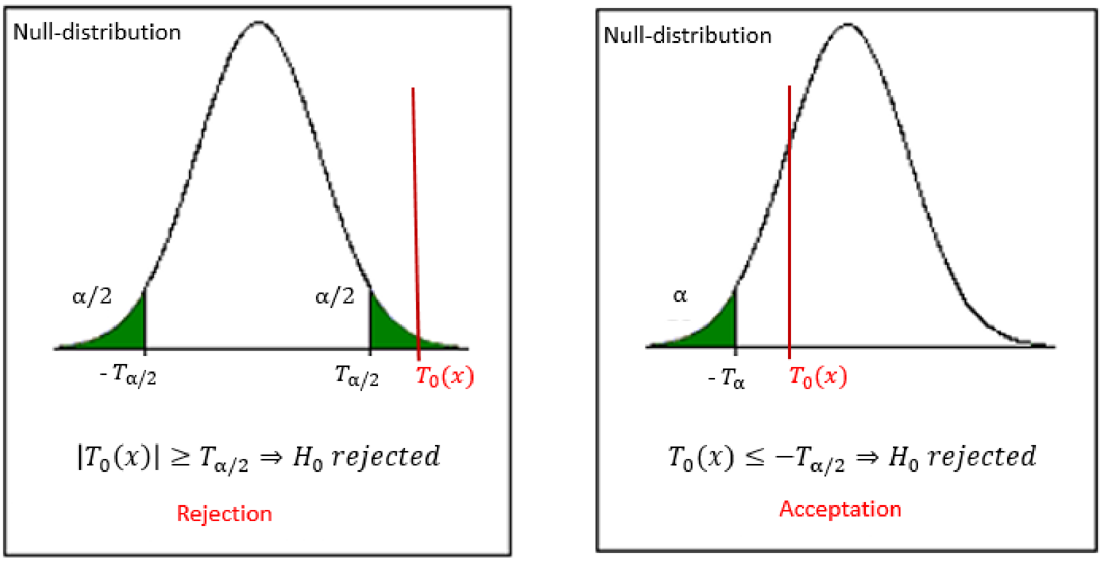


Figure 1.2: Two-tail test on the left, One-tail test on the right

The critical value T_α delimits the rejection region, at level α, for the computed statistic $T_0(x)$.

The Figure 1.2 shows the location of the critical value on an H_0 distribution (in parametric case), at a certain level α. If the computed statistic $T_0(x)$ falls in the green region the null hypothesis is rejected at level α.

From this visualization we can give the following equivalent definition of the p-value p of a test:

$$p \triangleq \mathbb{P}(T(X) \quad \text{"is more extreme than"} \quad T_0(x) \mid H_0) \tag{1.11}$$

i.e with the examples of the Figure 1.2:

$$p = \mathbb{P}(|T(X)| \geq T_0(x) \mid H_0) \tag{1.12}$$

$$p = \mathbb{P}(T(X) \leq -T_0(x) \mid H_0) \tag{1.13}$$

Chapter 2

Permutation tests

2.1 General principles

All permutations tests derive from the same fundamental principle. This is an intuitive principle as following: Under the null hypothesis, all the permutations of the observation sample have the same probability of being extracted. We require, that under the null hypothesis, all the observations being exchangeable. So, a permutation test consists in a comparison between the value of the statistic T without permutations on the dataset and the value T^* of the statistic with a permuted dataset. Critical probability are computed considering the conditional law w.r.t to the vector of the order statistic observed. And this conditional law provides an empirical distribution from which the test observed statistic is located.

To apply the permutation tests, the necessary and sufficient condition, under the null hypothesis (i.e the exchangeability of the observations) must be verified. Otherwise, their use becomes unappropriated and wrong.

In order to detail what has just been stated, we introduce some definitions and concepts, and we start by defining variables exchangeable.

Definition 2.1.1 We define Π the set of all permutations on the integer set $\{1,..,n\}$. A set of random variables $(X_1,...,X_n)$ is said exchangeable if for every permutation $\pi = (\pi_1,..,\pi_n) \in \Pi$ we have:

$$(X_1,..,X_n) \overset{d}{=} (X_{\pi_1},..,X_{\pi_n}) \tag{2.1}$$

and we said that a density function f is likelihood invariant w.r.t permutations if for every $\pi = (\pi_1,..,\pi_n) \in \Pi$ we have:

$$f(x_1,..,x_n) \overset{d}{=} f(x_{\pi_1},..,x_{\pi_n}) \tag{2.2}$$

Example 2.1.1
If the random variables $(X_1, .., X_n)$ are i.i.d then they clearly are exchangeable, because:

$$f(x_1, .., x_n) = \prod_{k=1}^{n} f(x_k) = f(x_{\pi_1}, .., x_{\pi_n})$$

The problem considered is that of testing a null hypothesis H_0, concerning the probability distribution features of a sample, against an H_1 alternative concerning this law.

First, we start by defining the order statistic and the vector of the ranks.

Definition 2.1.2 Let be a random vector $X = (X_1, ..., X_n)$ the order k statistic, denoted by $X_{(k)}$, is the k^{th} lowest value of X. The order statistic is defined by $X_{(.)} = (X_{(1)}, .., X_{(n)})$

Definition 2.1.3 The vector of ranks, denoted by $r = (r_1, .., r_n)$, is defined from the order statistic $X_{(.)}$ of the vector random X, as follows:

$$\forall i \in [1, .., n], \quad X_i = X_{(r_i)} \tag{2.3}$$

We present now an important theorem for non-parametric inference, and in particular the rank test (and any test that is based on the ranks of the data).

Theorem 2.1.1. *Let $X = (X_1, ..., X_n)$ a random vector, f_0 a density of X under H_0 that is invariant w.r.t to permutations. The rank vector r and the order statistic $X_{(.)}$ are independent. And r is uniformly distributed on Π, i.e :*

$$\forall \pi \in \Pi, \quad \mathbb{P}(r = \pi) = \frac{1}{n!} \tag{2.4}$$

According to this theorem, so thanks to the independence of the rank vector and the order statistic, we can write:

$$\mathbb{P}(X = x_\pi \mid X_{(.)} = x_{(.)}) = \mathbb{P}(r = \pi \mid X_{(.)} = x_{(.)}) = \frac{1}{n!} \tag{2.5}$$

We denote by $\Pi(x) \triangleq \{x_\pi \mid \pi \in \Pi\}$, the set that contains all the permutations of the observed vector x. We have $Card\{\Pi(x)\} = \frac{1}{n!}$.

The equality (2.5) is interpreted as follows: keeping fixed the observed vector $x_{(.)}$ of the order statistic, all the $n!$ permutations of vector values $x_{(.)}$ have the same likelihood. We conclude that, under H_0 which supposes the exchangeability of the variable, the conditional distribution $\mathbb{P}(. \mid X_{(.)} = x_{(.)})$ is uniform on $\Pi(x)$. In other words, under H_0, we can consider that the observed x has the same probability to be chosen as any other $\Pi(x)$ element, because all $\Pi(x)$ points are equiprobable.

2.2 Formal definitions of a permutation test

The formal definition of a permutation test can be presented in two ways: in terms of its critical function Φ, or in terms of statistics T (via the permutation principle describe below).

Our study focuses on the second aspect with the computation of one or more statistics for the test. But for sake of clarity we will present the two aspects, by mean of the two following definitions:

We set $N = Card\{\Pi(x)\}$

Definition 2.2.1 An α-level permutation test Φ expressed as function of the order statistic $x_{(.)}$ and the rank $\pi = (\pi_1, .., \pi_n)$ i.e $\Phi(x) = \Phi^*(x_{(.)}, \pi_1, .., \pi_n)$ and s.t

$$\frac{\sum_\pi \Phi^*(x_{(.)}, \pi_1, .., \pi_n)}{N} = \alpha \tag{2.6}$$

For a non-randomized test, Φ take only two discrete values 0 or 1. So we can't always get an α-level test. So the previous definition holds for $\alpha \in \{\frac{k}{N} \mid k = 1, .., N\}$, to guarantee the exactness of the test.

Remark In randomized test, the test Φ can take a continuum of values and can be define as $\Phi(x) = \mathbb{P}(x \notin R)$.

Now the approach with the computation of a statistic for the test, and we define the following permutation principle:

Definition 2.2.2 If we define the rejection region R with a statistic $T(x)$, this statistic must satisfy the **permutation principle** that is equivalent to a uniform draw under H_0. Let $\mathcal{T} = \{t \mid t = T(x_\pi), \quad x_\pi \in \Pi(x)\}$.
we must have:

$$\forall t \in \mathcal{T}, \quad \mathbb{P}(T(X) = t \mid X_{(.)} = x_{(.)}) = \frac{1}{n!} \tag{2.7}$$

In other words, the conditional law of the statistic T is uniform on $\mathcal{T}$. And then for a given significance level $\alpha = \frac{k}{N}$ with $k \in [1, .., N]$, the critical region R will contain k elements of $\mathcal{T}$.
For example the k greatest, or the $k/2$ greatest and the $k/2$ lowest depending on the test type (one-tail, two-tails).

To recap all the main explicated points about permutation tests:

- Permutation tests are conditional procedures where conditioning is done with respect to the vector of order statistic $X_{(.)}$ and this gives a discrete law on the permutation space $\Pi(x)$.

- Permutation tests are law-free tests because the law of the statistic does not depend on the invariant density f_0 of X under H_0, which can be partially, or completely unknown.

- The permutation tests are always exact or conservative since the probability of the type-I error is controlled for all possible samples of the invariant density f_0.

2.2.1 The steps to perform a permutation test

1. We begin with the analysis of the problem: define the null hypothesis, the alternative hypothesis, and the suppositions of the test.

2. Thereafter, we choose the most appropriate test statistic to distinguish the two hypotheses, and the statistic is calculated for the observations available.

3. Recalculate the test statistic for all possible permutations of the initial observations; note that it is necessary to **specify how to do the permutation according to each problem**.

4. Finally, we make the decision: accept or reject the null hypothesis using the law of the statistic as a guide. It is better to see if the statistic observed is extreme or not compared to the empirical law of the statistic of the test generated by permutations.

Remark : Considering steps (1) and (3), there are some knotty points with permutations test:
First, the statistic for the test under H_0, particularly in multivariate situations, may have forms that are difficult to express and calculate. Then, if the sample size is not small enough, a direct calculation of the law by enumeration of all permutations becomes impracticable because of the large number of permutations to be made.
For example, even if we compare the expectations of two samples of sizes $m = 7$ and $n = 5$, the number of possible permutations is $\binom{5+7}{7} = 30240$. Finally, an asymptotic approximation of the test statistic is not always appropriated, unless the number of observations is sufficiently great.

2.2.2 Monte-Carlo method

Although, in principle, it is always possible to make an exact calculation by listing · permutations, in practice we use approaches that are more efficient from the point of view of calculation. The most used method is the Monte-Carlo method (MC).

This technique involves drawing a random sample of size B, B being big, among all the possible permutations and apply the test only on this sample, instead of considering the complete law. Indeed, with a large sample, the distribution under H_0 can be well approximated by the Monte Carlo technique. So, we estimate the p-value with:

$$p - value = \frac{k^*(x)}{B} \tag{2.8}$$

where $k^*(x)$ is the number of times that the test statistic is more extreme than the statistic observed among the B permutations retained in the sample.

2.2.3 Concrete example: Comparison test of the means of two populations

This example resumes the test carried out in the R-Package function, `IWT2.R`.

So in order to illustrate the permutation principle, we take the example of comparison of two independent samples. We have two groups of observations $\{x_i, \quad i = l, .., n\}$ and $\{y_i, \quad i = l, .., m\}$ with sizes n and m, and we want to test $H_0 : \mu_x = \mu_y$ against the alternative $H_1 : \mu_x \neq \mu_y$, for example.

We choose as statistic $T(X, Y) = |\overline{Y} - \overline{X}|$. To make a permutation test, it is necessary to generate the conditional law of the statistic T under H_0.

First, we calculate the observed value $T_{obs} = |Y_{obs} - X_{obs}|$. We then take all the $(m + n)$ elements of the observed vectors $(x_1, ..., x_n)$ and $(y_1, ..., y_m)$ and we create the vector $(z_{(1)}, .., z_{(m+n)})$ of the order statistics of the $(m + n)$ observations. Afterwards, we mix (randomly switch) and randomly select two new groups with m and n elements respectively. The number of distinct values of T is equal to $\binom{m+n}{m}$ because if we switch the observations within each group the value of T remains the same.

If the sample sizes are small, we can do all the possible permutations. Otherwise, we obtain a good result by performing only a certain number B of permutations, for example $B = 2000$, chosen randomly and without discount among all those that are possible.

After generating the values of the statistic T , the values are sorted in increasing order, to obtained the ordered vector $(T_{(1)}, .., T_{(M)})$. For a significance level α, the test rejects H_0 if $T_{obs} \geq T_{(k)}$, where $T_{(k)}$ is the quantile that corresponds to $k = \lfloor B(1-\alpha) \rfloor$, the integer part of $B(1-\alpha)$. This means that we reject H_0 if the observed value T_{obs} is sufficiently extreme that it cannot be due only to chance.

Example 2.2.3.1
Let's do a numerical example to clarify the method.
The observed values are $x = (4.3, 6.0, 3.6)$ and $y = (7.4, 5.5, 6.2)$.
The order statistic vector for combined sample is $z_{(.)} = (3.6, 4.3, 5.5, 6.0, 6.2, 7.4)$.
The possible number of permutations is $B = \binom{6.0}{3} = 20$.
The Table 2.1 below, presents the different possible values x and y,
given $z_{(.)} = (3.6, 4.3, 5.5, 6.0, 6.2, 7.4)$, as well as the discrete law.

The observed value is $T_{obs} = t_{obs} = |\overline{y_{obs}} - \overline{x_{obs}}| = 6.4 - 4.63 = 1.7667$ and for each $t \in \mathcal{T}$ valued in column 8 of the Table 2.1, we have :

$$\mathbb{P}(\ T(X,Y) = t \mid H_0,\ Z_{(.)} = (3.6, 4.3, 5.5, 6.0, 6.2, 7.4)) = \frac{1}{20} \tag{2.9}$$

Indeed, with the theorem 2.1.1, under H_0 all the allocations of the values of the vector $z_{(.)} = (3.6, 4.3, 5.5, 6.0, 6.2, 7.4)$ between the x and the y values, have the same likelihood: 1/20.

For a significance level $\alpha = 0.05$ we have $k = B(1-0.05) = 20(1-0.05) = 19$.
Since the observed value $T_{obs} = 1.7667$ is greater than the critical value $T_{19} = 1.733$, we can reject the H_0.

Table 2.1: Possible permutations of the two groups of size m= 3 and n= 3

	x_1	x_2	x_3	y_1	y_2	y_3	$T(x,y)$	Rank
1	3.6	4.3	5.5	6.0	6.2	7.4	2.067	20
2	3.6	4.3	6.0	5.5	6.2	7.4	1.733	19
3	3.6	4.3	6.2	5.5	6.0	7.4	1.6	18
4	3.6	4.3	7.4	5.5	6.0	6.2	0.8	17
5	3.6	5.5	6.0	4.3	6.2	7.4	0.933	16
6	3.6	5.5	6.2	4.3	6.0	7.4	0.800	15
7	3.6	5.5	7.4	4.3	6.0	6.2	0	14
8	3.6	6.0	6.2	4.3	5.5	7.4	0.467	13
9	3.6	6.0	7.4	4.3	5.5	6.2	-0.333	12
10	3.6	6.2	7.4	4.3	5.5	6.0	-0.333	11
11	4.3	5.5	6.0	3.6	6.2	7.4	-0.467	l0
12	4.3	5.5	6.2	3.6	6.0	7.4	0.333	9
13	4.3	5.5	7.4	3.6	6.0	6.2	-0.467	8
14	4.3	6.0	6.2	3.6	5.5	7.4	0	7
15	4.3	6.0	7.4	3.6	5.5	6.2	-0.800	6
16	4.3	6.2	7.4	3.6	5.5	6.0	-0.933	5
17	5.5	6.0	6.2	3.6	4.3	7.4	-0.800	4
18	5.5	6.0	7.4	3.6	4.3	6.2	-1.6	3
19	5.5	6.2	7.4	3.6	4.3	6.0	-1.733	2
20	6.0	6.2	7.4	3.6	4.3	5.5	-2.067	1

Chapter 3

Fundamentals of the Interval-Wise Testing Procedure

The Interval-wise testing procedure, the so-called $IWTP$, is fundamentally a statistical test, based on functional data. So the main goal is to test some features of functional data along their definition domain. These features tested, are embedded in the hypotheses denoted, as usual, by H_0 (the null) versus H_1 (the alternative).

3.1 Theoritical principles of the IWT

The IWTP is a local inferential method that consists in a domain selection (intervals selection) of the underlying functional data to be tested, in order to know where in the domain the null hypothesis H_0 has to be rejected. This type of test relies directly on the functional data themselves (which can have been smoothed beforehand), without any projection of the data on any functional basis (e.g Fourier basis, B-splines basis). The outputs of the procedure are basically what we will call:
The **Unadjusted p-value** and the **Adjusted p-value**.

So one can ask how this kind of test can be implemented? What is the cost of such a test? With which precision the test is expected to be performed? We will consider, as it is done in the package, tests performed on functional data.

Suppose that, based on the observation of a set of L^2 random functions over the domain $T = (a, b) \subset \mathbb{R}$. Our goal is to test two hypotheses H_0 vs H_1, relative to some features, on the whole domain, the classic:

$$H_0 \quad vs \quad H_1 \tag{3.1}$$

To do so we will considered tests on sub-intervals of the domain, i.e we will reduce the test to many sub-tests on each interval as it is described below.

