Università degli Studi di Napoli Federico II

Scuola Politecnica e delle Scienze di Base

Corso di Laurea Magistrale

in

Ingegneria Gestionale

Dipartimento di Ingegneria Industriale

Functional clustering of spot welding dynamic resistance curves in the automotive industry

Relatori

Ch.mo Prof. Biagio Palumbo

Ch.mo Prof. Antonio Lepore

Co-relatori

Ing. Christian Capezza, DII

Ing. Gianmarco Genchi, CRF S.c.P.A.

Candidato

Gianluca Napoli

Matr. M62001434

Anno Accademico 2020/2021

Ai miei genitori

e ai miei amici

Declaration

Abstract

Quality assessment of resistance spot welding (RSW) joints of metal sheets in the automotive industry is typically based on off-line ultrasonic tests on completed sub-assemblies through indirect measurement of spot weld key characteristics. However these inspections are unfeasible on large scale production. On the other hand, the modern Industry 4.0 framework opens the possibility to automatically acquire on large scale electric process parameters and profiles, such as the so-called dynamic resistance curve (DRC) that is recognised as the full technological signature of the spot welds. This process signal can be used to indirectly give information on the weld joint quality, and, when possible, to associate them to the ultrasonic inspection results. The online monitoring of the spot weld quality could be used to point out the likely unacceptable welds and guide in a data driven fashion the ultrasonic test, which are costly in terms of money and time. In order to implement this strategy a cluster analysis of these curves is crucial for creating groups of DRCs corresponding to different welding quality levels. In this way, an online control can be made by assigning new curves to specific clusters and by focusing the ultrasonic tests on the spot welds whose DRC falls into clusters possibly corresponding to low quality. In place of the standard practice, which usually relies on the statistical analysis of scalar features extracted from the DRCs, this thesis work implements the clustering analysis of the complete profile of DRCs, based on functional data analysis techniques, which are not problem-specific and avoid collapsing the information contained in these profiles into problem-specific, correlated and sometimes arbitrary features.

Contents

List of Figures

List of Tables

Introduction

Context of the study

Resistance Spot Welding is the most common technique employed in joining metal sheets during body-in-white (BIW) assembly of automobiles [25], mainly because of its high speed and adaptability for automation that makes it suitable for mass production [33]. In fact, the typical car body contains about 5000 spot welds joining a mixture of sheet metal material types and thickness [52]. However, the individual weld quality is characterised by remarkable variability because it is influenced by many factors [12]. In order to guarantee the structural integrity of the welded assembly, it is a common practice in industry adding a significant number of redundant welds and controlling the most critical spots [19].

Furthermore, the tendency in the highly competitive automobile industry is to reduce the number of RSW joints per vehicle; this fact reinforces the interest in a tool capable of assisting the quality control of spot welds. Indeed, the fewer the RSW joints per vehicle, the stronger the requirements for each of them [33].

Historically, quality assessment of RSW joints was based on destructive tests on randomly sampled spot welds selected among the most important welds on the work pieces for the welded product solidity [19]. In this way, it was possible to estimate the size of the weld nugget, formed from the solidification of the molten metal, which determines the quality of the RSW joints [37]. Afterwards, in order to reduce the control costs, non destructive testing were introduced to estimate the nugget

geometry indirectly, e.g., by the ultrasonic techniques. It is a non destructive offline testing used to control weld quality. However, a wide scale deployment of ultrasonic testing is unfeasible in practice, because it does not allow for the online process control [19].

In the modern automotive Industry 4.0 framework, welders are controlled by automatic control systems that collect a large volume of data related to the electric process parameters during spot welding. These data can be used to recreate the profiles of the process parameters, like the dynamic resistance curve (DRC) associated to the spot weld. DRC is regarded as a technological signature that depends on the metallurgical development of spot welds and can be made available with low effort [16].

So, thanks to this massive availability of data, functional data analysis (FDA) techniques can be suitably used to study Industry 4.0 processes. Functional data analysis is a branch of statistics that analyzes data providing information about curves, surfaces or anything else varying over a continuum. Under a FDA framework, each sample element is considered to be a function, defined on time domain in most of the cases. Therefore, functional clustering techniques can be used to identify DRCs groups corresponding to certain welding qualities. They avoid the need for arbitrary and often controversial feature extraction to find out homogeneous groups of DRCs, which likely pertain to spot welds sharing common mechanical and metallurgical properties. Once one has found DRC groups, an online control of the process can be made by immediately evaluating the cluster a new DRC belongs. In this way, ultrasonic testing can be more wisely made, than at random and can be focused, e.g., on those spots whose DRC are likely to belong to low quality groups, saving time or money.

ICOSAF project

ICOSAF (Integrated and collaborative systems for the intelligent factory) project aims at technologies and systems for a collaborative factory with a growing integration

of the operator in line with the principles of Industry 4.0 (interconnected automation) and 5.0 (humanisation and re-use of resources). This vision includes mobile and fixed robotic systems, active monitoring systems of quality and machinery operator assistants and AGVs that interact with operators and environment.

One branch of this project involves the Centro Ricerche Fiat s.p.a., the University of Naples Federico II and the University of Modena and Reggio Emilia. Its aim is to estimate weld quality of each resistance spot weld according to a technological signature, the dynamic resistance curve, in the manufacturing environment of FCA Melfi plant in Zona Industriale di San Nicola di Melfi and FCA Mirafiori plat in Zona Industriale di Torino.

Chapter 1

Fundamentals of Resistance Spot Welding

1.1 Physical principle

Resistance Spot Welding (RSW) is an autogenous welding process in which two or more overlapping metal sheets are joint together at discrete spots by applying pressure to the weld area and heating up the metals, at the interface among the sheets, to their melting point through Joule effect, passing an high electrical current through the spot over a given period [5]. It is one of the oldest autogenous welding processes whereby two or more sheets of metal are welded together without the use of any filler material, so it is clean and environmentally friendly. The high current flowing into the joint is associated to low voltage, so this process is safe for the operators. The process is completely automatable, so it is the most common technique employed in joining metal sheets during body-in-white (BIW) assembly of automobiles and can reach high efficiency and productivity [25].

This process requires the flow of an electric current trough the material in the welding area, so sheets of every electrically conductive material can be welded by RSW. However, the spot welding is mostly used by the automotive industry, so the

welded work pieces are made of steel or aluminium alloys. It is important to state that in these applications the electrodes are made of copper alloys. After the two sheets to be joined are positioned precisely on top of each other, two electrodes press the weld area from two opposite sides. Then the voltage applied to the electrodes generates a current flowing between them trough the material. The resistance offered by metals to current flow heats up and liquefies the metal at the faying surfaces of the work pieces by Joule effect. The electrical resistance of the metal sheets is higher compared to the copper electrodes one. This causes a higher heat generation within the sheets, especially in their contact area until the metals melt. Finally, due to the mechanical pressure of the electrodes, the molten metal cools and solidifies forming a joint between the sheets. After solidification the molten volume of material is called nugget [3]. Steps involved in RSW are shown by Figure 1.1

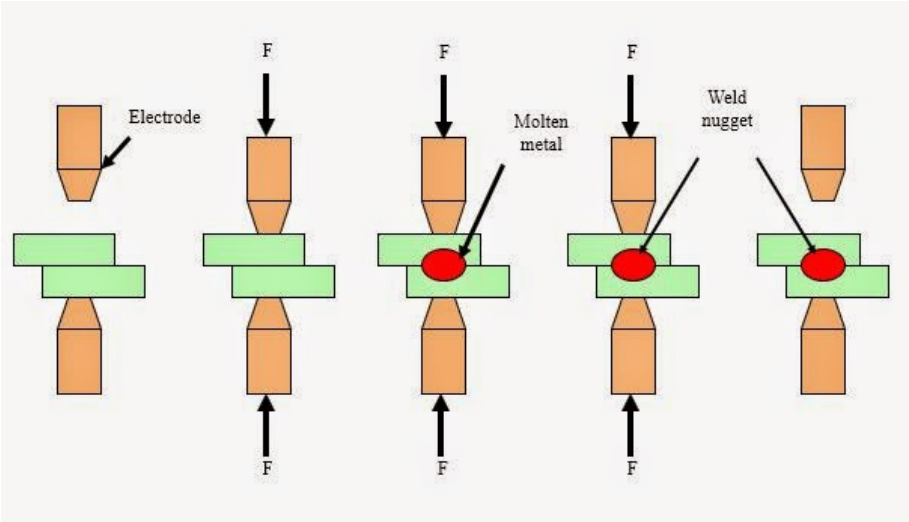

Figure 1.1: Phases of the resistance spot welding process.