Let $I =\subset T$ be an interval or a complementary interval of the form:
$I = (t_1, t_2)$ or $I = T \setminus (t_1, t_2)$ respectively, with $a \leq t1 < t2 \leq b$.

And let us define H_0^I and H_1^I as being the restriction of the null and alternative hypotheses on I, respectively.

Example 3.1.1
With a two-populations mean comparison test, the reduced test on I can be written as follows:

$$H_0^I : \mu_1^I = \mu_2^I \quad vs \quad H_1^I : \mu_1^I \neq \mu_2^I \tag{3.2}$$

The corresponding p-value of this I-reduced test is denoted by, p^I.

Let's now give two fundamental definitions:

Definition 3.1.1 $\forall t \in T$ we define the unadjusted p-value $p(t)$:

$$p(t) = \underset{I \to t}{\text{limsup}}\ p^I \tag{3.3}$$

$I \to t$: when both extremes of I converges to t.

Definition 3.1.2 $\forall t \in T$ we define the adjusted p-value $\tilde{p}(t)$:

$$\tilde{p}(t) = \sup_{I \ni t} p^I \tag{3.4}$$

So it is the supremum of all p-values p^I over all I (where I can be both interval or complementary interval) s.t $t \in I$.

The unadjusted p-value function $p(t)$ and the adjusted p-value function $\tilde{p}(t)$ present different inferential properties with respect to both type-I error control and consistency.

Here we set two fundamental theorems that show that the IWTP is guaranteeing a control on the type-I error (an interval-wise control)

Theorem 3.1.1. *The unadjusted p-value function $p(t)$ provides a control of the point-wise error rate:*

$$\forall \alpha \in (0,1),\ \forall t \in T\ s.t\ \exists I, t \in I : H_0^I\ true \Rightarrow\ \mathbb{P}(p(t) \leq \alpha) \leq \alpha \tag{3.5}$$

Theorem 3.1.2. *The adjusted p-value function $\tilde{p}(t)$ provides a control of the interval-wise error rate:*

$$\forall \alpha \in (0,1),\ \forall t \subseteq T : H_0^I\ true \Rightarrow\ \mathbb{P}(\forall t \in I,\ \tilde{p}(t) \leq \alpha) \leq \alpha \tag{3.6}$$

To sum up, the first theorem is saying that when assessing the test and making a decision by thresholding the unadjusted p-value $p(t)$ by α for every point t of the domain, then we have that the point-wise type-I error (H_0 is rejected in t, whereas H_0 is true in t) is controlled ($\leq \alpha$) for this t.

On the other side, the second theorem is saying that when assessing the test and making a decision by thresholding the adjusted p-value $\widetilde{p}(t)$ by α for every interval I of the domain such that $t \in I$, then we have that the interval-wise type-I error (H_0 is rejected in the whole I, whereas H_0 is true in I) is controlled ($\leq \alpha$).

It is also worth noticing, that the adjusted and unadjusted p-values are consistent, it is to say, when the sample size $n \to +\infty$, the type-I error rate goes to 0.

3.2 Details and Implementation of the IWTP

The time domain where are defined the functional data, is discretized into p evaluation points. Each of the data function is evaluated on this time-grid.

The goal of this section is to show up how the adjusted and unadjusted p-values of functional test are computed in each function of the package.

3.2.1 Unadjusted p-values computation

Here in this subchapter we will consider, the p univariate tests (marginal tests), to be performed on each points of the evaluation grid, denoted by : $H_i,\ i = 1, .., p$.

We define the T_0 statistic computed on the original data, as the vector of each statistic of the p univariate tests, denoted as follows:

$$T_0 = \begin{bmatrix} TH_1 & \dots & \dots & \dots & TH_p \end{bmatrix} \tag{3.7}$$

And then, the matrix T where each row i corresponds to the vector of the statistic of the p univariate test, computed on a σ_i-permuted dataset.

$$T = \begin{bmatrix} TH_1^{\sigma_1} & \dots & \dots & \dots & TH_p^{\sigma_1} \\ \vdots & \dots & \vdots & \dots & \vdots \\ \vdots & \dots & TH_j^{\sigma_i} & \dots & \vdots \\ \vdots & \dots & \vdots & \dots & \vdots \\ TH_1^{\sigma_B} & \dots & \dots & \dots & TH_p^{\sigma_B} \end{bmatrix} \tag{3.8}$$

For every $i \in [p:1]$ (more generally $i \in [p:\texttt{maxrows}]$, where `maxrows` is a scale parameter that determines the depth of the interval-wise testing procedure), and for every $j \in [1:p]$, we define :

$$T_{0,comb} = TH_j + \cdots + TH_{p-i+j} \in \mathbb{R}. \tag{3.9}$$

$T_{0,comb}$ is computed (and updated) at each iteration, i.e for each couple (i,j) . It computes the multivariate test on the original dataset (without any permutation).

$$T_{comb} := \begin{bmatrix} TH_j^{\sigma_1} & + & \dots & + & TH_{p-i+j}^{\sigma_1} \\ \vdots & & \vdots & & \vdots \\ \vdots & & \vdots & & \vdots \\ \vdots & & \vdots & & \vdots \\ TH_j^{\sigma_B} & + & \dots & + & TH_{p-i+j}^{\sigma_B} \end{bmatrix} \in \mathbb{R}^{B\times 1}. \tag{3.10}$$

So, as for $T_{0,comb}$, a new value of T_{comb} is computed (and updated) at each iteration, i.e for each couple (i,j). For a given couple (i,j), each component $(T_{comb})_k$ of the B-dimensional vector T_{comb}, is composed by the sum of the j^{th} statistic, $TH_j^{\sigma_k}$, associated to the k^{th} permutation σ_k; until the $(p-i+j)^{th}$ statistic, $TH_{p-i+j}^{\sigma_k}$, for the same permutation σ_k.

Remark It is relevant to note that in the T_{comb} vector ($T_{0,comb}$ scalar) the index j characterizes the **starting point** of the sum, so beginning with the test number j. On the other side, the index i characterizes the **size** of the sum, i.e the size of the combination of successive tests involved in the sum.

From this observation we can deduce that varying i and j we obtain all the possible tests combinations (starting from every point and with all possible sizes). The next step is the construction of the p-values of the tests $(H_j \cap ... \cap H_{p-i+j})_{(i,j)}$, so one p-value, denoted by $p_{l,m}$, for each couple $(l,m) \in [1..p]^2$ and associated for a given test $(H_m \cap ... \cap H_{p-l+m})$. Here intervene the Monte-Carlo procedure over B (permutations number) as the intrinsic method for the approximate computation of these p-values.

To do so, we compare for each couple (i,j) the two quantities:

$$T_{comb} \geq T_{0,comb} \tag{3.11}$$

i.e

$$\begin{bmatrix} TH_j^{\sigma_1} & + & \dots & + & TH_{p-i+j}^{\sigma_1} \\ \vdots & & \vdots & & \vdots \\ \vdots & & \vdots & & \vdots \\ \vdots & & \vdots & & \vdots \\ TH_j^{\sigma_B} & + & \dots & + & TH_{p-i+j}^{\sigma_B} \end{bmatrix} \geq TH_j + ... + TH_{p-i+j} \tag{3.12}$$

Remark This encloses the basic comparison of the two statistics on the permuted data T^B and on the original dataset T_0 : $T^B \geq T_0$

To create the p-values, i.e the probability that the permuted statistic exceeds the original statistic, we will compute an **empirical probability**.

To do such computation it is necessary to count out the number of the occurrences:

$$(T_{comb})_k \geq T_{0,comb}(k \in [1..B]) \tag{3.13}$$

We encode this occurrence by 1 and by 0 for the opposite, and retrieve an B-dimensional vector of 0 and 1 :

$$V := \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} \tag{3.14}$$

Finally we get all the p-values :

$$p_{i,j} = \frac{1}{B} \sum_{k=1}^{B} V_k \tag{3.15}$$

These p-values are stored into a combination matrix $P := [p_{i,j}]_{(i,j)} \in \mathbb{R}^{p \times p}$

$P :=$

$p_{1,1}$	$\dots$	$\dots$	$p_{1,p}$
$\vdots$	$\dots$	$\dots$	$\vdots$
$p_{p,1}$	$\dots$	$\dots$	$p_{p,p}$

The last row in gray corresponds to p-values of marginal tests (or univariate tests) i.e every test: H_j (last row jth column). We can establish the following definitions:

Definition 3.2.1.1 We call the set $\{p_{p,1}, \dots, p_{p,p}\}$, the set of **unadjusted p-values**.

3.2.2 Adjusted p-values computation

Now we will undertake the computation of the adjusted p-values, that guarantees the interval-wise control on the type-I error provided the IWT procedure.

Once we get P, we create a concatenate version of this p-values matrix, denoted by:

$$P_{2\times} := \big[P \quad | \quad P\big] \in \mathbb{R}^{p\times 2p} \tag{3.16}$$

$$P_{2\times} := \begin{array}{|c|c|c|c|c|c|c|c|} \hline p_{1,1} & \dots & \dots & p_{1,p} & p_{1,1} & \dots & \dots & p_{1,p} \\ \hline \vdots & \dots & \dots & \vdots & \vdots & \dots & \dots & \vdots \\ \hline p_{p,1} & \dots & \dots & p_{p,p} & p_{p,1} & \dots & \dots & p_{p,p} \\ \hline \end{array}$$

Marginal p-values are coloured in gray. Then we re-affect a row-reverted version to $P_{2\times}$:

$$P_{2x} \leftarrow P_{2x}[, 2p : 1]$$

That is:

$$P_{2\times} := \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline p_{1,p} & \dots & \dots & \dots & p_{1,1} & p_{1,p} & \dots & \dots & \dots & p_{1,1} \\ \hline \vdots & \dots & \dots & \dots & \vdots & \vdots & \dots & \dots & \dots & \vdots \\ \hline \vdots & \dots & \dots & \dots & \vdots & \vdots & \dots & \dots & \dots & \vdots \\ \hline \vdots & \dots & \dots & \dots & \vdots & \vdots & \dots & \dots & \dots & \vdots \\ \hline p_{p,p} & \dots & \dots & \dots & p_{p,1} & p_{p,p} & \dots & \dots & \dots & p_{p,1} \\ \hline \end{array}$$

The cause of this row-reverting is just to have the right tests combinations, for every column k, i.e for every k^{th} evaluated point of our functional data.

We define the matrix P_{adj}, visually, as follows:

$$(P_{adj})_{i,j} := max[\text{Blue cells}]$$

$$P_{2\times} := \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline p_{1,p} & \dots & \dots & \dots & p_{1,1} & p_{1,p} & \dots & \dots & \dots & p_{1,1} \\ \hline \vdots & \dots & \dots & \dots & \vdots & \vdots & \dots & \dots & \dots & \vdots \\ \hline \vdots & \dots & (i,j) & \dots & \vdots & \vdots & \dots & (i,p-i+j) & \dots & \vdots \\ \hline \vdots & \dots & \dots & \dots & \vdots & \vdots & \dots & \dots & \dots & \vdots \\ \hline p_{p,p} & \dots & (p,j) & \dots & p_{p,1} & p_{p,p} & \dots & \dots & \dots & p_{p,1} \\ \hline \end{array}$$

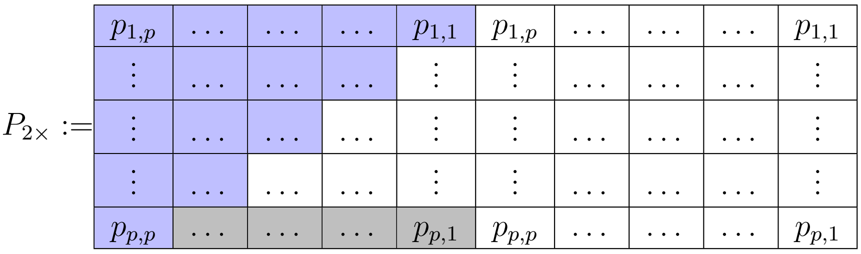
Explanation : Let k be an evaluation point index, of the grid. Then we define here the so-called **m-variate test** for the point k.

- univariate test : H_0^k
- bivariate tests :$H_0^{k-1} \cap H_0^k,\ H_0^k \cap H_0^{k+1}$
- 3-variate tests :$H_0^{k-2} \cap H_0^{k-1} \cap H_0^k,\ H_0^{k-1} \cap H_0^k \cap H_0^{k+1},\ H_0^k \cap H_0^{k+1} \cap H_0^{k+2}$
  $\vdots$
- p-variate test (global test):$\underset{i=1..p}{\cap} H_0^i$

So for a row i, we compute for each column j the maximum of the p-values, for univariate tests (row p), bivariate tests (row $p-1$),.., $(p-i+1)$-variate tests (row i), that all contain the evaluation point j.

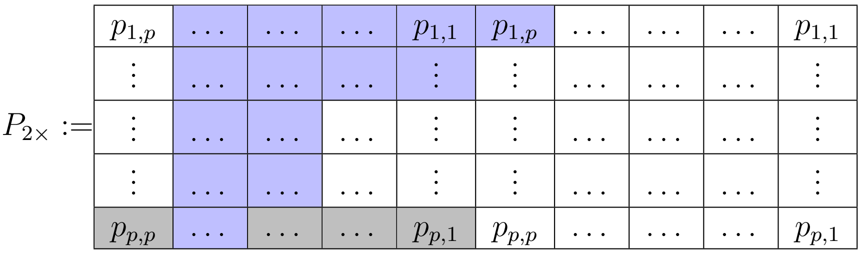
A practical example is given below, to explain this procedure phase, when it comes to the last row computation. The last row (until `maxrows`, in reality) assessment, in the computation of P_{adj}, will provide the adjusted p-values.

And so at the end, the algorithm that computes the p-value adjusted matrix is composed of p steps. At each step the blue triangle is moving to the right side, with path 1, on the gray rail (i.e starting from $p_{p,p}$ until $p_{p,1}$).
Furthermore, at each step the algorithm, moving up on the blue lines, computes the adjusted p-value corresponding to the blue block or more precisely to the corresponding evaluation point j. And at each step, this adjusted p-value is stored in the top left-hand corner of the blue triangle (in the right angle).



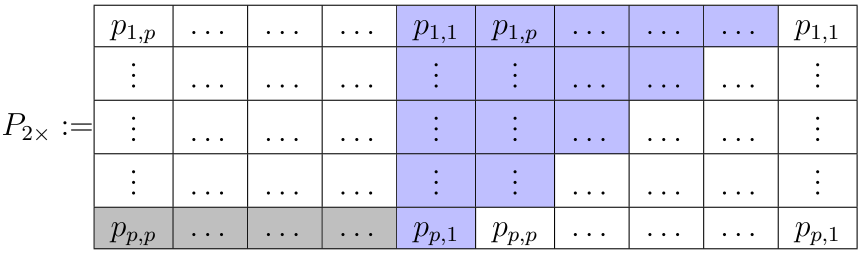
$$P_{2\times} := \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline p_{1,p} & \dots & \dots & \dots & p_{1,1} & p_{1,p} & \dots & \dots & \dots & p_{1,1} \\ \hline \vdots & \dots & \dots & \dots & \vdots & \vdots & \dots & \dots & \dots & \vdots \\ \hline \vdots & \dots & \dots & \dots & \vdots & \vdots & \dots & \dots & \dots & \vdots \\ \hline \vdots & \dots & \dots & \dots & \vdots & \vdots & \dots & \dots & \dots & \vdots \\ \hline p_{p,p} & \dots & \dots & \dots & p_{p,1} & p_{p,p} & \dots & \dots & \dots & p_{p,1} \\ \hline \end{array}$$

Figure 3.1: First step of the algorithm.



$$P_{2\times} := \begin{array}{|c|c|c|c|c|c|c|c|c|c|}\hline p_{1,p} & \dots & \dots & \dots & p_{1,1} & p_{1,p} & \dots & \dots & \dots & p_{1,1} \\ \hline \vdots & \dots & \dots & \dots & \vdots & \vdots & \dots & \dots & \dots & \vdots \\ \hline \vdots & \dots & \dots & \dots & \vdots & \vdots & \dots & \dots & \dots & \vdots \\ \hline \vdots & \dots & \dots & \dots & \vdots & \vdots & \dots & \dots & \dots & \vdots \\ \hline p_{p,p} & \dots & \dots & \dots & p_{p,1} & p_{p,p} & \dots & \dots & \dots & p_{p,1} \\ \hline \end{array}$$

Figure 3.2: Second step of the algorithm.



$$P_{2\times} := \begin{array}{|c|c|c|c|c|c|c|c|c|c|}\hline p_{1,p} & \dots & \dots & \dots & p_{1,1} & p_{1,p} & \dots & \dots & \dots & p_{1,1} \\ \hline \vdots & \dots & \dots & \dots & \vdots & \vdots & \dots & \dots & \dots & \vdots \\ \hline \vdots & \dots & \dots & \dots & \vdots & \vdots & \dots & \dots & \dots & \vdots \\ \hline \vdots & \dots & \dots & \dots & \vdots & \vdots & \dots & \dots & \dots & \vdots \\ \hline p_{p,p} & \dots & \dots & \dots & p_{p,1} & p_{p,p} & \dots & \dots & \dots & p_{p,1} \\ \hline \end{array}$$

Figure 3.3: Last step of the algorithm.

Remark The last column is redundant and is not used in the procedure. It can be cut when encoding the matrix.

Example : First step of the algorithm Figure 3.1
For each i we compute the maximum p-value of the corresponding tests written below (blue cells). For $i+1$, the maximum p-value found in i is compared with the new p-values of the tests computed in $i+1$.

i=p : (univariate test)

- H_p

i=p-1 : (univariate test) ∪ (bivariate tests)

- H_p
- $H_p \cap H_1$ and $H_{p-1} \cap H_p$

⋮

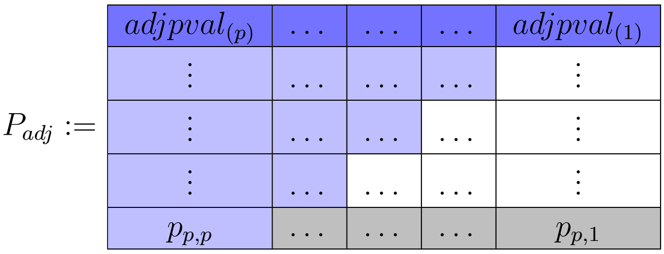
i=1 : (univariate test) ∪ (bivariate tests) ∪ ... ∪ (p-variate tests)

- H_p

- $H_p \cap H_1$ and $H_{p-1} \cap H_p$
- ...
- $H_p \cap H_1$,..,$\cap H_{p-1}$; $H_{p-1} \cap H_p \cap H_1$,..,$\cap H_{p-2}$ ... until $H_1 \cap$,..,$\cap H_p$

So here we see that the first blue triangle on the right side produces the adjusted p-value of the p^{th} evaluation point (with interval-wise control of type-I error), on the 1st row, 1st column.

The algorithm gives the P_{adj} matrix of adjusted p-values:



$$P_{adj} := \begin{array}{|c|c|c|c|c|} \hline adjpval_{(p)} & \dots & \dots & \dots & adjpval_{(1)} \\ \hline \vdots & \dots & \dots & \dots & \vdots \\ \hline \vdots & \dots & \dots & \dots & \vdots \\ \hline \vdots & \dots & \dots & \dots & \vdots \\ \hline p_{p,p} & \dots & \dots & \dots & p_{p,1} \\ \hline \end{array}$$

Figure 3.4: Adjusted p-values matrix.

From which we can retrieve our vector of the p adjusted p-values for each evaluation point:

Definition 3.2.1.2
We called the vector: `adjusted_pval` $:= [adjpval_{(1)}, ..., adjpval_{(p)}]$,
vector of the **adjusted p-values**.

3.2.3 Recycling and Non Recycling Procedure

As we will see in the next chapter, every functions of the package is passing an arguments called `recycle`. By default it is `TRUE`. It indicates the way the IWTP is performed, i.e with recycles or without.

The whole version of the IWTP explained in the previous section was the recycling one.

The basic difference between the two versions is that the `recycle=TRUE` version considered intervals and also complementary intervals, whereas the `recycle=FALSE` focuses only on intervals, excluding the complementary.

Illustration
We consider here, for sake of simplicity, a grid with just three points, i.e $p = 3$.

The following figures presented the hypotheses tested in the two versions of the IWTP:

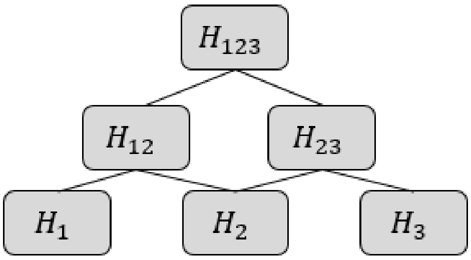


Figure 3.5: Hypotheses tested in the IWTP without recycle, presented with its typical pyramidal mesh.

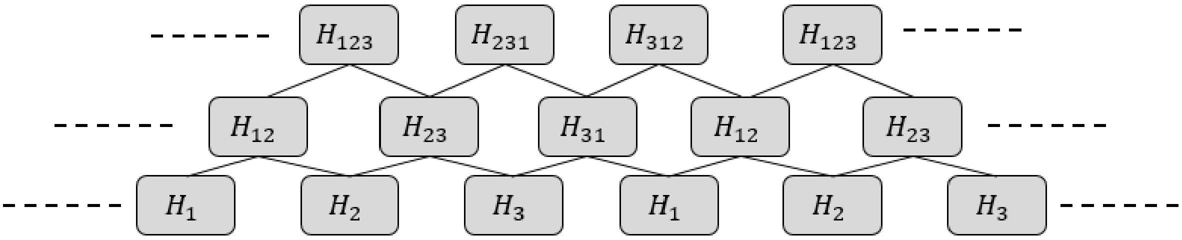


Figure 3.6: Hypotheses tested in the IWTP with recycles, presented with its typical rectangular cyclic mesh.

We remark that in the non recycling procedure, in Figure (3.5), H_2 is involved in four tests (H_2, H_{12}, H_{23}, H_{123}) whereas H_1 and H_3 are each one involved in just three tests. So this strategy favours clearly points that are located in the middle of the domain. Extreme points are less tested, and so the extremal hypotheses associated are more inclined to be rejected.

But the improved version of the IWTP with recycles, in Figure (3.6),can solve this problem. With the recycling version, for a given grid with p points, each point is tested the same number of times: $\frac{p(p+1)}{2}$.
Furthermore, the interval-wise error rate is controlled not only on the intervals but also on the complementary intervals.

Chapter 4

Package Functions Description

4.1 One- and Two-populations testing

The goal of the functions `IWT1.R` and `IWT2.R`, is to compare the means of a functional population with a reference mean (`IWT1.R`) or with the mean of another functional population (`IWT2.R`).

In formal form, given a reference mean μ_0, we test:

$$\texttt{IWT1.R} : H_0 : \mu = \mu_0 \quad vs \quad H_1 : \mu \underset{>}{\overset{<}{\neq}} \mu_0 \tag{4.1}$$

$$\texttt{IWT2.R} : H_0 : \mu_1 - \mu_2 = \mu_0 \quad vs \quad H_1 : \mu_1 - \mu_2 \underset{>}{\overset{<}{\neq}} \mu_0 \tag{4.2}$$

The principle of the method applied here, is described in detail, in section (2.2.3).

Then the IWTP and permutation tests, described above, can be applied for the interval selection of the functional domain.

Application:
We launch the following instance of the `IWT2.R`:

```
IWT2(functional_data1,functional_data2,mu=0,
            B=1000,paired=FALSE,recycle=FALSE,IW_scale=100)
```

where `functional_data1` and `functional_data2` are the functional data, as `fd` object, every day, during 22 years, in Milan and Paris, respectively.The argument μ corresponds to the mean difference to be tested.

The results can be summarized with the plots obtained with the `IWTimage.R` function:

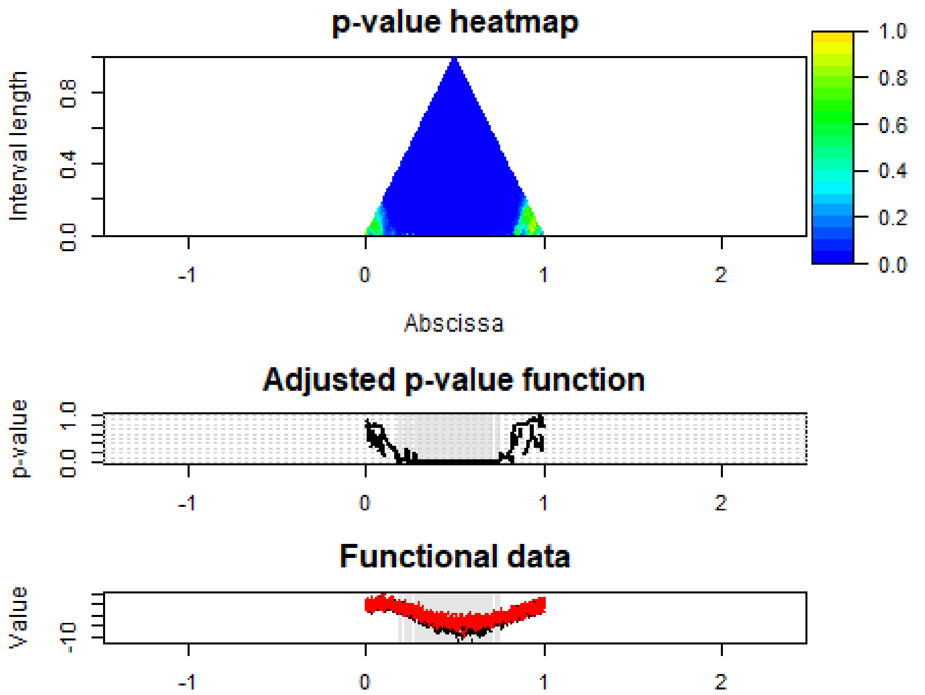


Figure 4.1: Functional data, adjusted and heatmap p-values, rejection of H_0 in gray (IWT2).

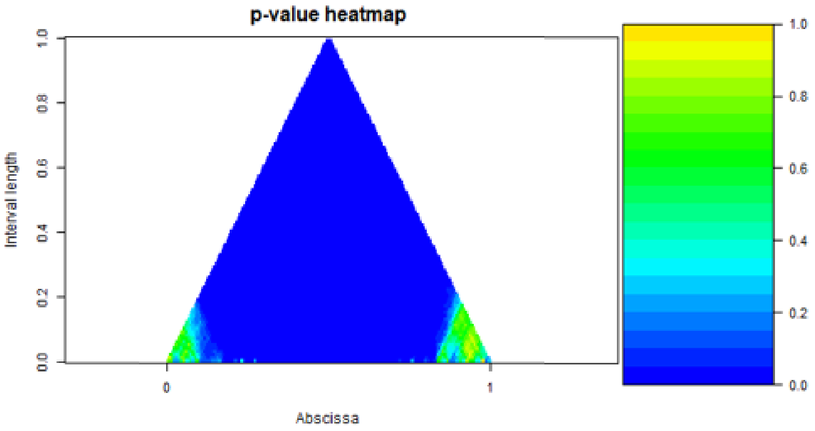


Figure 4.2: Zoom on the heatmap of p-values, p-value heat-level indicated on the right scale (IWT2).

4.2 Permutations method on Linear Models/ANOVA

The functions `IWTlm.R`, `IWTlm_FoF.R` and `IWTaov.R` aim at computing an adjusted p-value to select significant intervals where the null hypothesis can be rejected at a certain significance level, and this, for two different types of tests.

These tests are designed to test the significance of the coefficients (the effects) of the linear model, they are called **global test** and **partial test** (cf. sub-chapter below).

Functions `IWTlm.R`, `IWTlm_FoF.R` and `IWTaov.R` are also passing an argument called `method` that provides the way to assess the permutations on the data.

Two arguments can be chosen, `'responses'` (Manly's method) and `'residuals'` (Freedman and Lane method).

Example 4.2.1

```
IWTlm(formula,B = 1000, method = 'residuals')
```

4.2.1 Global test and Partial test in regression

The general linear model used here, and in the rest of the document, is the following:

$$Y_i = \beta_0 + \sum_{l=1}^{L} \beta_l.X_{l,i} + \epsilon_i \tag{4.3}$$

Ordinary, parametric assumptions would have considered that the errors were distributed according to a given law, generally the normal law. Here in the non-parametric framework, the environment is different. We will, further, consider a semi-parametric form of the linear regression, i.e expressed in terms of the parameters β_l, but with the distribution of the errors ϵ, which is not expressed.

The only assumption of the semi-parametric model is the following:

- The errors are independent and identically distributed from a zero-mean random variable.

So let's first talk about the global test. It is designed with the confrontation of the two following hypotheses:

$$H_0 : \beta_1 = \beta_2 = ... = \beta_L = 0 \quad vs \quad H_1 : \exists l \in [1,..,n], \beta_l \neq 0 \tag{4.4}$$

In other words, we want to test the hypothesis that the response variable Y is independent of the X_i variables against the assumption that Y is (linearly) linked to almost one of the X_i variables.

To assess such a test we can use the F-test statistic (Johnson, 1992):

$$T_F(Y) = \frac{\sum_{i=1}^{n}(\hat{y}_i - \overline{Y})^2/(L-1)}{\sum_{i=1}^{n}(\hat{y}_i - Y_i)^2/(n-L)} \tag{4.5}$$

Researchers are often interested in specific assumptions about partial regression coefficients, for example, for a given $k \in [1, ...L]$.:

$$H_{0,k} : \beta_k = 0 \quad vs \quad H_{1,k} : \beta_k \neq 0 \tag{4.6}$$

In fact, we want to know whether the variable X_k explains a part of the Y variability, taking into account the effect of the concomitant other variables $X_i,\ i \neq k$, which can be related to X_k.

In that case, if the classic parametric assumptions of the model are verified and if we condition with respect to the $X_i,\ i \in [1, ..., n]$, the statistical familiar parametric t-test takes the usual form described by the following equation:

$$t_{\beta_k} = \frac{\hat{\beta}_k}{\sigma(\hat{\beta}_k)} \sim T(n-L-1) \tag{4.7}$$

where $\hat{\beta}_k$ is the least squares estimator of the regression coefficient β_k, $\sigma(\hat{\beta}_k)$ is the estimator of its standard deviation and $T(n-L-1)$ represents the law of the t-Student distribution with $(n-L-1)$ degrees of freedom.

But when the conditions of the multiple linear regression models are not respected, parametric tests are not reliable.
Here and in the next part intervenes the permutation test. And so to remedy this situation, several researchers have proposed approximate the tests with permutation techniques that requires few assumptions on the linear model features.

That's the case with Freedman and Lane (1983), Manly (1997) who developed different permutation methods to assess test on linear model in non-parametric or semi-parametric framework. We will study these methods in the following. But let us focus on the particular interest of the permutation test in regression.

4.2.2 Interest of the permutation test in regression

Let's take a simplified multiple regression model with two predictors, without loss of generality.

Suppose we have measured three variables (X_1, X_2, Y) for n individuals and we observed $(x_{1,i}, x_{2,i}, y_i), i = 1, .., n$. For simplicity and without loss of generality, suppose that these variables have null expectations, $\mu_{X_1} = \mu_{X_2} = \mu_Y = 0$. The regression model multiple of Y on X_1 and X_2 is then defined by the following equation:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon \tag{4.8}$$

where Y is a random variable to be explained, X_1 and X_2 are explanatory variables, β_1 and β_2 are the partial regression coefficients and ϵ is the random error. The usual assumptions for the use of the multiple linear regression model are:

- Random errors $\epsilon_i, i = 1, ..n$ are i.i.d variables distributed according to $\mathcal{N}(0, \sigma_{\epsilon^2})$
- The errors ϵ_i are independent of the vectors of explanatory variables $(X_1, X_2), i = 1, .., n$.

On the other hand, in inference, we condition with respect to X_1 and X_2, and assume $X_{1,i}$ and $X_{2,i}$ fixed, $i = 1, .., n$. Note that we can consider a little more general assumptions as following:

- Random errors ϵ are identically distributed variables of symmetric law with respect to 0 and such that $cov(\epsilon_i, \epsilon_j) = 0, i \neq j$
- $cov(X_{1,i}, \epsilon_i) = cov(X_{2,i}, \epsilon_i) = 0, \forall i$

But when these assumptions are not verified, the former parametric tests enunciated above (F-test, t-tests) cannot hold in their usual standard parametric forms. The permutation tests can overcome that problem. Here is the main idea:

We must first, know which variables are exchangeable under the null hypothesis.

For example, in the global test, the model, under $H_0 : \beta_1 = \beta_2 = 0$, becomes:

$$Y_i = \epsilon_i \tag{4.9}$$

And then, the necessary assumption and sufficient to make the permutation test (the exchangeability of Y_i) is verified. So, under H_0, the variables X_1 and X_2 are not connected linearly and any pairing of the values of X_1 and X_2 is equiprobable.

So, by permuting the y_i values, we get $n!$ permutations possible, keeping fixed the $x_{1,i}$ and $x_{2,i}$ of the couples $\{(x_{j,1}, y_1), ..., (x_{j,n}, y_n) \mid j = 1, 2\}$ which are all equiprobable, under the null hypothesis.

So, under the null hypothesis, Y has no relation with X_1 and X_2 taken together. That means, that for fixed values $X_{1,i} = x_{1,i}$ and $X_{2,i} = x_{2,i}$, one can obtain any values of Y_i. In other words, if we take any permutations $\pi \in \Pi$, all the triples $(y_i^\pi, x_{1,i}, x_{2,i})$ where $(y_i^\pi$ corresponds to the π-permuted y_i, are equiprobable (conditionally to the order statistics of $y_i, i = 1, ..n)$.

After generating and sorting in increasing order the B ($B < n!$) values of the F-statistic, H_0 is rejected if $F_{obs} \geq F_{(k)}$ where $F_{(k)}$ is the quantile that corresponds to $k = \lfloor B(1-\alpha) \rfloor$, the integer part of $B(1-\alpha)$.

But in addition, we see that for the global test (F-test), according to the equation (4.7) the permutation, under the null hypothesis, can be assessed either on the response Y or on the residuals ϵ.

But to carry out partial tests, the permutation approach must be specify. This is the object of the next paragraph.