The melting among the sheets is localised in the pressure point among the electrodes and the molten area is approximately equal to that of the electrode tip [5]. Because of the electrodes' low resistivity, the electrical resistance, in which the welding current flows generating heat, can be approximated to that of the welding area, defined by the electrodes' pressing. Then, assuming an electric stationary regime and that the weld area is cylindrical shaped and uniform, the resistance is given by the Ohm's

laws

$$R = \frac{V}{I}\ , \quad R = \rho\frac{L}{S},$$

where R is the electrical resistance of the welding area $[\Omega]$; V is the voltage among the electrodes' tips $[V]$; I is the welding current $[A]$; ρ is the resistivity of the sheets' metal $[\Omega m]$; L is the thickness of the overlapped metal sheets $[m]$; S is the cross sectional area of the electrode tips $[m^2]$.

This simple model of the electrical resistance in the process is based on strong assumptions, so it can be used just to approach the process complexity. The RSW is based on the Joule heating effect, which is the generation of heat energy by the flow of an electrical current through a resistance over a given period. The electric energy corresponding to the flow of the welding current through the resistance caused by the voltage among the electrodes over a given period is given by the following equations

$$Q = I^2RT = PT$$

where Q is the heat produced by joule effect $[J]$, I is the current $[A]$, R is the resistance $[\Omega]$, T is the duration of the current, that is the time for which the current is applied $[s]$, P is the electric power $[W]$, the electric energy divided by the the period.

So the amount of heat generated Q can be quantified by the Joule's law $Q = I^2RTK = PTK$, where K is a dimensionless thermal constant.

1.1.1 Electrical resistance

More precisely, the electrical resistance through which the current flows is seen as a series of resistance values, as shown in Figure 1.2 where R_1, resistance of the top electrode, R_2, contact resistance at the interface between the top electrode and the top work piece, R_3, resistance of the top work piece, R_4, resistance of the interface between the top and the bottom work pieces, R_5, resistance of the bottom work piece, R_6, contact resistance at the interface between the bottom work piece and the

bottom electrode, R_7, resistance of the bottom electrode.

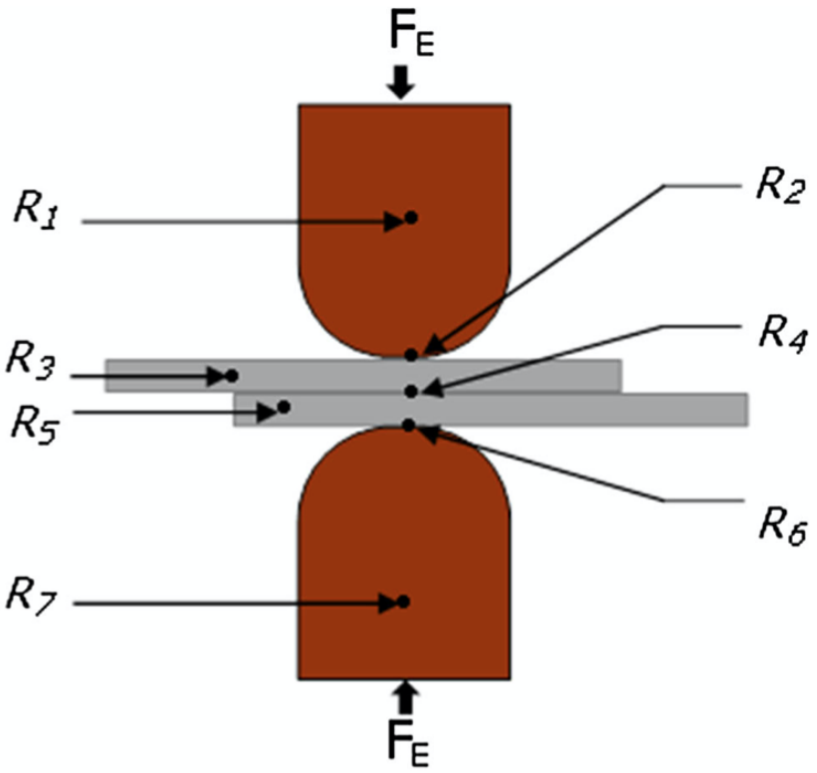

Figure 1.2: Series of Resistances

The total additive value of this electrical resistances affects the welding current fixing the voltage among the electrodes or the voltage among the electrodes fixing the welding current, so the heat generation in the weld area. The key fact is that these values will evolve, because the temperature and electrical resistivity increases during the process. Indeed, all metals exhibit an increase of resistivity with temperature. More in detail, the current is equal in all the parts of this electrical circuit, but the heat generation is not the same because of the different values of the resistances, so their values vary considerably. It is important to observe that in Figure 1.3 F_E is the electrode force $[N]$ [25].

From the second Ohm's law these resistances depend on geometries of the work pieces and the electrodes, materials of the sheets and the electrodes.

1.1.2 Contact resistance and Force

An electrical contact resistance is formed when two electrical conductors are in connection through an interface because the surfaces of the conductors are characterised by

- asperities, which allow only a small fraction of the surfaces to be in actual contact, so the current flow is constricted through small conducting spots;
- films, oxide layers, coatings or contaminants with higher resistivity than the conductive material's one.

The asperities and coatings of the interface of the metal sheets is shown in Figure 1.3.

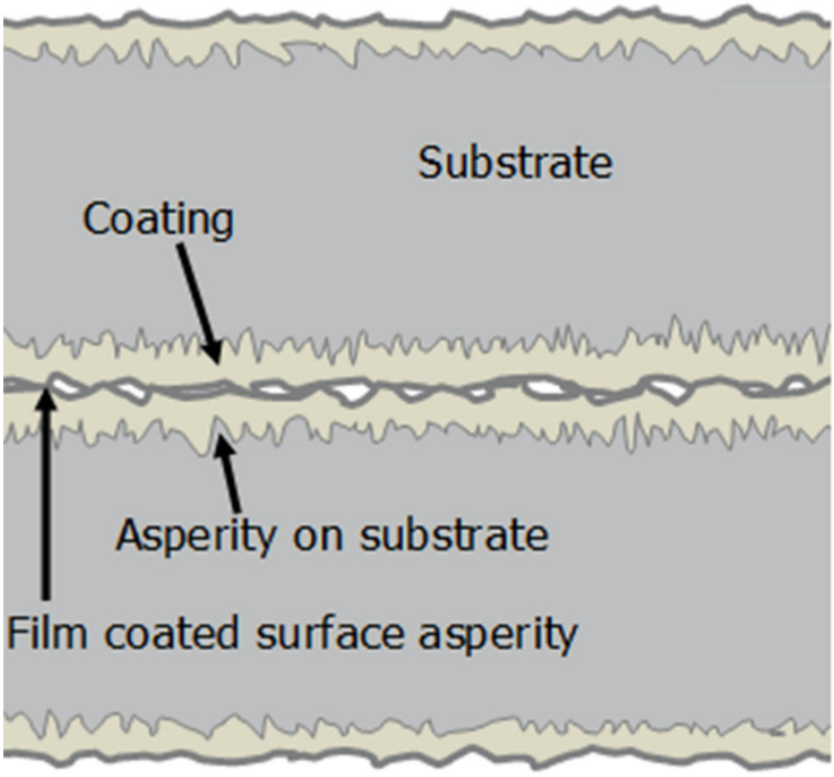

Figure 1.3: Contact Resistance

Most surface coatings on metal sheets are applied for corrosion protection or as a substrate for further surface treatment. These make more complex the tuning of the process influencing the contact resistance, so it is needed to adjust the process

parameters according to the king of coating. Some surface coatings on metal sheets are applied to facilitate the welding of difficult material combinations, so these are selected to achieve heat balance at the weld interface. Most of the surface coatings will melt before piece metal ans so it will be squeezed at the border of the weld zone by the electrode pressure, but sometimes it will remain at the weld interface as a braze metal.