4.2.3 Manly's method: permutation on the responses

The following instances corresponds to Manly's methods:

```
IWTlm(formula,B = 1000, method = 'responses')
IWTlm_FoF(formula,B = 1000, method = 'responses')
IWTlmaov(formula,B = 1000, method = 'responses')
```

We suppose here , for sake of simplicity, to study the linear model (4.6), with the semi-parametric assumptions, here it is:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$$

We suppose also to want to assess the partial test :

$$H_0 : \beta_2 = 0 \quad vs \quad H_1 : \beta_2 = 0 \tag{4.10}$$

that corresponds to the reduced model:

$$Y = \beta_0 + \beta_1 X_1 + \epsilon \tag{4.11}$$

The following five steps, give the details of the Manly permuting method:

1. The Variable Y is regressed on X_1 and X_2 together (using least squares) to obtain an estimate $\hat{\beta_2}$ of β_2 and a value of the usual t-statistic, t_{ref} for testing $\beta_2 = 0$ for the real data. We hereafter refer to this as the reference value of t.
2. The Y values are permuted randomly to obtain permuted values Y^{π}.
3. The Y^{π} values are regressed on X_1 and X_2 (unpermuted) together to obtain an estimate $\hat{\beta_2}^{\pi}$ of β_2^{π} and a value of t^* for the permuted data.
4. Steps 2-3 are repeated a large number of times, yielding a distribution of values of t^{π} under permutation.
5. The absolute value of the reference value t_{ref} is placed in the distribution of absolute values of t^{π} obtained under permutation (for a two-tailed t-test). The probability is calculated as the proportion of values in this distribution greater than or equal, in absolute value, to the absolute value of t_{ref}.

4.2.4 Freedman and Lane: permutation on the residuals

The following instances corresponds to Freedman and Lane methods:

```
IWTlm(formula,B = 1000, method = 'residuals')
IWTlm_FoF(formula,B = 1000, method = 'residuals')
IWTlmaov(formula,B = 1000, method = 'residuals')
```

We start with the same assumptions as the previous part, on Manly's method.
We suppose here, for sake of simplicity, to study the linear model (4.6), with the semi-parametric assumptions, here it is:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$$

We suppose,one more time, to want to assess the partial test :

$$H_0 : \beta_2 = 0 \quad vs \quad H_1 : \beta_2 = 0 \tag{4.12}$$

that corresponds to the reduced model:

$$Y = \beta_0 + \beta_1 X_1 + \epsilon \tag{4.13}$$

But we see that here, under the null hypothesis, the data Y are no longer exchangeable, due to the presence of the other covariates X_1. The errors ϵ of the reduced regression model (4.11) are the only exchangeable quantities. The main problem is that the ϵ_i are theoretical quantities, and so are unknown.

A common solution is based on exchanging the residuals $\hat{e}_i$ estimated from the model (4.11) instead of the errors ϵ_i. This solution leads to an asymptotically correct permutation test, as the residuals are asymptotically exchangeable quantities.
Here is the following steps.

1. Estimate the full model (4.6)

2. Calculate the t-tests statistics for the effect of the covariate considered in the partial test, i.e $\hat{\beta}_2$, its value is denoted by t_0

3. Estimate the reduced model (4.11), and save the residuals $\hat{e}_i$

4. We repeat the following B times:

   - Permute the residuals of the reduced model (4.11)
   - Calculate the permuted responses, by adding the permuted residuals $\hat{e}_i^\pi$ to the fitted value of the model (4.11) as follows:

     $$Y^\pi = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{e}_i^\pi \tag{4.14}$$

     where the $\beta_{k,\pi}$ correspond to the OLS estimation obtain in the step 3.
   - Estimate the full model (4.6) with the permuted responses Y^π.
   - And then, evaluate the t-test statistic for the effect, $\hat{\beta}_2$, of the covariate of interest X_2, its value is denoted by t^π.

5. Evaluation of the p-value of the test as the proportion of permuted scenarios in which the test statistic t^π exceeds t_0: Let $\mathcal{B}$ a subset of B permutations.

   $$p - value = \frac{Card\{t^\pi \geq t_0 \mid \pi \in \mathcal{B}\}}{B} \tag{4.15}$$

4.3 Linear Models

A General Linear Model, can be written as follows:

Let the responses are denoted by the variable Y and the observations $(y_{i,j})_{i,j}, i = 1, .., n$ and $j = 1, .., J$. With n the number of individuals in the experience, and J the number of classes or groups in the experiment design. Then, all the data obtained from Y variable, are pooled in the same pooled vector denoted from now Y, s.t :

$$Y := \begin{bmatrix} y_{1,1} \\ y_{2,1} \\ \vdots \\ y_{n,1} \\ y_{1,2} \\ \vdots \\ y_{n,J} \end{bmatrix} \in \mathbb{R}^{(n*J)\times 1} \tag{4.16}$$

We decide, for example, to explain Y with p explanatory variables $X_k,\ k = 1,..,p$.
In indices-form the model is expressed as follow:

$$Y_i = \beta_0 + \beta_1 X_{i,1} + ... + \beta_p X_{i,p} + \epsilon_i \tag{4.17}$$

In matrix-form:

$$Y = Z.\beta + \epsilon \tag{4.18}$$

Where :

$$Z = \begin{bmatrix} 1 & x_{1,1} & \dots & x_{1,p} \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n,1} & \dots & x_{n,p} \end{bmatrix} \in \mathbb{R}^{n\times(p+1)} \tag{4.19}$$

The matrix Z, is called design matrix of the model.

In R, it is obtained as output of `model.matrix(formula)`. The functions working on regression model (`IWTlm.R`, `IWTlm_FoF.R` and `IWTaov.R`) all compute, beforehand, the design matrix of the model, to handle more easily the different computations (model fitting) linked to the permutations tests.

In ANOVA framework, covariates are factors of different levels, we will use the so called dummy variables, and end up with a special design matrix. The details will be given in the corresponding section ANOVA.

4.3.1 Linear Model: Functional-on-Scalar

We start by defining the model:

$$Y = Z_{scal}.\beta + \epsilon \tag{4.20}$$

In this case, Z_{scal} refer to a design matrix made with scalar covariates, i.e n-dimensional vectors.

And the response Y is a functional variable, and therefore can be either represented by a $n \times p$ matrix (n: individuals , p: times scale discretization) or an `fd` object.

Then from this model, IWTP and permutation tests, described above, can be applied for the intervals selection of the functional domain.

Application:
We launch the following instance of the `IWTlm`

```
 IWTlm(formula= data ~ lab1 + lab2,
                 B = 1000,
                 method = 'residuals',
                 dx=NULL,
                 recycle=TRUE,
                 statistic='std',
                 IW_scale= 100)
```

where data is functional matrix or an `fd` object, gathering the temperature every day, during 22 years, in Milan and Paris. `lab1` and `lab2` are two scalar variables created as follows:

```
lab <- c(rep(0,22),rep(1,22))
lab1 <- lab
lab2 <- factor(rep(1:3,15)[-45])
```

The variable `lab2` is interpreted as factor, with three levels (1, 2 and 3). The three factor levels appear as dummy variables in the design matrix. To bring precision on the levels, as it happens with the first level 1, is taken as a reference level and doesn't appear in the design matrix.

Indeed, this to keep the design Z matrix being of rank p (regularity condition to allow $Z^t Z$ being invertible).That's why the t-test are performed on the levels 2 and 3 of the textttlab2 (`lab22`, `lab23` in the graphics).

This important role plays by dummy variables will be specified in more details in the section ANOVA.

The results and their interpretation can be visualized with the plots:

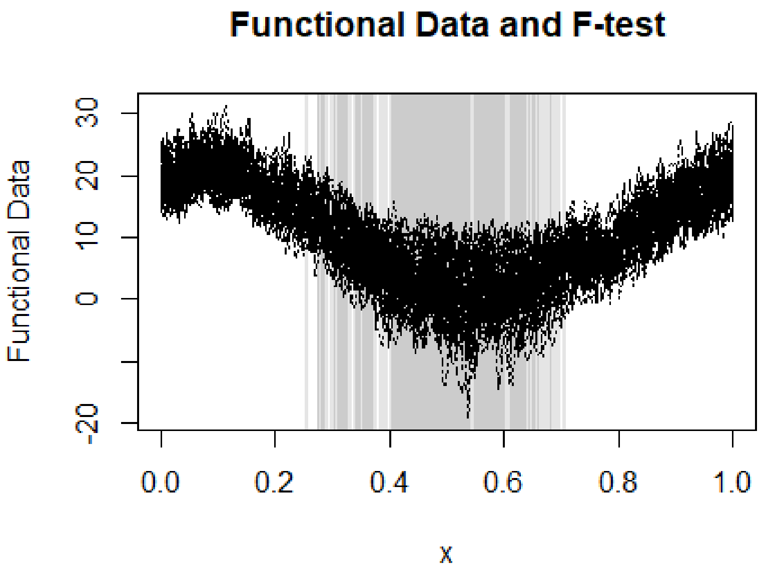


Figure 4.3: Functional data plot and the global F-test, rejection of H_0 in gray (IWTlm).

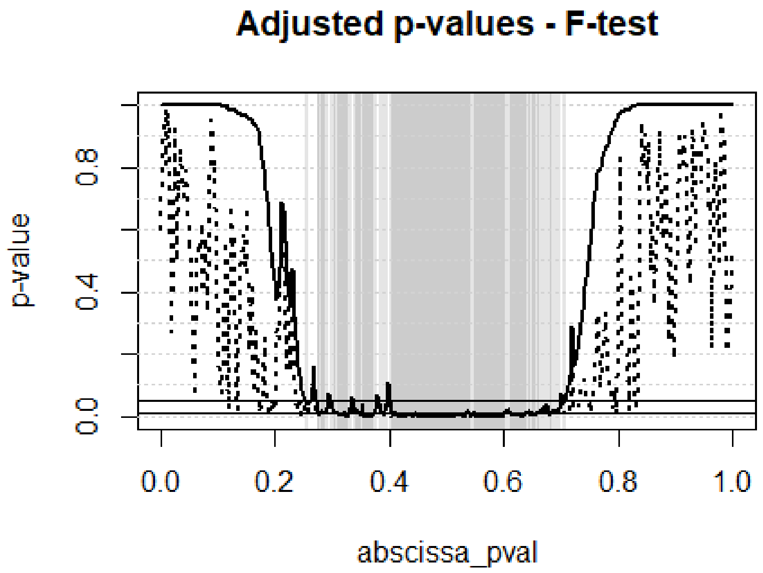


Figure 4.4: Adjusted and unadjusted (dotted) p-values for the global F-test, rejection of H_0 in gray (IWTlm).

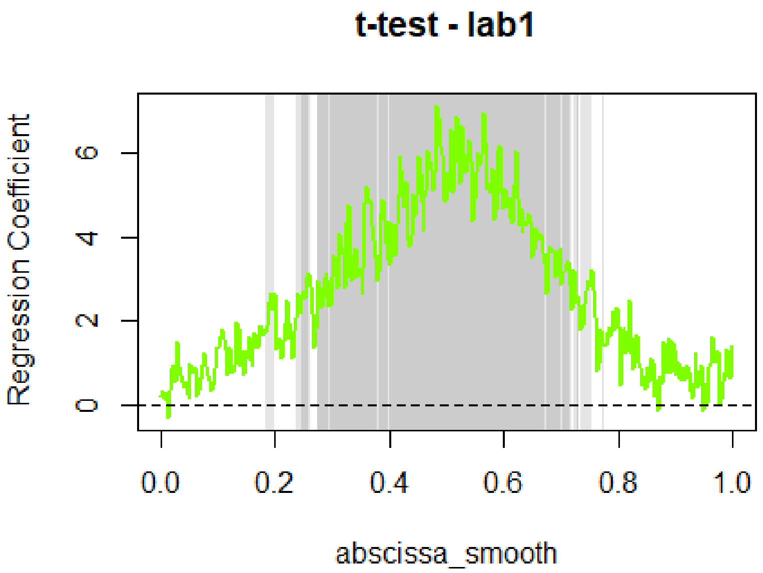


Figure 4.5: Partial t-test on `lab1`, rejection of H_0 in gray (IWTlm).

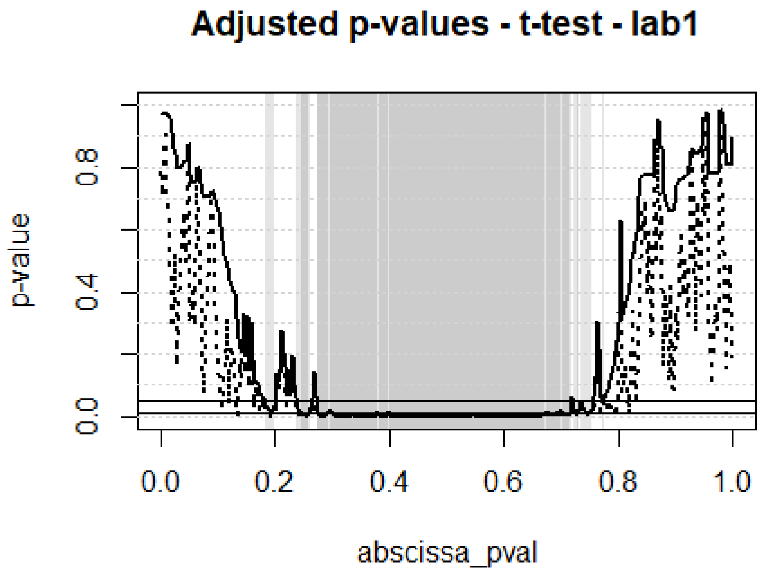


Figure 4.6: Adjusted and unadjusted (dotted) p-values for the partial t-test of `lab1`, rejection of H_0 in gray (IWTlm).

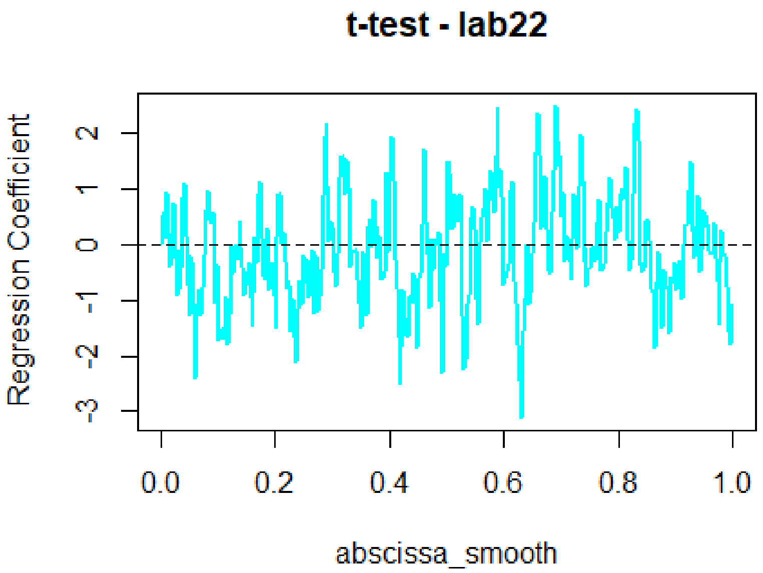


Figure 4.7: Partial t-test on `lab22` factor level, rejection of H_0 in gray (no rejection here) (IWTlm).

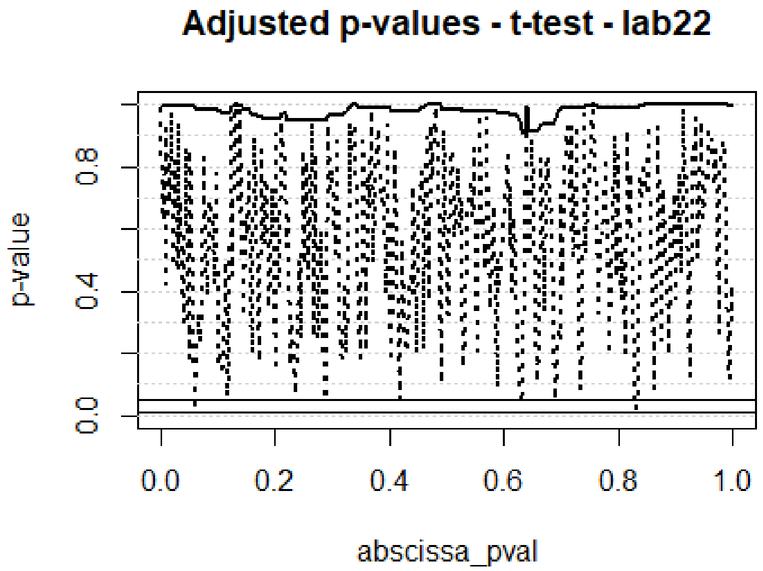


Figure 4.8: Adjusted and unadjusted (dotted) p-values for the partial t-test of `lab22` factor level, rejection of H_0 in gray (no rejection here) (IWTlm).

4.3.2 Linear Model: Functional-on-Functional

As in the previous, section we start by defining the model:

$$Y = Z_{func}.\beta + \epsilon \tag{4.21}$$

In this case, Z_{func} refers to a design matrix of functional variables (and that can be mixed with scalar), can be either represented by a $n \times p$ matrix or `fd` object if functional, or n-dimensional vector if scalar.

And the response Y is a functional variable, and therefore can be either represented by a $n \times p$ matrix (n: individuals, p: times discretization points) or an `fd` object.

Then from this model IWTP and permutation tests, described above, can be applied for the intervals selection of the functional domain.