The second Ohm's law states the resistance is directly proportional to the nominal contact area A_n, so in the case of the contact resistance to the force F_E, which associates intimately the surfaces clamping, the surfaces together and deforming the contact spots. The following relation

$$A_n = \frac{F_E}{\xi \cdot H}$$

is usually satisfied, where H is the contact hardness, and $0.2 < \xi < 1$.

At the same welding force and surface quality the contact resistance increases with the growth of material hardness, due to the rough surfaces asperities being more difficult to deform. This results in a small real contact area, and the contact resistance increases with roughness and oxide contamination growth, while it decreases rising the force [25]. The contact resistances at the tip to sheet and sheet to sheet interfaces are subject to the same force, but the contact areas are different because of force transmission. The contact resistance decreases with temperature, because of hardness reduction and burning of the contaminants of the surface [16].

1.1.3 Heat balance

During RSW, a part of the heat generated is lost to the surroundings by conduction, convection, and radiation. Heat balance is a function of part material and geometry, electrode material and geometry and a set of process parameters. The goal of good resistance welding is to focus the heat generated close to the weld interface at the spot where the weld is desired, so the contact resistance among the sheets has to be bigger than the others among the sheet and the electrode. In general, the highest

resistance results in the highest heat assuming that the resistance welding power supply can produce sufficient energy to overcome the resistance [5].

The contact resistance is the most important process factor related to materials in the first few milliseconds of the RSW process. Indeed, it focuses the heat generation at the faying surfaces of the metal sheets. On the peaks of these surfaces, where the contact pressure is sufficiently high, the oxide layer breaks and forms a limited number of metal-to-metal bridges. The weld current is distributed over a large area as it passes through the bulk metal. However, as it approaches the interface, the current is forced to flow through these metallic bridges. This "necking down" increases the current density, generating enough heat to cause melting. As the first of these bridges melt and collapse, new peaks come into contact, forming new bridges and additional current paths. The resistance of the molten metal is higher than that of the new bridges so that the current flow transfers from bridge-to-bridge. This process continues until the entire interface is molten. When the current stops, the electrodes rapidly cool the molten metal, which solidifies and forms a weld [16]. During the process, the resistance evolves with increasing temperature in the weld zone due to the Joule effect. This is softly counteracted by the thermal conductivity of material, thus it can be considered a dynamic resistance. This variation is due to the the evolution of its components

- the resistance of work pieces increase with temperature because of material resistivity growth;
- the contact resistance at the interface among the metal sheets reduces with temperature because of the decomposition of surface contaminants and bigger deformation softening of surface asperities caused by metal softening at high temperatures.

As well as the resistance, voltage and current also change during the process and are considered as dynamic electrical parameters during spot welding [16].

1.1.4 Materials

In the best condition, electrode material must be characterised by a high thermal conductivity and hardness, and a low electrical resistivity, compared to the work piece metal involved. This ensures the following aspects:

- the heat is generated preferentially in the work pieces, where the weld is desired, rather than in the electrodes;
- the heat is lost in the electrodes, by avoiding electrode melting, and remains in the air gap among the faying surfaces, hence heating up the metal to the welding point at the desired spot;
- the electrode deformation is reduced under the pressure to form metallic bridges at the interface despite the softening caused by heating, which is counteracted by heat transmission;
- the asperities at the interface among the sheets are deformed adequately to get a proper value of contact resistance.

The electrodes are generally made of copper or alloy, which are good conductors of current and heat, while the sheets are made of steel having low thermal conductivity and high electrical resistivity. This is plain observing the table 1.4 showing the electrical resistivity and thermal conductivity of pure commercial metals at 293 K. However the RSW can be used also for welding metals that do not satisfy this condition. In this case, the electrode material can not be characterised by a lower electrical resistance and higher thermal conductivity than the sheet, but also a higher welding point. For example, it is possible to weld aluminium sheets because of their low melting point, by using electrodes made of special alloys with higher melting point.

In different industrial contexts, the RSW is used to weld thermal and electrical conductive materials, such as copper or gold, using electrodes made of thermal and electrical non-conductive metals with higher melting point. The heat is generated mainly within the refractory electrode tip and conducted into the materials to be joined with melting. When dissimilar metals are welded, more heat will be generated

TABLE 20.1 Electrical Resistivity and Thermal Conductivity of Copper and Other Pure Commercial Metals at 293 K.

(Metal 100)	Electrical Resistivity at 293 K, $\mu\Omega$cm	Thermal Conductivity $Wm^{-1}k^{-1}$	Relative Electrical Conductivity (Copper = 100)	Relative Thermal Conductivity (Copper = 100)
Silver	1.63	419	104	106
Copper	1.694	397	100	100
Gold	2.2	316	77	80
Aluminum	2.67	238	63	60
Beryllium	3.3	194	51	49
Magnesium	4.2	155	40	39
Tungsten	5.4	174	31	44
Zinc	5.96	120	28	30
Nickel	6.9	89	24	22
Iron	10.1	78	17	20
Platinum	10.58	73	16	18
Tin	12.6	73	13	18
Lead	20.6	35	8.2	8.8
Titanium	54	22	3.1	5.5
Bismuth	117	9	1.4	2.2

Adapted from Brandes, E. A., Ed., *Smithells Metals Reference Book*, Sixth Edition, Butterworth, Inc. 1983. (Used by permission.)

Figure 1.4: Electrical Resistivity and Thermal Conductivity

in the metal with higher resistivity. This should be considered when selecting the forms of the electrodes in spot welding [3].

1.2 Process Tuning

1.2.1 Lobe Curve

As stated before, fixed the initial electrical resistance of the welding zone, the heat generation due to Joule effect, which is the core of RSW process, is proportional to the welding time and the square of the welding current. Due to the heat transfer from the weld zone to the base metals and to the electrodes, as well as to the heat loss from the free surfaces to the surrounding ones, a minimum welding current, as well as a minimum welding time, will be needed to make a weld. If the welding current is too low, there is no fusion of metal and the welding time increase do not allow for the production of a weld. When the welding current is high enough, the size of the molten volume of metal, so the weld nugget size, can be inadequate to guarantee the integrity of the joint in the operating conditions. In this case, the weld size grows

with welding time and this is established to get a proper weld. However, an excessive duration of the welding current could increase the size of the molten volume of metal beyond the electrode tip contact area, so causing the expulsion of molten metal from the weld zone to the space among metal sheets. Sometimes these conditions cause also the sticking of the electrode to the work piece [51]. These observations depend on electrical resistance, so on the material (resistivity, thermal conductivity, specific heat, latent heat of fusion, chemical composition, micro structure) and geometry of the work pieces and electrodes. Furthermore, they depend on the surface condition and value of the clamping force and therefore, they should be obtained experimentally for the specific weld. These data are expressed by mean of weld lobe or weld ability diagram, i.e., a diagram with welding current on the y-axis and welding time on the x-axis [29, 5]. An example of welding lobe is shown in Figure 1.5.

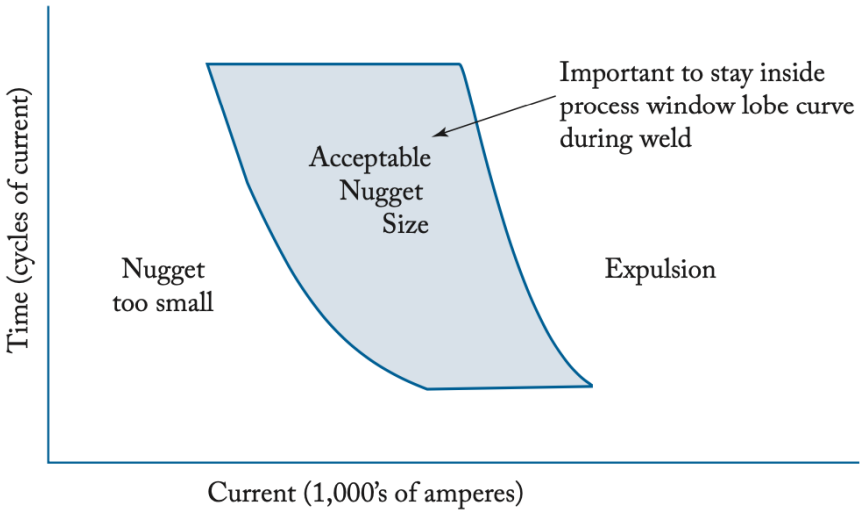

Figure 1.5: Welding Lobe

It refers to a specific value of force, a certain geometry of the electrode and some values of the metal sheets thickness [5]. The curve on the left represents the locus of nuggets whose diameter is big enough. The curve on the right represents the locus of nuggets whose diameter is bigger than the containment region at the interface with he expulsion of molten metal. In the region at the left of the area, the process parameters do not produce a nugget or produce a too small nugget. In the region at the right, the process parameters produce expulsion of molten metal [1]. Therefore,

it is important to stay inside process lobe curve during welding. Different welding currents and other process parameter settings are required depending on the thickness of metal sheets to be welded. However, bigger volumes of metal among the electrode tips generally require larger welding currents.

1.2.2 Metallography

Figure 1.6 represents the scheme of micrograph obtained by cross sectioning the weld in the longitudinal direction. It is characterised by:

- a nugget, formed by the solidification of the mix of the molten metals of the two metal sheets.
- a heat affected zone (HAZ), the volume of material at or near the weld whose properties have been altered due to the weld heat.
- parent metal, the metal which has not been altered by the process.
- the indentation, reduction of the the metal thickness caused by the application of the force at high temperatures.

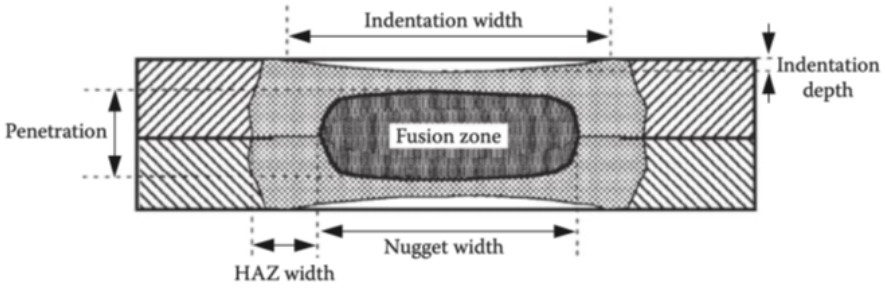

Figure 1.6: Metallographic section

The nugget has gross grain due to the relatively slow solidification, while the heat affected zone has an annealing treatment, which has reduced its strain hardening. While the former has higher properties than the parent metal, the latter has lower ones. For this reason the minimisation of the HAZ is crucial for the quality of the spot weld. A defective spot weld is characterised by inadequate dimensions of nugget and heat affected zone, besides by the presence of cracks and voids within the nugget

[51, 5]. The surface of the spot weld is characterised by a mark because of the high temperature reached at the electrode-sheet interface.

Beyond the electrical resistance, which depends on the material and geometry of the electrodes and work pieces, other process parameters are:

- I electrical current which flows through the weld area generating by Joule's effect;
- V voltage applied to the electrodes to guarantee the electric flow;
- P electrical power which is the product between the electric current and the voltage;
- T time for which the current is allowed to flow;
- P pressure applied through the electrodes to the work pieces to hold in intimate contact the parts at the joint interface.

The force, applied to the electrodes, has not to pull the work pieces together, but to ensure consistent electrical resistance and conductivity at the weld point during the flowing of current. When the flow of current stops, the electrode force is maintained, for a fraction of a second, while the weld rapidly cools and solidifies.

1.3 Weld schedule

The weld schedule is the set of the process parameters during the RSW process to get the best possible spot weld and to provide the correct heat input to the the weld zone. The most general weld schedule is shown in Figure 1.7 and includes

- **squeeze time**, the amount of time that the electrode force is applied to the welding zone before current begins to flow in order to stabilise it;
- **electrode force**, the force applied on the weld zone to provide electrical continuityamong the sheets establishing the correct value of contact resistance;
- **weld time**, the amount of time that the welding current flows in the weld

zone;

- **welding current**, the current flowing through the resistance of the secondary circuit which generate heat for metal melting;
- **quench time**, the amount of time that the electrode forging force is applied before the temper current begins to flow;
- **temper current**, the current that temper the joint;
- **temper time**, the amount of time that the temper current flows in the weld zone;
- **hold time**, the amount of time that the electrode force is applied after current stops to flow allowing the molten metal at the faying surfaces of the work pieces to solidify under pressure before the electrodes are removed;

It is important to note that the temper of the joint using the joule effect is optional as quench time, temper current and temper time. Its goal is to increase the toughness of the weld joint [15, 29].

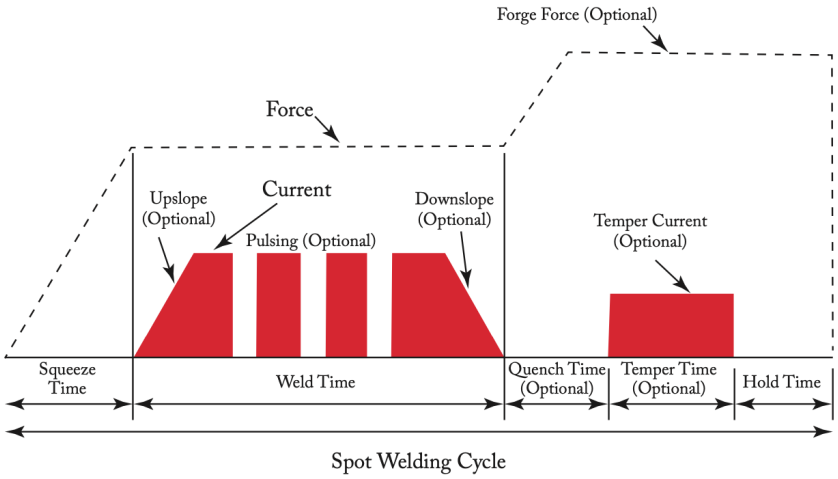

Figure 1.7: Weld Schedule

1.4 Equipment

A resistance spot welder is composed by:

- the **power supply**, which extracts energy from the power line and provide the energy to the welding joint for melting of the metal;
- the **welding control**, which modulates and controls the power supply, so the energy extracted from the power line and feed to the weld according to a user defined or user programmed "weld schedule";
- the **mechanical system**, which is formed by the elements which directly. allow the execution of the process.

The Mechanical System is composed by:

- the **electrodes**, which apply pressure the metal sheets in the zone to be welded, conduct the welding current to the focal point of the pressure and facilitating the cooling of the weld zone;
- a **gun**, on which the electrodes are mounted, which positions the electrodes in the zone where the spot weld has to be created and provides the force to the electrodes;
- a **cooling system**, which use the flow of water or air through a pipe in the electrode to cool it maintaining their integrity of shape and characteristics of thermal and electrical conductivity under working conditions.

1.4.1 Gun

There are two different types of gun according to the motion of the electrodes:

- the c-type gun, which provides high rigidity and high tooling flexibility because of the collinear motion of the electrodes. So, it is used for welding of thick materials with high applied forces.;
- the x-type (scissor) gun, which provides low rigidity and low tooling flexibility because the motion of the electrodes is not collinear. In this case the electrodes have to be doom shaped. Despite these limitations, it is used because of the great reachable work space.