Application:
We launch the following instance of the `IWTlm_FoF`

```
IWTlm_FoF(formula= data1 ~ years + data2, IW_scale= 100)
```

where the data are represented as follows:

```
years <- 1:22
data1 <- NASAtemp\$milan
data2 <- NASAtemp\$paris
```

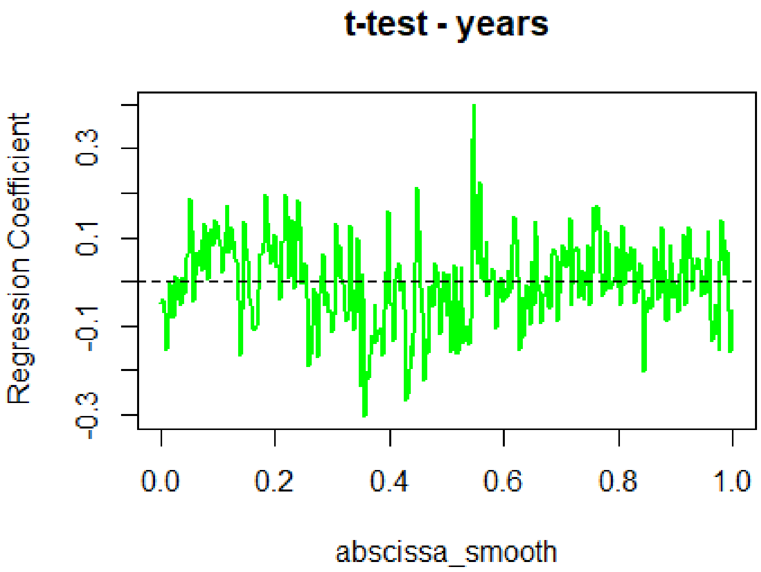


Figure 4.9: Functional data plot and the global F-test, rejection of H_0 in gray (IWTlm_FoF).

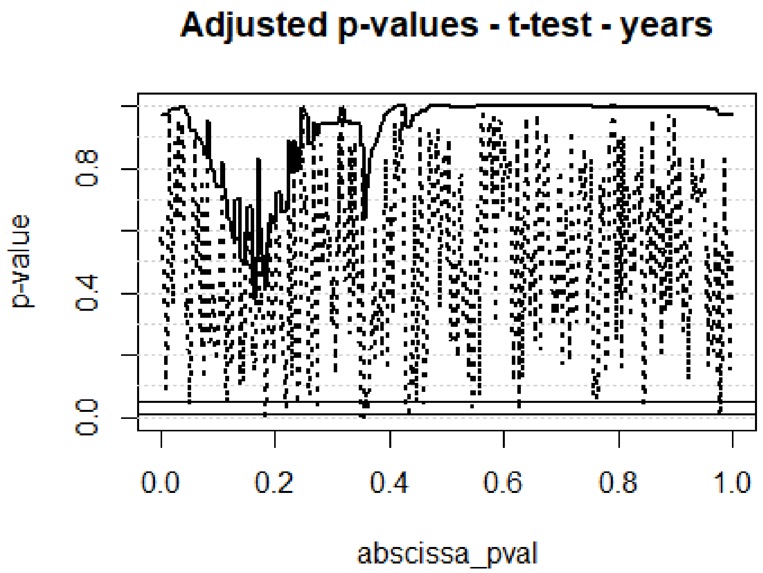


Figure 4.10: Adjusted and unadjusted (dotted) p-values for the global F-test, rejection of H_0 in gray (IWTlm_FoF).

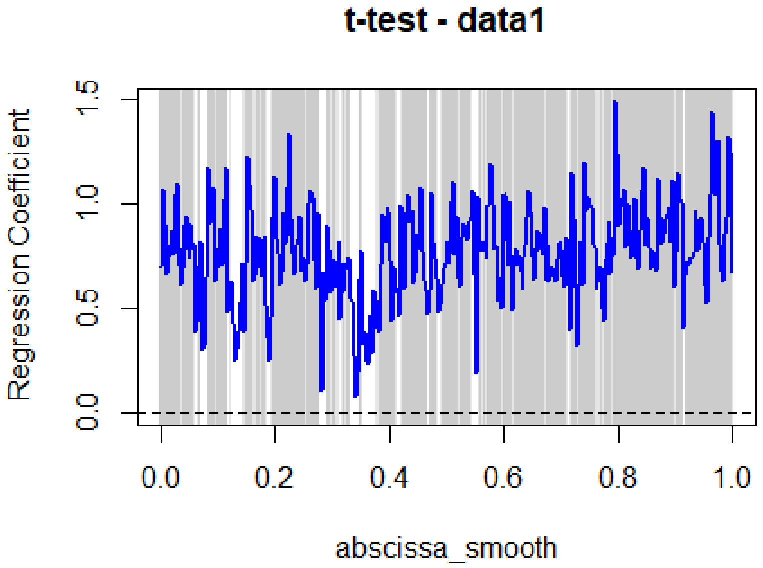


Figure 4.11: Partial t-test on `data1`, rejection of H_0 in gray (IWTlm_FoF).

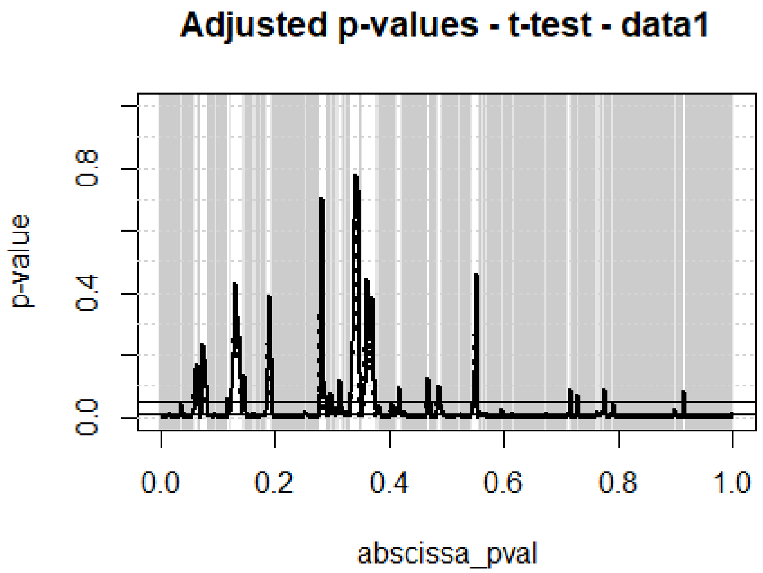


Figure 4.12: Adjusted and unadjusted (dotted) p-values for the partial t-test of `data1`, rejection of H_0 in gray (IWTlm_FoF).

We add the following example: the test of `data1` $\sim$ `years`, to highlight the effects of the variables `years`, and be sure that it has not been masked by the other covariate `data2`: `IWTlm_FoF(formula= data1` $\sim$ `years, IW_scale= 365)`

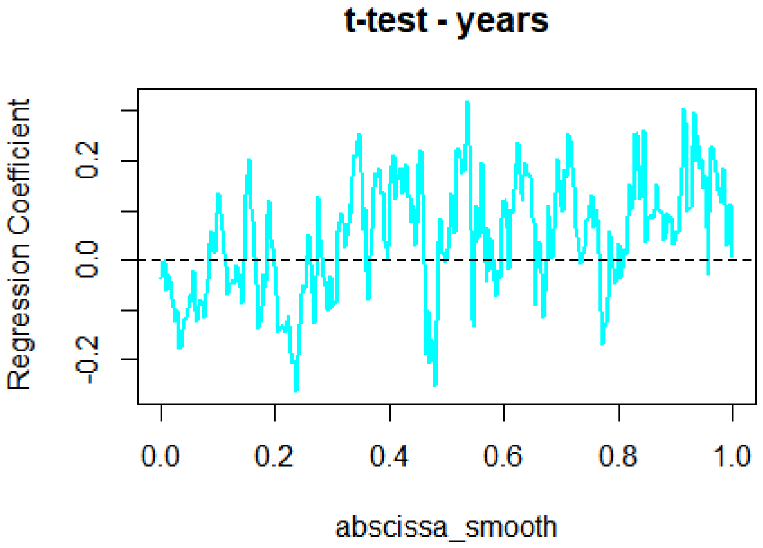


Figure 4.13: Partial t-test of `years`, rejection of H_0 in gray (IWTlm_FoF).

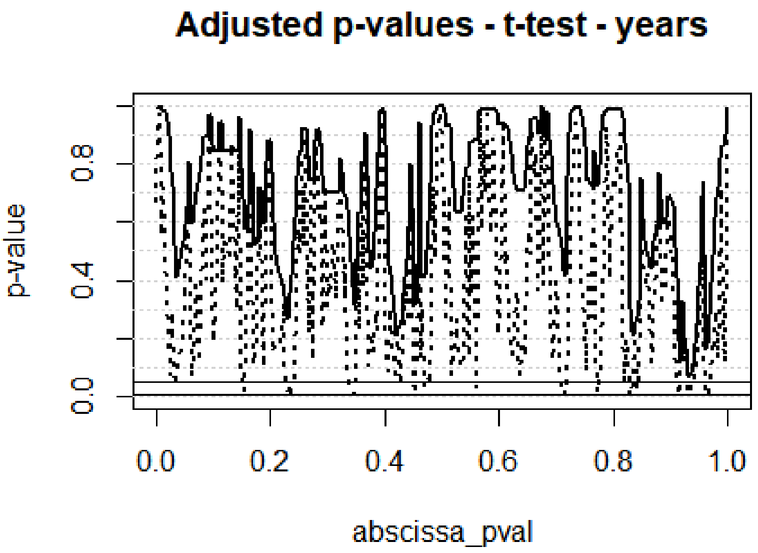


Figure 4.14: Adjusted and unadjusted (dotted) p-values for the partial t-test of `years`, rejection of H_0 in gray (IWTlm_FoF).

As we see on the last graph, there is no rejection zone (in gray) plotted. So the hypothesis $H_0 : \beta_{years} = 0$, is interval-wise accepted on the functional domain.

4.4 ANOVA

The implemented function `IWaov.R` implements the IWTP in the framework of an ANOVA regression, which is named MANOVA when the analysis of variance deals with models where the dependent variables are multivariate. In our study, we are also very interested in the case where the data are functional (`fd` object in the package fda), we talk about FANOVA.

The formal model of a FANOVA is:

$$Y_{i,j}(t) = \mu(t) + \alpha_j(t) + \epsilon_{i,j}(t) \tag{4.22}$$

where $Y_{i,j}(t)$ is the responses data corresponding to the individuals i in group j at time t, $\mu(t)$ is the intercept at time t and corresponds to the great mean (over all individuals and groups), $\alpha_j(t)$ to the mean (the effect over Y) of the group j at time t, and finally the unexplained variability is gathered into the errors $\epsilon_{i,j}$.

For sake of simplicity we will take the example of 3 levels for the factors α_j (i.e $j = 1, ..., 3$), so denoted at time t by : $\alpha_1(t)$, $\alpha_2(t)$, $\alpha_3(t)$. And considered $n = 6$ individuals, with two individuals by groups, i.e in the following pooled vector:

$$Y := \begin{bmatrix} y_{1,1} \\ y_{2,1} \\ y_{3,2} \\ y_{4,2} \\ y_{5,3} \\ y_{6,3} \end{bmatrix} \in \mathbb{R}^{6\times 1} \tag{4.23}$$

We recall the usual regression design matrix-form equation:

$$Y = Z.\beta + \epsilon \tag{4.24}$$

In our case we can write it as follows:

$$Y = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \mu(t) \\ \alpha_1(t) \\ \alpha_2(t) \\ \alpha_3(t) \end{bmatrix} + \epsilon \tag{4.25}$$

In this design the covariates are encoded for each levels with 0 and 1, they are called dummmy variables. And so we see that this model can be put in standard matrix form (equation 4.22), with:

$$\beta_0(t) = \mu(t),\ \beta_1(t) = \alpha_1(t),\ \beta_2(t) = \alpha_2(t),\ \beta_3 = \alpha_3(t) \tag{4.26}$$

But a problem can arise from that design form:

In that case the design matrix has the property that the sum of columns 2,3,4 will equal the first column for the intercept. So therefore the linear model is no more regular (i.e with Z no more injective).
As consequence, the coefficients $\beta(t)$ cannot be identified uniquely when performing the least square minimization:

$$\underset{\beta}{\operatorname{argmin}} \ \mathbf{LMSSE}(\beta) = ||Y - Z.\beta||^2 \tag{4.27}$$

The solution is to perform such a minimization under a colinearity constraint, as follows:

$$\underset{\beta}{\operatorname{argmin}} \ \mathbf{LMSSE}(\beta) = ||Y - Z.\beta||^2 \quad \mathbf{s.t} \quad \sum_{j=1}^{3} \beta_j = 0 \tag{4.28}$$

Note that the constraint is equivalent to $\sum\limits_{j=1}^{3} \beta_j = 0$.

In practice, the design returned is regular, i.e one factor level is omitted from the coding. It does not matter which one we eliminate, although some refer, in certain cases, to the eliminated level as a ‘reference’ level. Then from this model IWTP and permutation tests, described above, can be applied for the interval selection of the functional domain.

Application:
We launch the following instance of the `IWTaov`:
`IWTaov.result<-IWTaov(formula= data~lab1 + lab2, IW_scale=365)`

where data is functional matrix or an `fd` object, gathering the temperature every day, during 22 years, in Milan and Paris. `lab1` and `lab2` are two factors of respectively two and three levels created as follows:

```
lab <- c(rep(0,22),rep(1,22))
lab1 <- lab
lab2 <- factor(rep(1:3,15)[-45])
```

The results and their interpretation can be visualized with the plots obtained with the `IWTimage.R` function:

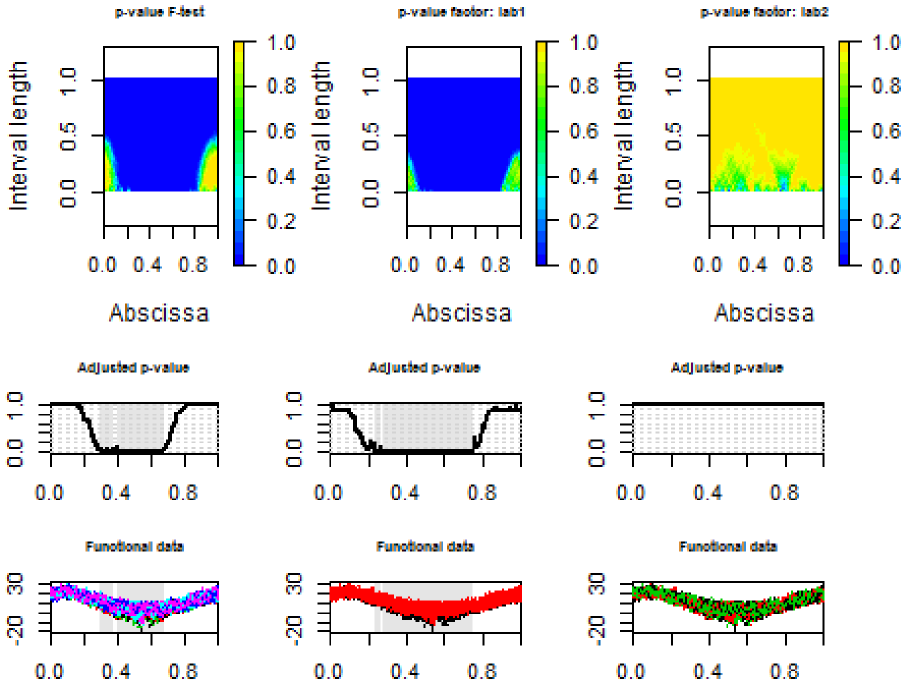


Figure 4.15: Functional data, adjusted and heatmap p-values, rejection of H_0 in gray (IWTaov).

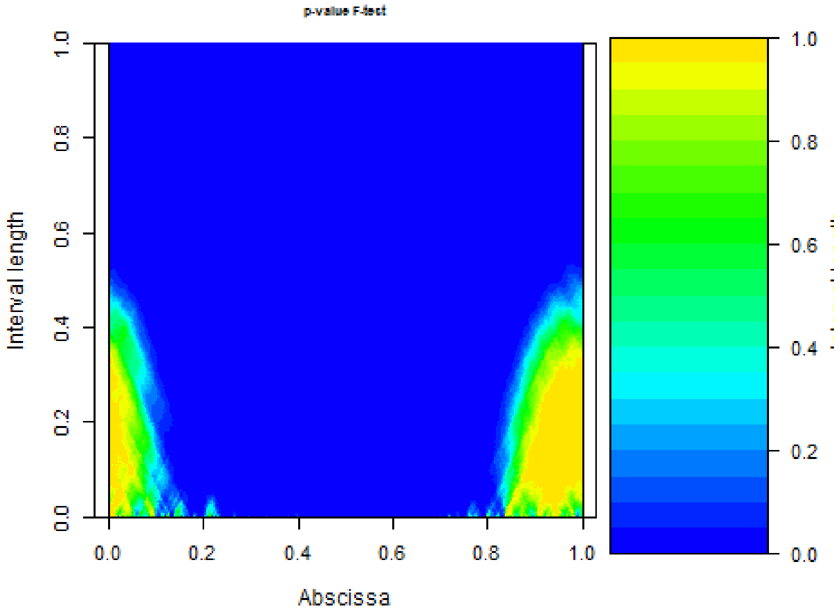


Figure 4.16: Zoom on the heatmap of p-values for the global test, p-value heat-level indicated on the right scale (IWTaov).