Figure 1.8 shows an example of each kind of gun. [15]

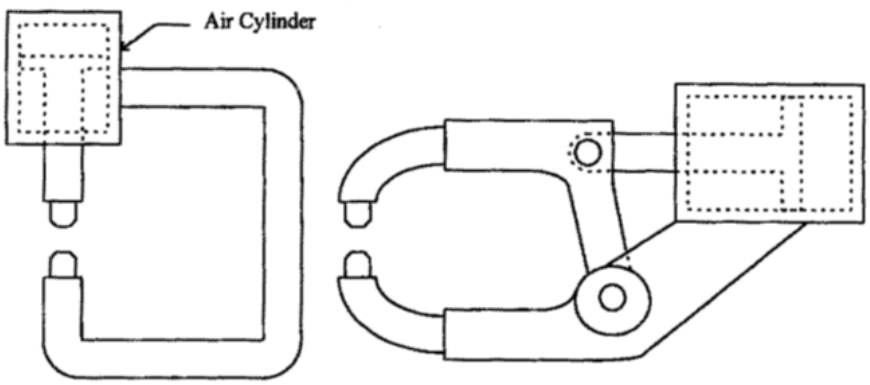

Figure 1.8: Types of Gun

The gun is provided with a couple of electrode holders. These have various shapes and sizes, like the electrodes, that have to be positioned in them. The taper serves three functions: electrical contact, watertight seal for the cooling system and retention of the electrode. Special offset holders are available and designed to reach locations not accessible to straight electrode holders. It is important to observe that the holder size has to be enough big to provide sufficient wall thickness to resist the expanding force caused by the tapered electrode expansion during the welding process because of heating. For this purpose holder sizes have been standardised, but other combinations may be used even if increasing the risk of holder taper damage [3].

1.4.2 Electrodes

The first function of the electrodes is to conduct the welding current to the welding zone in order to heat the work piece metal to the melting point. So, they have to be made with high electrical conductivity metals.

The second function of the electrodes is to apply a force in the weld zone in order to focus the welding and the heating in the region where the spot weld has to be created by pursuing this task they forge the metal. For this reason, they are subjected to a high tension state at elevated temperature. Then, electrodes must be made with hard materials that avoid excessive deformation and drift from the nominal geometry.

The third function of the electrodes to dissipate heat from the weld zone during cool time. So, they must be made with high thermal conductivity metals.

The material of the electrodes have to be resilient to high temperatures and maintain the shape integrity and the nominal characteristics of thermal and electrical conductivity under working conditions, despite of the cooling system. This can retard the unavoidable wear of the electrodes increasing electrode life for the stability of the weld quality. These properties guarantee the proper evolution of the heat balance in the weld one. Clearly, a cooling tube permitting cooling water near the welding surface improve weld quality and electrode life. The use of high electrical and thermal conductivity metal for the electrodes avoid the welding of the electrodes to the sheet surface, because of overheating beyond the annealing temperature that cause tip pickup, the alloying between the work piece and electrode, and/or sticking, the attachment of the work piece to the electrode. The severity of these conditions is influenced also by the electrode geometry, the welding schedule and the cooling system. These variables have to be adjusted to reduce the interface temperature between the electrode and the work piece.

According to the Resistance Welding Manufacturing Alliance (RWMA), an industry partner of the American Welding Society (AWS), the materials being used for the electrodes fall into three groups:

Group A, Copper Base Alloys;

Group B, Refractory Metals and Refractory Metal Composites;

Group C, Speciality Materials.

These groups are subdivided in classes according to their properties and recommended applications, the Group A is used in the automotive industry.

The quality of the RSW is influenced by electrodes. Then, it is necessary to select a material in the proper class, proper geometry and proper size for the application. The electrode tip geometry, in terms of size and contour, influences the welding pressure and current density which must be within an acceptable range. Incorrect tip face geometry will also result in increased surface marking. The size and geometry

of the electrodes follow the suggested procedure of the American Welding Society (AWS) or those of the equipment supplier, which have been used to optimise the electrode setup by trial and error procedure.

The electrode consists of the shank end, which fits into the holder, and the nose end, which contacts the work piece. Although there is a great variety of electrodes commercially available in term of alloys, types, sizes and shapes, six standard nose configurations have been designed to fulfil the majority of the welding requirements. These are **A** (Pointed); **B** (Dome); **C** (Flat); **D** (Offset); **E** (Truncated); **F** (Radius).

Among these, which are shown in Figure 1.9 [29], the most frequently used for spot welding are C, F, B. Therefore, most of the available welding schedules are based on one of these three shapes.

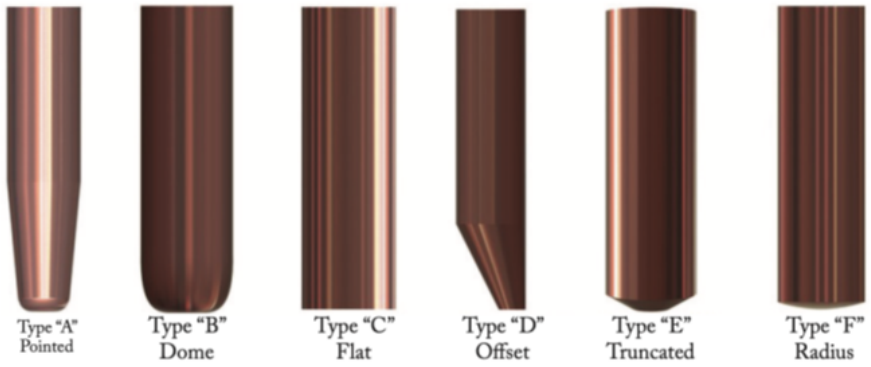

Figure 1.9: Typical Spot Welding Electrode Geometries

The types A, B and E are most commonly used for general welding applications, while C and F are used where minimal surface marking is required. Besides, D is necessary when a weld must be closer to upturned flanges or corners. For each configuration, the electrodes are produced using a number of different alloys and providing the best combination of material and geometry, that is of electrical and mechanical properties, for each process. The other sizes and shapes for the nose are often required for special applications, e.g., to conform to the contour of the weld zone or to suit other conditions.

The three standard shank designs, as shown in Figure 1.10, are: **tapered**, often used because they are easily installed and removed, showed in Figure 1.10; **threaded**,

used for high force applications and uniform height adjustment; **straight**, used for ease of positioning.

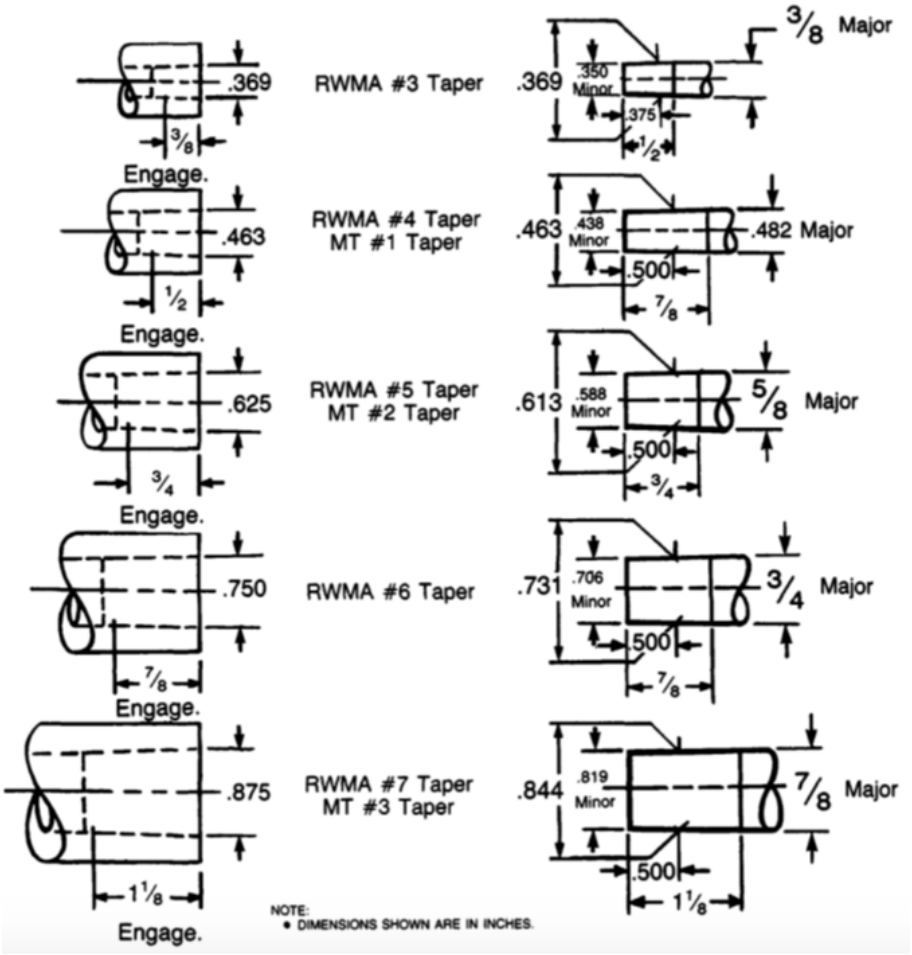

Figure 1.10: Tapered Electrodes

The electrode usually consists of two parts: an adapter shank and a replaceable cap. The electrode wear mainly concerns the nose at contact with the work. So this solution offers economical advantages because when the nose geometry is worn out, the cap needs to be replaced, only, whose cost is far less than a standard one-piece electrode (a tuffcap shank will normally outlast twenty caps). In addition, electrodes warehouse stock can be kept small because all nose designs will fit the same size shank. Figure 1.11 shows male adapter and female caps.

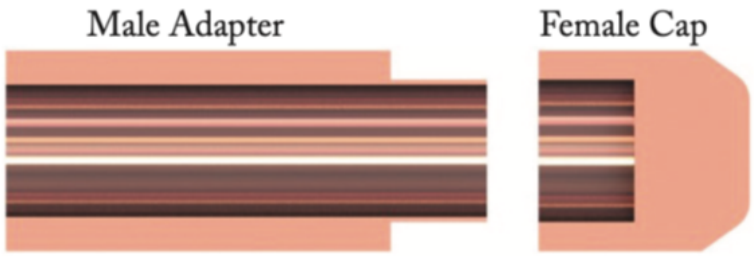

Figure 1.11: Male Adapter and Female Cap

1.5 Control Systems

The RSW is managed by a control system that respects the welding schedule, which is set to get the best possible spot weld and to provide the correct heat input into the the weld zone. However, the execution of the schedule can be compromised by unwanted variations of the process parameters caused by the evolution of the process in the melting zone and fluctuation of the energy source. Originally, the control system was an open loop, so it could not react to process parameter drift from the set values and, in some cases, there was a warning signal to inspect the weld. Instead the modern control systems are in a closed loop. Thus they use the result of the monitoring process, to contrast the fluctuation of process parameters. However, this happens with some delay with respect to the recognition of the dissimilarity among set and monitored values. Besides the compensation systems to contrast undesired variations, there are adaptive feedback control systems that use the technological information in these electrical signals to obtain the desired weld quality. This adaptive welding schedule provides acceptable welds over a wide range of welding parameters and possibly compensates critical welding parameter variations which can interfere with weld quality. In RSW, a classical approach is to fix the optimum combination of process parameter values, such as force, tip voltage and welding current, and to maintain them at the set values. It is important to specify that no welding machine can produce acceptable welds if the welding parameters are unacceptable such as high resistance due to a poor fit up of the work pieces, improper force, insufficient welding current available, improper welding program or unacceptable

welding electrode condition [3].

1.5.1 Welding Current and Constant Current Control

The welding current, which flows into the secondary circuit of the welder, is measured using a current transformer or air-core toroid employed in the primary circuit of the welding transformer. Comparing the sampled values to the tolerance limit for the set value it is possible to warn the operator if the current is out of tolerance and to reject the weld. A typical approach is to compensate the current to maintain it within a few percent of its preset value, this control logic is called Constant Current Control (CCC). The adjustments following the drift are not immediate. The effective delay can be less than three cycles of the line voltage. Constant current still requires the use of steppers to maintain current density with the mushrooming of electrodes. It is recommended only for single transformers (one gun welding at a time).

1.5.2 Electrode Tip Voltage and Constant Voltage Control

The electrode tip voltage is measured using copper wires attached to the the electrodes. If the welding current is not direct, the secondary voltage is read when the current is at its maximum when the inductive pick up is at minimum. A typical approach is to compensate the secondary voltage to maintain it within a few percent of its preset value. This control logic is called Constant Voltage Control (CVC). Despite the flattening or mushrooming of the electrode tips or the shunting of the welding current through closely spaced adjacent weld nuggets potentially providing an insufficient current density, this logic of control supplies more current to the lower impedance increasing the current density and the probability of a good weld. Although the magnitude of the current in the active welding area is not strictly controlled, many practical applications for this type of regulation have been found. It should be noted that the constant tip voltage is detrimental, perhaps resulting in severe over-current or under-current, when the thickness of the work pieces or the number of weld interfaces of the work pieces change and no manual heat adjustment is made.

1.5.3 Power and Constant Heat Control

The electrode tip voltage and the welding current signal are multiplied in real time to obtain the power signal, providing information on the heat input supplied to the weld. Obviously this signal is affected by the variation of the signals from which it is evaluated. A typical approach is to fix constant the power, this control logic is called Constant Heat Control (CHC). An historical approach in the heat input evaluation is the use of the Watt-second monitor, an instrument that will measure the actual energy input into the weld ($IVT = I^2RT$). The instrument integrates each signal separately which involves the time in the heat equation; then, it multiplies the results of the two integration and arrives at a numerical amount equal to the energy input to the weld. This measurement method is very effective in monitoring the consistency of the heat input to a weld, indeed charting these values for homogeneous spot welds with verifying their quality can be used to monitor weld quality. Instead the Millivolt-Second Monitor measures the relative energy input into the weld ($VT = IRT$) integrating the electrode tip voltage. This measurement method is a relative energy measurement rather than an actual energy (watt-seconds) method and it is referred to other spot welds produced in the same conditions. In the processing of identical welds the contact resistance between the two metals and between the electrode tips may change, while the weld current amplitude did not significantly change relatively to the impedance of the load, then the voltage among the electrodes' tips reflects any change in resistance. So the Millivolt-Second Monitor has a greater impact on monitoring the effect of resistance changes that may occur only between the electrodes because it does not utilise the I term. Charting these values for homogeneous spot welds with the verification can be used to monitor weld quality [3].

1.5.4 Dynamic Resistance

The values of the resistance of the secondary circuit of the resistance spot welder is obtained as the ratio between the secondary voltage and current sampled values by the use of Ohm's Law. These points can be used to create a curve which represents

the variation of resistance the dynamic resistance curve.

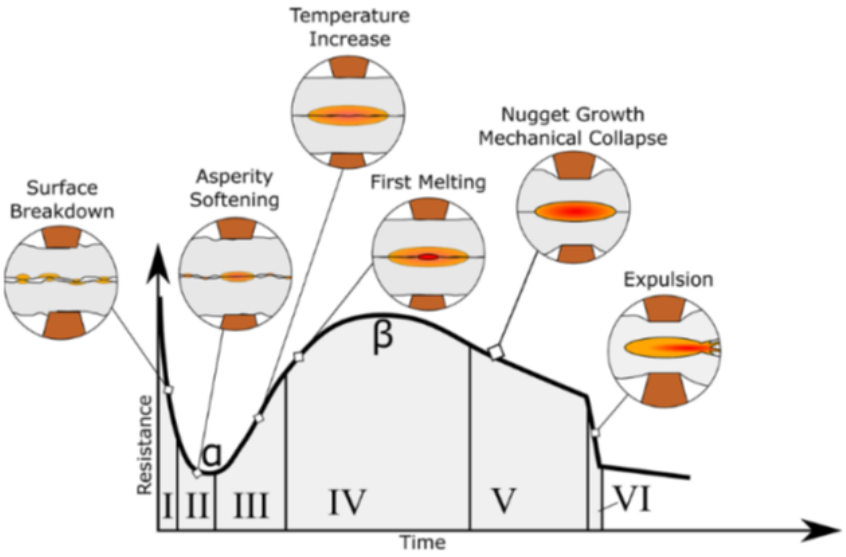

Figure 1.12: Dynamic Resistance Curve

According to literature [16], the typical shape of the dynamic resistance curve, represented in Figure 1.12 [2], can be interpreted referring to the physical change which occurs in the weld zone material throughout the course of the spot welding. This series of events, divided in five stages, offers a consistent interpretation of the shape of the dynamic resistance curves observed for spot welds made in the plain carbon AK material; however it can be generalised also to thin coated materials.