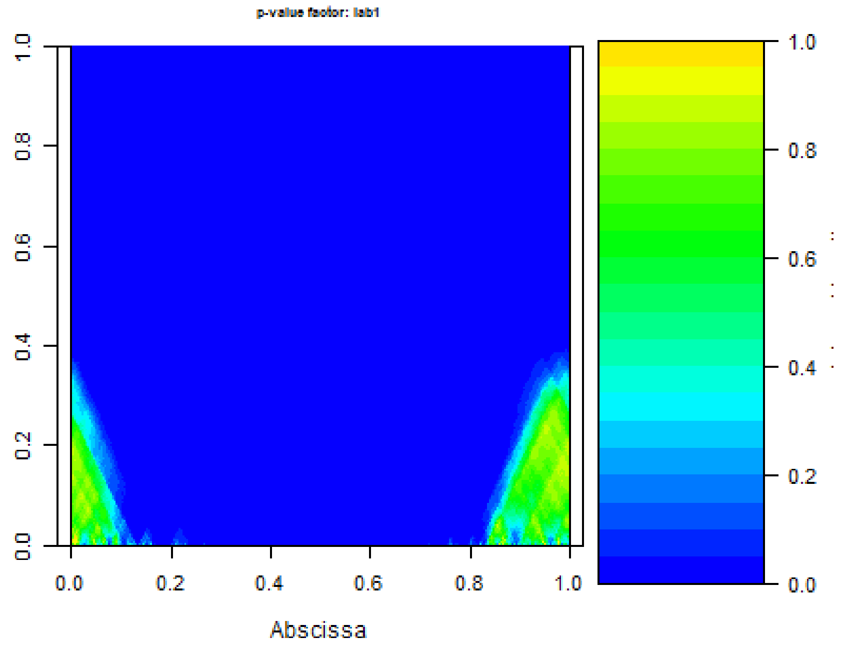


Figure 4.17: Zoom on the heatmap of p-values the partial tests on the 1^{st} factor, p-value heat-level indicated on the right scale (IWTaov).

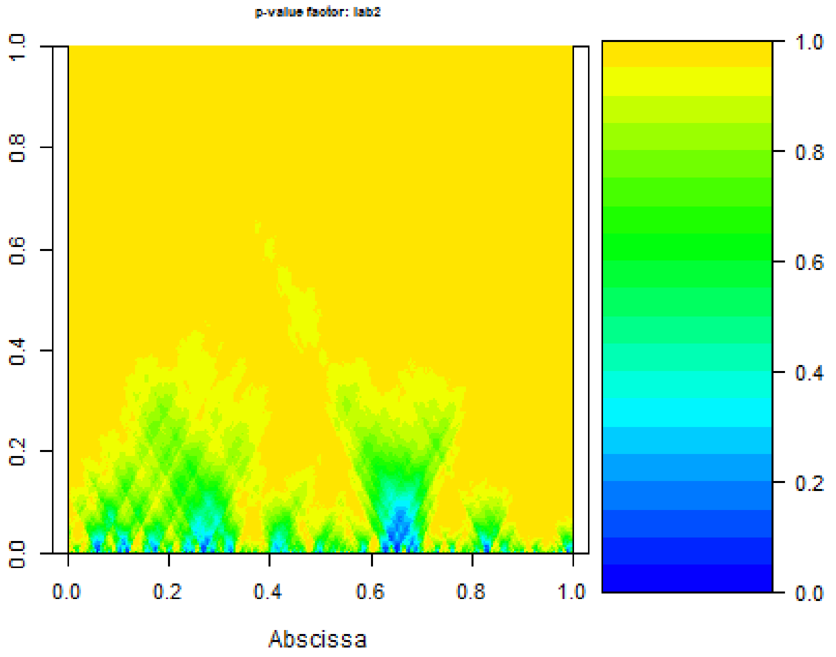


Figure 4.18: Zoom on the heatmap of p-values the partial tests on the 2^{nd} factor, p-value heat-level indicated on the right scale (IWTaov).

The function produces also the following design matrix obtained with:
`IWTaov.result$design_matrix`

Table 4.1: ANOVA design_matrix

	(Intercept)	lab1	lab22	lab23
1	1	0	0	0
2	1	0	1	0
3	1	0	0	1
4	1	0	0	0
5	1	0	1	0
6	1	0	0	1
7	1	0	0	0
8	1	0	1	0
9	1	0	0	1
10	1	0	0	0
11	1	0	1	0
12	1	0	0	1
13	1	0	0	0
14	1	0	1	0
15	1	0	0	1
16	1	0	0	0
17	1	0	1	0
18	1	0	0	1
19	1	0	0	0
20	1	0	1	0
21	1	0	0	1
22	1	0	0	0
23	1	1	1	0
24	1	1	0	1
⋮	⋮	⋮	⋮	⋮
44	1	1	1	0

So as we said before, it is interesting to remark that the second column corresponds to the first covariate `lab1` defined as a standard scalar variables, and that the two following columns correspond to the 2 and 3 level of the variable `lab2`, encoded as dummy variables (with 0 and 1).

The level 1 does not appear to preserve the regularity condition of the model, and can be interpreted as a 'reference' variable.

Package ‘IWTP’

April 12, 2018

Type Package

Title Interval-Wise testing procedure for functional data

Version 0.1.0

Date 2017-06-21

Depends fda

Author A.Pini, S.Vantini

Maintainer Alessia Pini <alessia.pini@polimi.it>

Description Implementation of the local inferential procedure
for functional data in different frameworks
(i.e., one or two-population frameworks, functional linear models).
The current version of the package requires functional data evaluated
on a uniform grid or fda objects and provides the adjusted and unadjusted
p-value functions based on nonparametric local tests.
The package provides a plotting function creating a graphical
output of the procedure: the p-value heat-map, the plot
of the p-value functions, and the plot of the functional data.

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

R topics documented:

IWT1 *One population Interval-Wise Testing Procedure*

Description

The function implements the Interval-Wise Testing Procedure for testing the center of symmetry of a functional population evaluated on a uniform grid. Data are functional represented by means of matrix or object of class fd (package "fda") and the significance of each evaluation point of the functional population with respect to the center of symmetry, is tested with a control of the interval-wise error rate.

Usage

```
IWT1(data, mu = 0, B = 1000, dx = NULL, recycle = TRUE, IW_scale = 2)
```

Arguments

`data` Pointwise evaluations of the functional data set on a uniform grid, in the form of matrices (1) or functional data objects (2). (1). Matrix of dimension `c(n,p)` containing the evaluations of functional data on the grid, where `n` is the sample sizes and `p` is the number of grid points. (2).Object of class fd (package "fda")

`mu` The center of symmetry under the null hypothesis: either a constant (in this case, a constant function is used) or a p-dimensional vector containing the evaluations on the same grid where data are evaluated. The default is `mu=0`.

`B` The number of iterations of the MC algorithm to evaluate the p-values of the permutation tests. The default is `B=1000`.

`dx` Uniform discretization grid step size, for objects of class fd. The default is `dx=NULL`.

`recycle` Flag indicating whether a recycled version of IWT has to be performed. The default is `recycle=TRUE`.

`IW_scale` Scale parameter that determines the depth of the interval-wise testing procedure: `maxrows`. `maxrows= p-IW_scale+1` defines the maximum size of the permutation intervals use in the testing procedure.

Value

`IWT1` returns an object of class `"IWT1"`. An object of class `"IWT1"` is a list containing at least the following components:

`test` String vector indicating the type of test performed. In this case equal to `"1pop"`.

`mu` Center of symmetry under the null hypothesis (as entered by the user).

`adjusted_pval` Vector of adjusted p-values evaluated on the same grid as functional data.

`unadjusted_pval` Vector of unadjusted p-values evaluated on the same grid as functional data.

`pval_matrix` Matrix of dimensions `c(p,p)` of the p-values of the multivariate tests. The element `(i,j)` of matrix `pval_matrix`, contains the p-value of the joint NPC test of the evaluation points `(j,j+1,...,j+(p-i))`.

`data.eval` Evaluation of functional data.

Author(s)

Alessia Pini, Simone Vantini

References

A. Pini and S. Vantini (2016). The Interval Testing Procedure: A General Framework for Inference for Functional Data Analysis. MOX-report 2016, Politecnico di Milano

A. Pini and S. Vantini (2017). Interval-wise Testing for Functional Data, Journal of Non-parametric Statistics,DOI:10.1080/10485252.2017.1306627

See Also

See also ITW2, IWTimage

Examples

```
# Importing the NASA temperatures data set
data(NASAtemp)

# Data as 'fd' object
library("fda")
m <- 4
p <- dim(data1)[2]
abscissa <- 1:p
bspl_basis <- create.bspline.basis(range(abscissa), norder = m, nbasis = p)
functional_temp <- Data2fd(argvals = abscissa, y = t(NASAtemp$milan), bspl_basis)


# Performing the IWTP for one populations
IWT.result <-IWT1(NASAtemp$paris, mu=4, recycle=TRUE, IW_scale=30)

IWT.result <-IWT1(functional_temp, mu=2, recycle=FALSE, IW_scale=10)

# Plotting the results of the IWT
plot(IWT.result, main='Milan temperatures')

# Plotting the p-value heatmap
IWTimage(IWT.result)

# Selecting the significant points for the radius at 5% level
which(IWT.result$adjusted_pval < 0.05)
```

IWT2 *Two population Interval-Wise Testing Procedure*

Description

The function implements the Interval-Wise Testing Procedure for testing the difference between two functional populations evaluated on a uniform grid. Data are either functional represented by means of matrix or object of class fd (package "fda") and the significance of each evaluation point of each functional population with respect to to the other population to be compared, is tested with a control of the interval-wise error rate.

Usage

```
IWT2(data1,data2,mu=0,B=1000,paired=FALSE,dx=NULL,recycle=TRUE,
     alternative="two.sided",IW_scale=2)
```

Arguments

data1 Pointwise evaluations of the first population's functional data set on a uniform grid, in the form of matrices (1) or functional data objects (2). (1). Matrix of dimension c(n1,p) containing the evaluations of functional data on the grid, where n1 is the sample sizes and p is the number of grid points. (2). Object of class fd (package "fda")

data2 Pointwise evaluations of the second population's functional data set on a uniform grid, in the form of matrices (1) or functional data objects (2). (1). Matrice of dimension c(n2,p) containing the evaluations of functional data on the grid, where n2 is the sample sizes and p is the number of grid points. (2).Object of class fd (package "fda")

mu The difference between the first functional population and the second functional population under the null hypothesis. Either a constant (in this case, a constant function is used) or a p-dimensional vector containing the evaluations on the same grid which data are evaluated.The default is mu=0.

B The number of iterations of the MC algorithm to evaluate the p-values of the permutation tests. The default is B=1000.

dx Uniform discretization grid step size, for functional data of class fd. The default is dx=NULL.

recycle Flag indicating whether a recycled version of IWT has to be performed. The default is recycle=TRUE.

alternative Set the type of the test alternative hypothese. Possible values are "two.sided", "less", "greater". The default is alternative="two.sided".

IW_scale Scale parameter that determines the depth of the interval-wise testing procedure: maxrows. maxrows= p-IW_scale+1 defines the maximum size of the permutation intervals use in the testing procedure.

Value

IWT2 returns an object of class "IWT2". An object of class "IWT2" is a list containing at least the following components:

test String vector indicating the type of test performed. In this case equal to "2pop".

mu Center of symmetry under the null hypothesis (as entered by the user).

adjusted_pval Adjusted p-values for each evaluation point.

unadjusted_pval Unadjusted p-values for each evaluation point.

pval_matrix Matrix of dimensions c(p,p) of the p-values of the multivariate tests. The element (i,j) of matrix pval_matrix, contains the p-value of the joint NPC test of the evaluation points (j,j+1,...,j+(p-i)).

data_eval Matrix of the centered data evaluated on the uniform grid.

ord_labels Vector of labels indicating the group membership of data_eval.

Author(s)

Alessia Pini, Simone Vantini

References

A. Pini and S. Vantini (2016). The Interval Testing Procedure: A General Framework for Inference for Functional Data Analysis. MOX-report 2016, Politecnico di Milano

A. Pini and S. Vantini (2017). Interval-wise Testing for Functional Data, Journal of Non-parametric Statistics,DOI:10.1080/10485252.2017.1306627

See Also

See also `ITW1`, `IWTimage`

Examples

```
# Importing the NASA temperatures data set
data(NASAtemp)

# Data as 'fd' object
library("fda")
m <- 4
p <- dim(data1)[2]
abscissa <- 1:p
bspl_basis <- create.bspline.basis(range(abscissa), norder = m, nbasis = p)
functional_temp1 <- Data2fd(argvals = abscissa, y = t(NASAtemp$paris), bspl_basis)
functional_temp2 <- Data2fd(argvals = abscissa, y = t(NASAtemp$geneva), bspl_basis)

# Performing the IWTP for two-populations
IWT.result <-IWT2(NASAtemp$paris,NASAtemp$geneva,mu=0,IW_scale=10)

IWT.result <-IWT2(functional_temp1, functional_temp2, mu=0, IW_scale=10)

# Plotting the results of the IWT
plot(IWT.result, main='Paris vs Geneva temperatures')

# Plotting the p-value heatmap
IWTimage(IWT.result, plot_unadjusted=TRUE)

# Selecting the significant points for the radius at 5% level
which(IWT.result$adjusted_pval < 0.05)
```

`IWTaov` *Interval-Wise Testing Procedure for testing Functional analysis of variance*

Description

IWTaov is used to fit and test functional analysis of variance. The function implements the Interval-Wise Testing Procedure for testing for significant differences between several functional population evaluated on a uniform grid. Data are represented by means of class fd object or embedded in a matrix of dimensions c(n,p). The significance of each functional evaluation point is tested with a control of the interval-wise error rate.

Usage

```
IWTaov(formula,B=1000,method='residuals',dx=NULL,recycle=TRUE,IW_scale=2)
```

Arguments

formula	An object of class "formula" specifying the model to fit. Example: y ~ A + B where y is a matrix of dimension c(n,p) containing the point-wise evaluations of the n functional data on p points or an object of class fd (package "fda") containing the functional data set. A, B are n-dimensional factors.
B	The number of iterations of the MC algorithm to evaluate the p-values of the permutation tests. The default is B=1000.
method	Permutation method used to calculate the p-value of permutation tests. Choose "residuals" for the permutations of residuals under the reduced model, according to the Freedman and Lane scheme, and "responses" for the permutation of the responses, according to the Manly scheme.
dx	Uniform discretization grid step size, for objects of class fd. The default is dx=NULL.
recycle	Flag specifying whether the recycled version of IWT has to be used. The default is recycle=TRUE.
IW_scale	Scale parameter that determines the depth of the interval-wise testing procedure: maxrows. maxrows= p-IW_scale+1 defines the maximum size of the permutation intervals use in the testing procedure.

Value

IWTaov returns an object of class "IWTaov".

The function summary is used to obtain and print a summary of the results.

An object of class "IWTaov" is a list containing at least the following components:

call	The matched call.
design_matrix	The design matrix of the linear model.
unadjusted_pval_F	Unadjusted p-values of the F-test.
unadjusted_pval_factors	Unadjusted p-values of the functional t-tests on each factor, separately (rows) on each domain point (columns).
pval_matrix_F	Matrix of dimensions c(p,p) of the p-values of the multivariate F-tests. The element (i,j) of matrix pval_matrix, contains the p-value of the joint NPC test of the evaluation points (j,j+1,...,j+(p-i)).

`pval_matrix_factors`
: Array of dimensions `c(nvar+1,p,p)` of the p-values of the multivariate t-tests. Where `nvar+1` corresponds to the number of factors `nvar` plus the intercept. The element `(n,i,j)` of array `pval_matrix` contains the p-value of the joint NPC test on factor number `n` of the evaluation points `(j,j+1,...,j+(p-i))`.

`adjusted_pval_F`
: Adjusted p-values of the functional F-test for each evaluation point.

`adjusted_pval_factors`
: Adjusted p-values of the functional t-tests on each factor (rows) on each domain point (columns).

`data.eval` Evaluation of functional data.

`coeff.regr.eval`
: Evaluation of the regression coefficients.

`fitted.eval` Evaluation of the fitted values.

`residuals.eval` Evaluation of the residuals.

`R2.eval` Evaluation of the functional R-squared

Author(s)

Alessia Pini, Simone Vantini

References

D. Freedman and D. Lane (1983). A Nonstochastic Interpretation of Reported Significance Levels. *Journal of Business & Economic Statistics* 1.4, 292-298.

B. F. J. Manly (2006). Randomization, *Bootstrap and Monte Carlo Methods in Biology*. Vol. 70. CRC Press.

A. Pini and S. Vantini (2016). The Interval Testing Procedure: A General Framework for Inference for Functional Data Analysis. MOX-report 2016, Politecnico di Milano

A. Pini and S. Vantini (2017). Interval-wise Testing for Functional Data, Journal of Non-parametric Statistics,DOI:10.1080/10485252.2017.1306627

See Also

See `summary.IWTaov` for summaries and `plot.IWTaov` for plotting the results.

See also `IWTlm` to fit and test a functional-on-scalar linear model applying the IWT, and `IWT1`, `IWT2`, for one-population and two-populations tests.

Examples

```
# Importing the NASA temperatures data set
data(NASAtemp)

# Data definition as matrix
temperature <- rbind(NASAtemp$milan,NASAtemp$paris)

# Data definition as 'fd' object
functional_temp <- Data2fd(argvals = abscissa, y = t(temperature), bspl_basis)

# Defining factors
groups <- c(rep(0,22),rep(1,22))
```

```
group1 <- groups
group2 <- factor( rep(1:3,15)[-45] )

# Performing the IWT
IWT.result <- IWTaov(functional_temp ~ groups, B=1000,method='residuals',
              dx=NULL,recycle=TRUE,IW_scale=10)

IWT.result <- IWTaov(temperature ~ group1 + group2, B=1000,method='responses',
              dx=NULL,recycle=TRUE,IW_scale=10)

IWT.result <- IWTaov(temperature ~ group1 * group2, B=1000,method='residuals',
              dx=NULL,recycle=TRUE,IW_scale=10)

# Summary of the IWT results
summary(IWT.result)

# Plot of the IWT results
plot(IWT.result, plot_adjpval = TRUE, plot_unadjpval = TRUE)
```

IWTimage *Plot of the Interval-Wise Testing Procedure results*

Description

Plotting function creating a graphical output of the IWTP: the p-value heat-map, the plot of the adjusted p-values, and the plot of the functional data.