Stage I

The work pieces are brought into contact under the pressure provided by the electrode force. This creates areas of electrical contact at the points where asperities on the surface meet. Voltage is applied between the electrodes causing current to flow at the micro contact points to generate the heat necessary to melt the metal to form a weld nugget. The resistance between electrodes at the contact point is equal to the sum of the bulk resistance of the two work pieces, the contact resistance between the electrodes and the work pieces and the contact resistance between the work pieces. The initial contact resistance will be very high because of poor contact among metal sheets' surfaces and of presence of surface films, oxide layers or other contaminant

on the work pieces' surfaces. Indeed, these contaminants are essentially insulators. Therefore the initial generation of heat will be concentrated at the surfaces, especially at the interface between the work pieces. As the heat increases, the metal of the work piece will soften and surface contaminants will be burned away to form a more intimate contact at the welding interfaces, resulting in a very sharp drop in resistance. However due to surface irregularities, corrosion or poor contact, the initial electrical resistance of the welding load may be inconsistent and fluctuate in an undefined manner.

Stage II

After the breakdown of surface contaminants, metal-to-metal contact occurs. However, the surface resistance may be still relatively high due to the limited area for current flow provided by the asperity contacts. Heating then is concentrated at the work pieces interface, and temperature in this region and in the bulk material will increase. As heating progresses, the asperities soften and the contact area increases, thus causing resistance to decrease. At the same time, increasing temperature results in increasing resistivity, thus providing an opposite effect. The competition between these two mechanisms determines whether resistance is increasing or decreasing and thus determines the position of the minimum. Eventually, the increase in contact area will be overcome by the increasing temperature effect, and the total resistance will begin to rise.

Stage III

During this period, the increase in resistivity resulting from increasing temperature dominates the resistance curve. The end of stage III should correspond to local melting beginning to occur at the asperity contacts. The transition to stage IV is likely to occur near the inflection point in the curve $\frac{d^2R}{dt^2} = 0$.

Stage IV

Bulk temperature of work pieces continues to increase, thus causing resistivity and resistance to increase. But the heat being generated also causes additional melting to occur at the surfaces, increasing the size of the molten region and the crosssectional

area at the sheets interface available for current flow. This mechanism causes a resistance decrease. In addition, increased softening will result in some mechanical collapse, shortening the path for current flow and reducing the resistance. The peak is a consequence of the temperature beginning to stabilise, while nugget growth and mechanical collapse begin to dominate, and therefore resistance starts to decrease.

Stage V

Beyond the peak, the growth of the molten nugget and the mechanical collapse continue to cause resistance to decrease. So the size of the the weld nugget may be represented by the percent drop of the electrical resistance with respect to the peak of electrical resistance attained by the welding load. If the nugget grows to such a size that it can no longer be contained by the surrounding solid metal under the compressive electrode force, expulsion will occur. Variables expected to cause significant variation in the shape include current level, electrode force and material being welded. If the current is too low or the welding force is too high the resistance profile doesn't show the peak; the curves showing this flattening characterise spot welds with a reduced nugget diameter. The local maximum appears increasing the first and reducing the latter, however if this change is too enhanced the expulsion of molten metal can occur with a sharp drop in the resistance value. Besides if the welding current is higher, the maximum occurs earlier and the final value of the curve is lower.

The effect of the current level can be interpreted referring to the joule effect, so greater its intensity bigger the heating and the melting of metal which remarkably affect the dynamic Resistance trend. While the clamping force effect interpretation is more complex, when it is larger the contact resistance and the overall resistance are lower as the thermal power developed by joule effect, so the supply of the heat energy required to eliminate this surface is slower, furthermore the time when the first melting point appears is postponed. Initially there is an higher efficiency of heat energy contributing on nugget growth making the liquid nugget increasing faster, so the peak value is smaller, after a rapid development the molten nugget stabilises its size to a lower value. The time among the local maximum and the first melting point

is smaller. It is important to note that the dynamic resistance is lower when large electrode force is applied, because the latter promotes the intimate contact among the metal sheets causing the first drop in resistance of the welding area. Finally despite the expulsion of molten metal has been introduced with a low inadequate value of the clamping force, this can be caused also by a too high value. More in details the larger electrode force produce a higher indentation, this favours the drain if heat from the faying surface among the sheets to the environment in the vertical direction, reducing the nugget growth in the horizontal direction, this occurs fast when the heat generation and dissipation achieves an equilibrium [54].

1.5.5 Dynamic Resistance Adaptive Feedback Control System

Because the dynamic resistance curve can be interpreted in terms of nugget formation, it can be used to establish the welding schedule to determine the heat and the weld time to be used. After the collection of the profiles relative to acceptable manual weld as reference, an algorithm can determine what heat and "weld time" should be used to produce welds with the same resistance characteristics, or else good welds. Instead a Dynamic Resistance Adaptive Feedback Control System uses the dynamic resistance curve to adapt the welding schedule in real time on the basis of reference curves. The welding current can be terminated after the resistance signal has reached a value indicating an acceptable weld nugget has been formed, so the weld time is not constant. If the signal does not evolve as expected the welding current or the weld time can be increased. So this system set up to automatically adapt to the variables of many welding configurations, such as changes in the thickness of the welding stock, in the number of weld interfaces or in the line voltage. Typically, welding controls employing the resistance monitoring adaptive feedback principle include the following

- An adjustable length blanking period to ignore the undefined resistance excursions at the beginning of the welding period.
- Memorisation of the peak value of the electrical load resistance attained.

- Continuous monitoring of electrical resistance value of the welding load.
- An adjustable percent drop reference point to begin weld termination procedure when the monitored electrical resistance has decayed from the memorised peak to the selected value.
- An adjustable delay after the percent drop reference point has been reached.
- Asynchronous termination of the welding current at the end of the delay.
- An adjustable weld time limit to terminate the welding current when a weld has not been successfully made, due to an improper setup or welding parameters that are out of the range of the control.
- Adjustment of the ranges of the welding current and the tip voltage anticipated for a particular welding machine.

There are many control algorithms depending on the manufacturerssupplier. The R Drop control algorithm adjust the weld time until the programmed drop from the peak of the resistance curve for the given weld is reached. However if this drop does not occur in a window, if it occurs after the superior limit for consecutive welds the heat input is increased to bring back the R drop in the window, if it occurs before the inferior limit the control acts oppositely. For galvanised metals, the low and high limits are set narrow mainly because of their greater sensitivity to currents rather than time. The Forcing the Curve algorithm increase the current if its slope is low adapting the weld schedule to many thicknesses. The Expulsion Control evaluates the the weld time to avoid the expulsion of molten metal. It is used for mild steel and some coated metals, while it is not recommended for metals which does not produce the classical resistance curve like the low resistive ones (brass, copper and aluminium). It is important to observe that the diffusion of zinc into the electrodes causes an improper resistance drop to occur [3].

1.5.6 Displacement

The thickness of the welding zone changes during the RSW, so it can be measured as electrode displacement using a displacement sensor positioned on the moving electrode. A distance curve can be generated by the sampled values of the electrode displacement. This curve is influenced by the physical change which occurs in the weld zone material throughout the course of the spot welding. Its typical shape is shown in Figure 1.13, where it is divided into two stages for its interpretation. During Stage 1 the work pieces expand during heating. In Stage 2, at the point the nugget begins to fully form, the work pieces compress provided the weld gun has the required follow-up characteristics. This method has provided excellent results for expulsion detecting, indeed this phenomenon correspond to a sudden drop in the curve. For displacement measuring the classical potentiometer can be affected by wire interference due to the noise from the magnetic field, so, in some cases, the displacement is measure using a digital optical encoder. It is difficult to install and maintain a sensor in the secondary of high production machines. However its fundamental limitation is the lack of robustness [3].

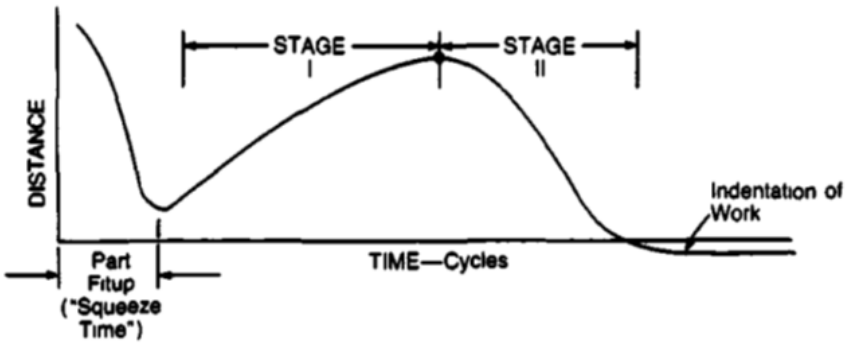

Figure 1.13: Displacement Curve

1.6 Quality

If the welding parameters are improperly set, they will not guarantee the quality of the spot weld. This setting is referred to the material and geometry of the work

pieces and of the electrodes, these factors are affect by a great variability [6].