Usage

```
IWTimage(IWT.result, alpha = 0.05, abscissa.range = c(0, 1), nlevel = 20, plot_unadjusted=FALSE)
```

Arguments

`IWT.result` Results of the IWTP, as created by `IWT1`, `IWT2`, `IWTaov`.

`alpha` Level of the hypothesis test. The default is `alpha`=0.05.

`abscissa.range` Range of the plot abscissa. The default is `c(0,1)`.

`nlevel` Number of desired color levels for the p-value heat-map. The default is `nlevel=20`.

`plot_unadjusted`
A logical indicating whether the plots of unadjusted p-values have to be done. Default is `plot_adjpval = FALSE`.

Value

No value returned. The function produces a graphical output of the IWT results: the p-value heatmap, a plot of the adjusted and unadjusted p-values and the plot of the functional data. The intervals selected as significant by the test at level `alpha` are highlighted in the plot of the adjsuted p-values by a gray area.

Author(s)

Alessia pini, Simone Vantini

References

A. Pini and S. Vantini (2016). The Interval Testing Procedure: A General Framework for Inference for Functional Data Analysis. MOX-report 2016, Politecnico di Milano

A. Pini and S. Vantini (2017). Interval-wise Testing for Functional Data, Journal of Non-parametric Statistics,DOI:10.1080/10485252.2017.1306627

See Also

See `plot.IWT1`, `plot.IWT2`, `plot.IWTlm`, and `plot.IWTaov` for the plot method applied to the IWT results of one- and two-populations tests, linear models, and ANOVA, respectively.

See also `IWT1`, `IWT2`, `IWTlm` and `IWTlm_FoF` for applying the IWTP.

Examples

```
# Importing the NASA temperatures data set
data(NASAtemp)

# Performing the IWT2 for two-populations
IWT.result <- IWT2(NASAtemp$milan,NASAtemp$paris, IW_scale=10)

# Plotting the results of the IWT
IWTimage(IWT.result)

# Selecting the significant components for the radius at 5% level
which(IWT.result$adjusted_pval < 0.05)
```

`IWTlm` *Interval-Wise Testing Procedure for testing Functional-on-Scalar Linear Models*

Description

`IWTlm` is used to fit and test functional-on-scalar linear models. It can be used to carry out regression, and analysis of variance. The function implements the Interval-Wise Testing Procedure for testing the significance of the effects of scalar covariates on a functional population evaluated on a uniform grid. Data are represented by means of `class fd` object or embedded in a matrix of dimensions `c(n,p)`. The significance of each functional evaluation point is tested with a control of the interval-wise error rate.

Usage

```
IWTlm(formula, B=1000, method='residuals', dx=NULL, recycle=TRUE, statistic='std', IW_scale=2)
```

Arguments

`formula` An object of class "`formula`" specifying the model to fit. Example: `y ~ x1 + x2` where `y` is a matrix of dimension `c(n,p)` containing the point-wise evaluations of the `n` functional data on `p` points or an object of class `fd` (see "fda" package) containing the functional data set. The scalar covariates `x1`, `x2` are `n`-dimensional vectors.

B — The number of iterations of the MC algorithm to evaluate the p-values of the permutation tests. The default is B=1000.

method — Permutation method used to calculate the p-value of permutation tests. Choose "residuals" for the permutations of residuals under the reduced model, according to the Freedman and Lane scheme, and "responses" for the permutation of the responses, according to the Manly scheme.

dx — Uniform discretization grid step size, for objects of class fd. The default is dx=NULL.

recycle — Flag specifying whether the recycled version of IWT has to be used.The default is recycle=TRUE.

statistic — Calculation mode of the statistic used for the covariates partial tests (t-test). The mention 'nstd' refers to non-standardized statistics, 'std' refers to standardized statistics, i.e divided by the standard-error.

IW_scale — Scale parameter that determines the depth of the interval-wise testing procedure: maxrows. maxrows= p-IW_scale+1 defines the maximum size of the permutation intervals use in the testing procedure. The default is IW_scale=2.

Value

IWTlm returns an object of class "IWTlm".

The function summary is used to obtain and print a summary of the results.

An object of class "IWTlm" is a list containing at least the following components:

call — The matched call.

design_matrix — The design matrix of the functional-on-functional linear model.

unadjusted_pval_F — Unadjusted p-values of the functional F-test for each evaluation point.

unadjusted_pval_part — Unadjusted p-values of the functional t-tests for each partial regression coefficients.

pval_matrix_F — Matrix of dimensions c(p,p) of the p-values of the multivariate F-tests. The element (i,j) of matrix pval_matrix contains the p-value of the joint NPC test of the evaluation points (j,j+1,...,j+(p-i)).

pval_matrix_part — Array of dimensions c(nvar+1,p,p) of the p-values of the multivariate t-tests. Where nvar+1 corresponds to the number nvar of covariates plus the intercept. The element (n,i,j) of array pval_matrix contains the p-value of the joint NPC test on covariate number n of the evaluation points (j,j+1,...,j+(p-i)).

adjusted_pval_F — Adjusted p-values of the functional F-test for each evaluation point.

adjusted_pval_part — Adjusted p-values of the functional t-tests for each partial regression coefficient.

data.eval — Matrix of dimensions c(n,p) of the p evaluation points of the functional data. Rows are associated to units and columns to the evaluation grid.

coeff.regr.eval — Matrix of dimensions c(nvar+1,p) of the p evaluation points of the intercept (first row) and the nvar effects of the covariates specified in formula. Columns are associated to the evaluation points indices.

`fitted.eval` Evaluation on a fine uniform grid of the fitted values of the functional regression.

`residuals.eval` Evaluation on a fine uniform grid of the residuals of the functional regression.

`R2.eval` Evaluation on a fine uniform grid of the functional R-squared of the regression.

`heatmap.matrix.F` Heatmap matrix of p-values of functional F-test (used only for plots).

`heatmap.matrix.t` Heatmap matrix of p-values of functional t-tests (used only for plots).

Author(s)

Alessia Pini, Simone Vantini

References

D. Freedman and D. Lane (1983). A Nonstochastic Interpretation of Reported Significance Levels. *Journal of Business & Economic Statistics* 1.4, 292-298.

B. F. J. Manly (2006). Randomization, *Bootstrap and Monte Carlo Methods in Biology*. Vol. 70. CRC Press.

A. Pini and S. Vantini (2016). The Interval Testing Procedure: A General Framework for Inference for Functional Data Analysis. MOX-report 2016, Politecnico di Milano

A. Pini and S. Vantini (2017). Interval-wise Testing for Functional Data, Journal of Non-parametric Statistics,DOI:10.1080/10485252.2017.1306627

See Also

See `summary.IWTlm` for summaries and `plot.IWTlm` for plotting the results.

See also `IWTlm_FoF` to fit and test functional-on-functional linear models applying the IWTP, and `IWT1`, `IWT2` for one-population and two-population tests.

Examples

```
# Importing the NASA temperatures data set
data(NASAtemp)
data <- rbind(temperature1,temperature2)

# Defining the covariates
temperature1 <- NASAtemp$milan
temperature2 <- NASAtemp$geneva

labels <- c(rep(0,22),rep(1,22))
lab1 <- labels
lab2 <- factor(rep(1:3,15)[-45])

# Defining responses as 'fd' object
library("fda")
m <- 4
p <- dim(data1)[2]
abscissa <- 1:p
bspl_basis <- create.bspline.basis(range(abscissa), norder = m, nbasis = p)
functional_temp <- Data2fd(argvals = abscissa, y = t(temperature2), bspl_basis)

# Performing the IWTP
IWTlm.result <- IWTlm(formula= data ~ lab1 + lab2, B = 1000, method = 'responses',dx=NULL,
                 recycle=TRUE,statistic='std',IW_scale=10)

IWTlm.result <- IWTlm(functional_temp ~ lab1 + lab2, B=1000, method='residuals',dx=NULL,
                recycle=TRUE, statistic='std',IW_scale=10)

# Summary of the IWTP results
summary(IWTP.result)

# Plot of the IWTP results
plot(IWTlm.result, plot_adjpval = TRUE, plot_unadjpval = TRUE)
```

IWTlm_FoF *Interval-Wise Testing Procedure for testing Functional-on-Functional Linear Models*

Description

IWTlm_FoF is used to fit and test functional-on-functional linear models. It can be used to carry out regression, and analysis of variance. The function implements the Interval-Wise Testing Procedure for testing the significance of the effects of functional covariates on a functional population evaluated on a uniform grid. Data are represented by means of class fd object or embedded in a matrix of dimensions c(n,p). The significance of each functional evaluation point is tested with a control of the interval-wise error rate.

Usage

```
IWTlm_FoF(formula, B=1000, method='residuals', dx=NULL, recycle=TRUE,
          statistic='std', IW_scale=2)
```

Arguments

formula
: An object of class "formula" specifying the model to fit. Example: y ~ x1 + x2 where y is a matrix of dimension c(n,p) containing the point-wise evaluations of the n functional data on p points or an object of class fd (see fda package) containing the functional data set. The covariates x1, x2 can be either functional as matrix of dimensions c(n,p) or scalar as n-dimensional vectors.

B
: The number of iterations of the MC algorithm to evaluate the p-values of the permutation tests. The default is B=1000.

method
: Permutation method used to calculate the p-value of permutation tests. Choose "residuals" for the permutations of residuals under the reduced model, according to the Freedman and Lane scheme, and "responses" for the permutation of the responses, according to the Manly scheme.

dx
: Uniform discretization grid step size, for objects of class fd. The default is dx=NULL.

recycle
: Flag specifying whether the recycled version of IWT has to be used. The default is recycle=TRUE.

statistic
: Calculation mode of the statistic used for the covariates partial tests (t-test).The mention 'nstd' refers to non-standardized statistics, 'std' refers to standardized statistics, i.e divided by the standard-error.

IW_scale — Scale parameter that determine the depth of the interval-wise testing procedure: maxrows. maxrows= p-IW_scale+1 defines the maximum size of the permutation intervals use in the testing procedure.The default is IW_scale=2.

Value

IWTlm_FoF returns an object of class "IWTlm".

The function summary is used to obtain and print a summary of the results.

An object of class "IWTlm" is a list containing at least the following components:

call — The matched call.

design_matrix — The design matrix of the functional-on-functional linear model.

unadjusted_pval_F — Unadjusted p-values of the functional F-test for each evaluation point.

unadjusted_pval_part — Unadjusted p-values of the functional t-tests for each partial regression coefficients.

pval_matrix_F — Matrix of dimensions c(p,p) of the p-values of the multivariate F-tests. The element (i,j) of matrix pval_matrix contains the p-value of the joint NPC test of the evaluation points (j,j+1,...,j+(p-i)).

pval_matrix_part — Array of dimensions c(nvar+1,p,p) of the p-values of the multivariate t-tests. Where nvar+1 corresponds to the number of covariates nvar plus the intercept. The element (n,i,j) of array pval_matrix contains the p-value of the joint NPC test on covariate number n of the evaluation points (j,j+1,...,j+(p-i)).

adjusted_pval_F — Adjusted p-values of the functional F-test for each evaluation point.

adjusted_pval_part — Adjusted p-values of the functional t-tests for each partial regression coefficient.

data.eval — Matrix of dimensions c(n,p) of the p evaluation point of the functional data. Rows are associated to units and columns to the evaluation grid.

coeff.regr.eval — Matrix of dimensions c(nvar+1,p) of the p evaluation points of the intercept (first row) and the nvar effects of the covariates specified in formula. Columns are associated to the evaluation points indices.

fitted.eval — Evaluation on a fine uniform grid of the fitted values of the functional regression.

residuals.eval — Evaluation on a fine uniform grid of the residuals of the functional regression.

R2.eval — Evaluation on a fine uniform grid of the functional R-squared of the regression.

heatmap.matrix.F — Heatmap matrix of p-values of functional F-test (used only for plots).

heatmap.matrix.t — Heatmap matrix of p-values of functional t-tests (used only for plots).

Author(s)

Alessia Pini, Simone Vantini

References

D. Freedman and D. Lane (1983). A Nonstochastic Interpretation of Reported Significance Levels. *Journal of Business & Economic Statistics* 1.4, 292-298.

B. F. J. Manly (2006). Randomization, *Bootstrap and Monte Carlo Methods in Biology*. Vol. 70. CRC Press.

A. Pini and S. Vantini (2016). The Interval Testing Procedure: A General Framework for Inference for Functional Data Analysis. MOX-report 2016, Politecnico di Milano

A. Pini and S. Vantini (2017). Interval-wise Testing for Functional Data, Journal of Non-parametric Statistics,DOI:10.1080/10485252.2017.1306627

See Also

See `summary.IWTlm` for summaries and `plot.IWTlm` for plotting the results.

See also `IWTlm` to fit and test functional-on-scalar linear models applying the IWTP, and `IWT1`, `IWT2` for one-population and two-population tests.

Examples

```
# Importing the NASA temperatures data set
data(NASAtemp)

# Defining the covariates
years <- 1:22
temperature1 <- NASAtemp$milan
temperature2 <- NASAtemp$geneva

# Defining responses as 'fd' object
library("fda")
m <- 4
p <- dim(data1)[2]
abscissa <- 1:p
bspl_basis <- create.bspline.basis(range(abscissa), norder = m, nbasis = p)
functional_temp <- Data2fd(argvals = abscissa, y = t(temperature2), bspl_basis)

# Performing the IWTP
IWTlm.result <- IWTlm_FoF(temperature2 ~ years + temperature1, B=1000,
                method='residuals',dx=NULL, recycle=TRUE, statistic='std',IW_scale=30)

IWTlm.result <- IWTlm_FoF(functional_temp ~ years, B=1000, method='responses',dx=NULL,
                recycle=TRUE, statistic='std',IW_scale=35)

# Summary of the IWTP results
summary(IWTP.result)

# Plot of the IWTP results
plot(IWTlm.result, plot_adjpval = TRUE, plot_unadjpval = TRUE)
```

`NASAtemp` *NASA daily temperatures data set*

Description

It contains the daily mean temperatures registered from July 1983 to June 2005 and stored in the NASA database Earth Surface Meteorology for Solar Energy of three different geographical locations: the region (45-46 North, 9-10 East), including the city of Milan (Italy), the region (48-49 North, 2-3 East), including the city of Paris (France), and the region (46-47 North, 6-7 East), including the city of Geneva (Switzerland).

Usage

```
data(NASAtemp)
```

Format

List of 3 elements:

- `milan`Matrix of dimensions `c(22,365)` containing the daily mean temperatures of the region (45-46 North, 9-10 East), including the city of Milan (Italy) registered from July 1983 to June 2005 (22 years).
- `paris` Matrix of dimensions `c(22,365)` containing the daily mean temperatures of the region (48-49 North, 2-3 East), including the city of Paris (France) registered from July 1983 to June 2005 (22 years).
- `geneva` Matrix of dimensions `c(22,365)` containing the daily mean temperatures of the region (46-47 North, 6-7 East), including the city of Geneva (Switzerland) registered from July 1983 to June 2005 (22 years).

Source

These data were obtained from the NASA Langley Research Center Atmospheric Science Data Center Surface meteorological and Solar Energy (SSE) web portal supported by the NASA LaRC POWER Project. Data are freely available at: NASA Surface Meteorology and Solar Energy, A Renewable Energy Resource web site (release 6.0): http://eosweb.larc.nasa.gov

Examples

```
data(NASAtemp)
## Not run:

matplot(t(NASAtemp$milan),type='l')
matplot(t(NASAtemp$paris),type='l')
matplot(t(NASAtemp$geneva),type='l')

## End(Not run)
```

plot.IWT1 *Plotting IWTP results for one-population tests*

Description

plot method for class "IWT1". Plotting function creating a graphical output of the IWTP for the test of the mean of one population: functional data and interval-wise adjusted p-values are plotted.

Usage

```
## S3 method for class 'IWT1'
plot(x, xrange = c(0, 1), alpha1 = 0.05, alpha2 = 0.01,
     ylab = "Functional Data", main = NULL, lwd = 1, col = 1,
     ylim = NULL, type='l', ...)
```

Arguments

x The object to be plotted. An object of class "IWT1", that is, a result of an IWT for comparison between two populations. Usually a call to IWT1.

xrange Range of the x axis.

alpha1 First level of significance used to select and display significant differences. Default is alpha1 = 0.05.

alpha2 Second level of significance used to select and display significant differences. Default is alpha1 = 0.01. alpha1 and alpha2 are s.t. alpha2 < alpha1. Otherwise the two values are switched.

ylab Label of y axis of the plot of functional data. Default is "Functional Data".

main An overall title for the plots (it will be pasted to "Functional Data" for the first plot and "adjusted p-values" for the second plot).

lwd Line width for the plot of functional data.

col Color used to plot the functional data.

ylim Range of the y axis.

type Type of plotted lines.

... Additional plotting arguments that can be used with function plot, such as graphical parameters (see par).

Value

No value returned. The function produces a graphical output of the IWT results: the plot of the functional data and the one of the adjusted p-values. The intervals selected as significant by the test at level alpha1 and alpha2 are highlighted in the plot of the interval-wise adjusted p-values and in the one of functional data by gray areas (light and dark gray, respectively).

Author(s)

Alessia Pini, Simone Vantini

References

A. Pini and S. Vantini (2016). The Interval Testing Procedure: A General Framework for Inference for Functional Data Analysis. MOX-report 2016, Politecnico di Milano

A. Pini and S. Vantini (2017). Interval-wise Testing for Functional Data, Journal of Non-parametric Statistics,DOI:10.1080/10485252.2017.1306627

See Also

`IWTimage` for the plot of p-values heatmaps.

See also `IWT1` to perform the IWTP to test for the mean of a functional populations. See `plot.IWT2` and `plot.IWTlm` for the plot method applied to the IWTP results of two-population tests and linear models, respectively.

Examples

```
# Importing the NASA temperatures data set
data(NASAtemp)

# Performing the IWT for one population
IWT.result<- IWT1(NASAtemp$paris,mu=4,IW_scale=30)

# Plotting the results of the IWT
plot(IWT.result,xlab='Day',xrange=c(0,365),main='Paris temperatures')

# Selecting the significant domain part for the radius at 5% level
which(IWT.result$adjusted_pval < 0.05)
```

`plot.IWT2` *Plotting IWTP results for two-populations tests*

Description

plot method for class "`IWT2`". Plotting function creating a graphical output of the IWT for the test of comparison between two populations: functional data and IWT-adjusted p-values are plotted.

Usage

```
## S3 method for class 'IWT2'
plot(x, xrange = c(0, 1), alpha1 = 0.05, alpha2 = 0.01,
     ylab = "Functional Data", main = NULL, lwd = 1,
     col = c(1, 2), ylim = NULL, type='l' , ...)
```

Arguments

`x` The object to be plotted. An object of class "`IWT2`", that is, a result of an IWT for comparison between two populations. Usually a call to `IWT2`.

`xrange` Range of the `x` axis.

alpha1 First level of significance used to select and display significant differences. Default is alpha1 = 0.05.

alpha2 Second level of significance used to select and display significant differences. Default is alpha1 = 0.01. alpha1 and alpha2 are s.t. alpha2 < alpha1. Otherwise the two values are switched.

ylab Label of y axis of the plot of functional data. Default is "Functional Data".

main An overall title for the plots (it will be pasted to "Functional Data" for the first plot and "adjusted p-values" for the second plot).

lwd Line width for the plot of functional data.

col Color used to plot the functional data.

ylim Range of the y axis.

type Type of plotted lines.

... Additional plotting arguments that can be used with function plot, such as graphical parameters (see par).

Value

No value returned. The function produces a graphical output of the IWTP results: the plot of the functional data and the one of the adjusted p-values. The intervals selected as significant by the test at level alpha1 and alpha2 are highlighted in the plot of the adjusted p-values and in the one of functional data by gray areas (light and dark gray, respectively).

Author(s)

Alessia Pini, Simone Vantini

References

A. Pini and S. Vantini (2016). The Interval Testing Procedure: A General Framework for Inference for Functional Data Analysis. MOX-report 2016, Politecnico di Milano

A. Pini and S. Vantini (2017). Interval-wise Testing for Functional Data, Journal of Non-parametric Statistics,DOI:10.1080/10485252.2017.1306627

See Also

IWTimage for the plot of p-values heatmaps.

See also IWT2 to perform the IWT to test for differences between two populations. See plot.IWT1 and plot.IWTlm for the plot method applied to the IWTP results of one-population tests and a linear models, respectively.

Examples

```
# Importing the NASA temperatures data set
data(NASAtemp)

# Performing the IWT for one population
IWT.result<- IWT2(NASAtemp$paris,NASAtemp$milan,mu=4,IW_scale=30)

# Plotting the results of the IWT
plot(IWT.result,xlab='Day',xrange=c(0,365),main='Paris vs Milan temperatures')
```

```
# Selecting the significant domain part for the radius at 5% level
which(IWT.result$adjusted_pval < 0.05)
```

---

plot.IWTaov *Plotting IWT results for functional analysis of variance testing*

Description

plot method for class "IWTaov". Plotting function creating a graphical output of the IWTP for the test on a functional analysis of variance: functional data, and interval-wise adjusted p-values of the F-tests on the whole model and on each factor are plotted.

Usage

```
## S3 method for class 'IWTaov'
plot(x,xrange=c(0,1), alpha1=0.05, alpha2=0.01,
     plot_adjpval=TRUE, plot_unadjpval = FALSE, ylim= NULL,col=1,
     ylab='Functional Data',main=NULL,lwd=1,type='l',...)
```

Arguments

x	The object to be plotted. An object of class "IWTaov", usually, a result of a call to IWTaov.
xrange	Range of the x axis.
alpha1	First level of significance used to select and display significant effects. Default is alpha1 = 0.05.
alpha2	Second level of significance used to select and display significant effects. Default is alpha1 = 0.01. alpha1 and alpha2 are s.t. alpha2 < alpha1. Otherwise the two values are switched.
plot_adjpval	A logical indicating whether the plots of adjusted p-values have to be done. Default is plot_adjpval = TRUE.
plot_unadjpval	A logical indicating whether the plots of unadjusted p-values have to be done. Default is plot_unadjpval = FALSE.
col	Colors for the plot of functional data. Default is col = 1.
ylim	Range of the y axis. Default is ylim = NULL.
ylab	Label of y axis of the plot of functional data. Default is "Functional Data".
main	An overall title for the plots (it will be pasted to "Functional Data and F-test" for the first plot and "factor" for the other plots).
lwd	Line width for the plot of functional data. Default is lwd=1.
type	Type of plotted lines.
...	Additional plotting arguments that can be used with function plot, such as graphical parameters (see par).

Value

No value returned. The function produces a graphical output of the IWTP results: the plot of the functional data, functional regression coefficients, and interval-wise adjusted p-values of the F-tests on the whole model and on each factor. The intervals selected as significant by the tests at level `alpha1` and `alpha2` are highlighted in the plot of the interval-wise adjusted p-values and in the one of functional data, by gray areas (light and dark gray, respectively). The first plot reports the gray areas corresponding to a significant F-test on the whole model. The remaining plots report the gray areas corresponding to significant F-tests on each factor (with colors corresponding to the levels of the factor).

Author(s)

Alessia Pini, Simone Vantini

References

A. Pini and S. Vantini (2016). The Interval Testing Procedure: A General Framework for Inference for Functional Data Analysis. MOX-report 2016, Politecnico di Milano

A. Pini and S. Vantini (2017). Interval-wise Testing for Functional Data, Journal of Non-parametric Statistics,DOI:10.1080/10485252.2017.1306627

K. Abramowicz, S. De Luna, C. Häger, A. Pini, L. Schelin, and S. Vantini (2015). Distribution-Free Interval-Wise Inference for Functional-on-Scalar Linear Models. *MOX-report 3/2015*, Politecnico di Milano.

See Also

See also `IWTaov` to fit and test a functional analysis of variance applying the IWTP, and `summary.IWTaov` for summaries. See `plot.IWTlm`, `plot.IWT1`, and `plot.IWT2` for the plot method applied to the IWTP results of functional-on-scalar or functional-on-functional linear models, one-population and two-population, respectively.

Examples

```
# Importing the NASA temperatures data set
data(NASAtemp)

temperature <- rbind(NASAtemp$milan,NASAtemp$paris)
groups <- c(rep(0,22),rep(1,22))

# Performing the IWTP
IWT.result <- IWTaov(temperature ~ groups)

# Plot of the IWT results
plot(IWT.result, main='NASA data temperature', plot.adjpval = TRUE,
     plot_unadjpval = TRUE, xlab='Day',xrange=c(1,365))
```

plot.IWTlm *Plotting IWTP results for functional-on-scalar and functional-on-functional linear model testing*

Description

plot method for class "IWTlm". Plotting function creating a graphical output of the IWT for the test on a functional-on-scalar and functional-on-functional linear models: functional data and interval-wise adjusted p-values for the F-test and t-tests are plotted.

Usage

```
## S3 method for class 'IWTlm'
plot(x, xrange = c(0, 1), alpha1 = 0.05, alpha2 = 0.01,
                      plot_adjpval = TRUE, plot_unadjpval = FALSE,
                      col = c(1,rainbow(dim(x$adjusted_pval_part)[1])),
                      ylim = NULL, ylab = "Functional Data", main = NULL,
                      lwd = 1,type='l' ...)
```

Arguments

x The object to be plotted. An object of class "IWTlm", usually, a result of a call to IWTlm or IWTlm_FoF.

xrange Range of the x axis.

alpha1 First level of significance used to select and display significant effects. Default is alpha1 = 0.05.

alpha2 Second level of significance used to select and display significant effects. Default is alpha1 = 0.01. alpha1 and alpha2 are s.t. alpha2 < alpha1. Otherwise the two values are switched.

plot_adjpval A logical indicating whether the plots of adjusted p-values have to be done. Default is plot_adjpval = TRUE.

plot_unadjpval A logical indicating whether the plots of unadjusted p-values have to be done. Default is plot_unadjpval = FALSE.

col Vector of colors for the plot of functional data (first element), and covariates (following elements). Default is col = c(1,rainbow(dim(x$adjusted_pval_part)[1])).

ylim Range of the y axis. Default is ylim = NULL.

ylab Label of y axis of the plot of functional data. Default is "Functional Data".

main An overall title for the plots (it will be pasted to "Functional Data and F-test" for the first plot and "t-test" for the other plots).

lwd Line width for the plot of functional data. Default is lwd=1.

type Type of plotted lines.

... Additional plotting arguments that can be used with function plot, such as graphical parameters (see par).

Value

No value returned. The function produces a graphical output of the IWTP results: the plot of the functional data, functional regression coefficients, and interval-wise adjusted p-values for the F-test and t-tests. The intervals selected as significant by the tests at level `alpha1` and `alpha2` are highlighted in the plot of the adjusted p-values and in the one of functional data, by gray areas (light and dark gray, respectively). The plot of functional data reports the gray areas corresponding to a significant F-test. The plots of functional regression coefficients report the gray areas corresponding to significant t-tests for the corresponding covariate.

Author(s)

Alessia Pini, Simone Vantini

References

A. Pini and S. Vantini (2016). The Interval Testing Procedure: A General Framework for Inference for Functional Data Analysis. MOX-report 2016, Politecnico di Milano

A. Pini and S. Vantini (2017). Interval-wise Testing for Functional Data, Journal of Non-parametric Statistics,DOI:10.1080/10485252.2017.1306627

K. Abramowicz, S. De Luna, C. Häger, A. Pini, L. Schelin, and S. Vantini (2015). Distribution-Free Interval-Wise Inference for Functional-on-Scalar Linear Models. *MOX-report 3/2015*, Politecnico di Milano.

See Also

See also `IWTlm` or `IWTlm_FoF` to fit and test a functional-on-scalar or functional-on-functional linear models applying the IWT, and `summary.IWTlm` for summaries. See `plot.IWTaov`, `plot.IWT1` and `plot.IWT2` for the plot method applied to the IWT results of functional analysis of variance, one-population and two-population, respectively.

Examples

```
# Importing the NASA temperatures data set
data(NASAtemp)
data <- rbind(NASAtemp$milan,NASAtemp$paris)

# Defining covariate
lab <- c(rep(0,22),rep(1,22))

# Performing the IWTlm
IWT.result <- IWTlm(data ~ lab, IW_scale=10)

IWT.result <- IWTlm_FoF(data ~ lab, IW_scale=10)

# Plot of the IWT results
plot(IWT.result, plot_adjpval = TRUE, plot_unadjpval = TRUE,
           col = c(1, rainbow(dim(IWT.result$adjusted_pval_part)[1])),
           main='NASA data',xlab='Day',xrange=c(1,365))
```

summary.IWTaov *Summarizing Functional Analysis of Variance Fits*

Description

summary method for class "IWTaov".

Usage

```
## S3 method for class 'IWTaov'
summary(object, ...)
```

Arguments

object	An object of class "IWTaov", usually, a result of a call to IWTaov.
...	Further arguments passed to or from other methods.

Value

The function summary.IWTaov computes and returns a list of summary statistics of the fitted functional analysis of variance given in object, using the component "call" from its arguments, plus:

factors	A matrix of dimensions c(nvar,1) displaying for each nvar factors of the ANOVA, the corresponding (two-sided) interval-wise adjusted minimum p-values of the corresponding tests of significance (i.e., the minimum p-value over all p evaluation points of the functional data).
R2	Range of the functional R-squared.
ftest	Interval-wise adjusted minimum p-value of functional F-test.

Author(s)

Alessia Pini, Simone Vantini

References

A. Pini and S. Vantini (2013). The Interval Testing Procedure: Inference for Functional Data A. Pini and S. Vantini (2016). The Interval Testing Procedure: A General Framework for Inference for Functional Data Analysis. MOX-report 2016, Politecnico di Milano

A. Pini and S. Vantini (2017). Interval-wise Testing for Functional Data, Journal of Non-parametric Statistics,DOI:10.1080/10485252.2017.1306627

K. Abramowicz, S. De Luna, C. Häger, A. Pini, L. Schelin, and S. Vantini (2015). Distribution-Free Interval-Wise Inference for Functional-on-Scalar Linear Models. *MOX-report 3/2015*, Politecnico di Milano.

See Also

See IWTaov for fitting and testing the functional ANOVA and plot.IWTaov for plots. See also IWTlm, IWTlm_FoF, IWT1 and IWT2.

Examples

```
# Importing the NASA temperatures data set
data(NASAtemp)
temperature <- rbind(NASAtemp$milan,NASAtemp$paris)

# Defining covariates
groups <- c(rep(0,22),rep(1,22))

# Performing the IWT
IWT.result <- IWTaov(temperature ~ groups)

# Summary of the IWT results
summary(IWT.result)
```

summary.IWTlm *Summarizing Functional-on-Scalar and Functional-on-Functional Linear Models Fits*

Description

summary method for class "IWTlm".

Usage

```
## S3 method for class 'IWTlm'
summary(object, ...)
```

Arguments

object	An object of class "IWTlm", usually, a result of a call to IWTlm or IWTlm_FoF.
...	Further arguments passed to or from other methods.

Value

The function summary.IWTlm computes and returns a list of summary statistics of the fitted functional-on-scalar or functional-on-functional linear models given in object, using the component "call" from its arguments, plus:

ttest	A matrix of dimensions c(nvar+1,1) displaying for each nvar +1 covariates plus intercept the corresponding (two-sided) interval-wise adjusted minimum p-values of t-tests (i.e., the minimum p-value over all p evaluation points of the functional data).
R2	Range of the functional R-squared.
ftest	Interval_wise adjusted minimum p-value of functional F-test.

Author(s)

Alessia Pini, Simone Vantini

References

A. Pini and S. Vantini (2016). The Interval Testing Procedure: A General Framework for Inference for Functional Data Analysis. MOX-report 2016, Politecnico di Milano

A. Pini and S. Vantini (2017). Interval-wise Testing for Functional Data, Journal of Non-parametric Statistics,DOI:10.1080/10485252.2017.1306627

K. Abramowicz, S. De Luna, C. Häger, A. Pini, L. Schelin, and S. Vantini (2015). Distribution-Free Interval-Wise Inference for Functional-on-Scalar Linear Models. *MOX-report 3/2015*, Politecnico di Milano.

See Also

See `IWTlm` and `IWTlm_FoF` for fitting and testing the functional linear models and `plot.IWTlm` for plots. See also `IWTaov`, `IWT1` and `IWT2`.

Examples

```
# Importing the NASA temperatures data set
data(NASAtemp)
temperature <- rbind(NASAtemp$milan,NASAtemp$paris)

# Defining covariates
groups <- c(rep(0,22),rep(1,22))

# Performing the IWT
IWT.result <- IWTlm(temperature ~ groups,IW_scale=10)

# Summary of the IWT results
summary(IWT.result)
```

Conclusion

In Pini and Vantini (2016), the issue of testing functional data in order to assess test of domain interval selection have already been studied. It has been studied by the bias of the decomposition of the functional data on functional local basis (e.g B-splines), and then univariate and multivariate tests on the coefficients of the basis expansion are performed.

But a real domain selection, where the null hypothesis is violated, cannot be assessed with Fourier basis expansion or other non-parametric smoothing methods (e.g Kernel's method). But the interval-wise test proposed here focuses directly on functional data, which can be obtained beforehand, from any smoothing methods.

This package provides an implementation of the innovative IWT procedure, associated with the non-parametric permutation test, to evaluate the test significance, with an interval-wise control of type-I error, of each evaluation points of the discrete grid.

To conclude, this procedure achieves to enlarge the scope of inference in functional data framework, focusing directly, decomposing, remixing the domain of the curves instead of handling the whole domain as only one static block. This inferential perspective is embedded in the concepts of unadjusted and adjusted p-value that are the true innovative quantities provided by the method, and will very useful and accessible tools for researchers and statisticians.

Bibliography

1. A. Pini and S. Vantini (2016). The Interval Testing Procedure: A General Framework for Inference for Functional Data Analysis. MOX-report 2016, Politecnico di Milano

2. A. Pini and S. Vantini (2017). Interval-wise Testing for Functional Data, Journal of Non-parametric Statistics,DOI:10.1080/10485252.2017.1306627

3. Ramsay, J. O, & Silverman, B. (2005). Functional data analysis. New York : Springer, c1997.

4. Alessia Pini and Simone Vantini. The interval testing procedure: Inference for functional data controlling the family wise error rate on intervals. Technical report, MOX, 2013b.

5. Alessia Pini and Simone Vantini. fdatest: Interval Testing Procedure for func- tional data, 2013a. URL http://CRAN.R-project.org/package=fdatest. R package version 1.0.

6. Alessia Pini. Permutation Tests for Univariate and Multivariate Data,