The metal sheets to be welded are made of a specific material and with geometrical specification, however they can exhibit some differences compared to the nominal condition in a range of tolerance established with the supplier. For this reason the initial value of the electrical resistance of the weld zone is affected by a variability due to the variability of its components. The great variability of the contact resistances is due to the variability of characteristics of sheet surface roughness, thickness of the coating, hardness of the coating or of the nude metal, presence of contaminants.

The variability of the resistance of the work pieces is due to the variability of their characteristics, i.e., thickness and resistivity.

For the resulting value of the electrical resistance of the welding zone there is a new welding lobe different from the one evaluated for the application. In an effort to guarantee weld quality, welding parameters are sometimes set close to a splash condition. This is done so that using the same welding parameters small changes in electrical resistance are unlikely to give small welds, but rather lead to splash, or else the expulsion of molten metal from the weld nugget at the same welding parameters. This latter condition is easy to be detected also by visual inspection without using particular quality tests [5].

1.6.1 Electrodes

As previously stated the electrodes are made of a specific material and geometry, however they can exhibit some differences compared to the nominal condition in a range of tolerance established with the supplier. Besides the electrode material and geometry is subject to evolution associated with wear during the production of subsequent spot welds. There are two type of electrode wear

- mechanical wear, the alteration of the electrode shape, especially the nose with increase of its diameter, because of the deformation under the high electrode force facilitated by the softening of the electrode at high temperatures and counteracted by the cooling of the electrode;

- chemical wear, the alteration of the electrode material because of the diffusion of the material of the sheets in the electrode, diffusion pollution, this is enhanced in the automotive industry where the zinc coating of the sheets and the copper of the electrodes are affine from a chemical point of view.

The size and the geometry of the electrodes affect the current density distribution and the pressure distribution, so the result of the process. With mechanical wear the current density reduces with the same current and the pressure reduces with the same electrode force. This justifies the choice to set welding parameters close to the splash condition or else the right limit of the welding lobe, indeed it assures the quality of the weld in a certain range of degradation before the reaching of the left border.

Referring to the automotive industry, where copper alloy electrodes are used in RSW of zinc coated steel sheets, it is important to underline that the electrode is of "pure copper" just in the production of the first weld. In the following spot welds the electrode nose becomes a compound of copper (Cu), zinc (Zn), oxides, oils and powder. This cause the so called "inversion of contact resistances". In the spot welding of uncoated metal sheets the contact resistance at the interface among the work pieces is bigger than that among the electrode and the work piece, so the temperature diagram, shown in Figure 1.14, reach higher temperatures at the faying surfaces because the heating is proportional to resistance according to the Joule effect law. Besides the contact resistances at the interface among the electrode and the metal sheets have to be lower as possible to avoid melting of the metals at the interface and sticking of the electrode tips to the work piece. To avoid this the electrode material has to be highly electrical and thermal conductive and cooled by a proper cooling system.

Instead when welding coated sheets, as in the modern automotive industry, the temperature diagram has this shape just in the production of the first spot weld by the electrodes. After with electrode wear the contact resistances at the interface among the electrode and the work piece increase and become bigger than the contact resistance at the interface among the sheets. So referring to these different electrical

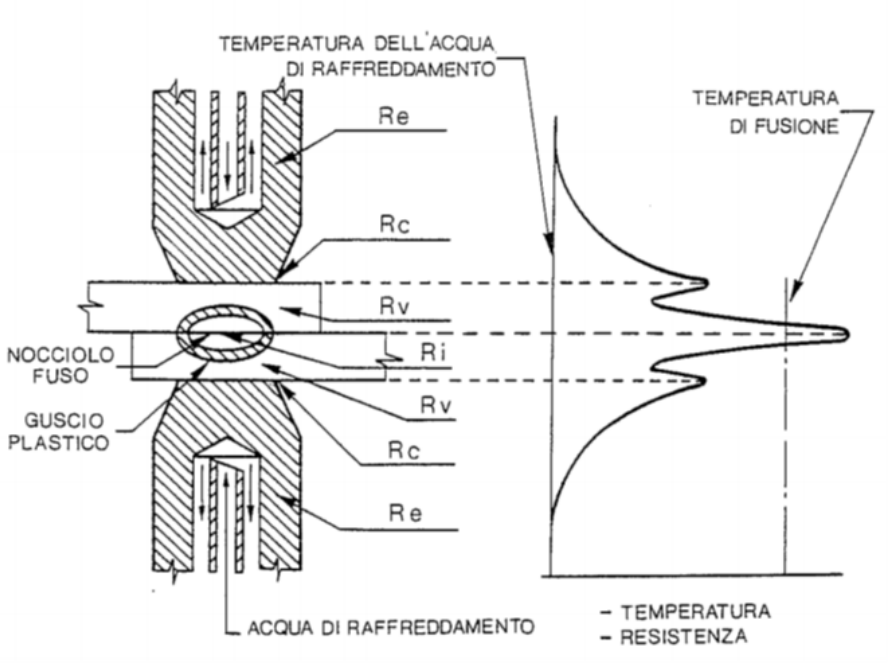

Figure 1.14: Contact Resistances

characteristics the temperature diagram has the shape in Figure 1.15.

In this new condition the diagram is inverted compared to necessary conditions for nugget formation and the welding among among the electrodes and the work pieces are probable. To avoid this the electrode force and the cooling system efficiency have to be increased, besides the welding time has to be increased in order to make evolve the resistances to the the proper condition for nugget formation. The welding current has not to be increased, this increase the electrode sticking and external splash.

The wear of the electrodes grows with the number of spot welds produced, however it has to be limited at an acceptable level to stabilise the process with the electrode dressing. The dressing does restore the original shape of the electrode cleaning it from the metal crust formed on it, however the material of the electrode nose is not restored to the original chemical composition, so it is affected by diffusion pollution. The dresser act on the lateral tapered surface and not on the flat surface bringing back the diameter to the original condition.

The constant diameter of the electrode nose stabilises the process and guarantees the quality of the spot weld, ensuring a constant current density and pressure, so the

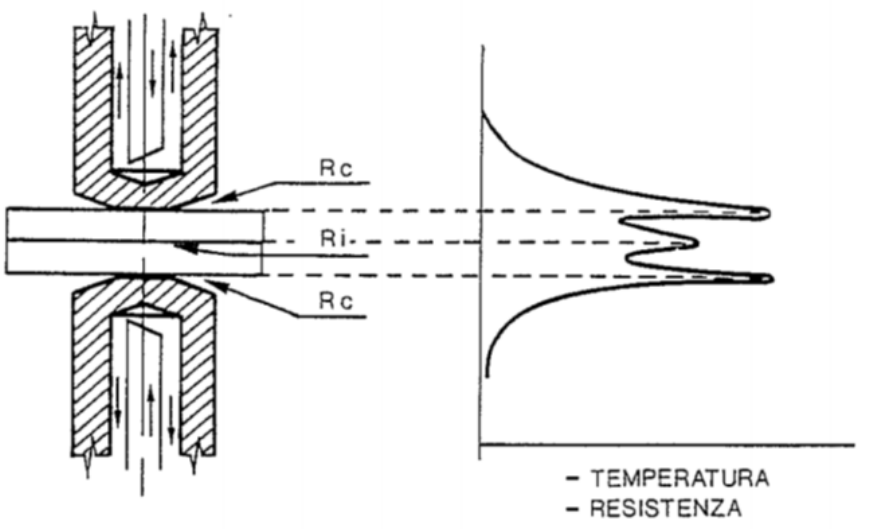

Figure 1.15: Contact Resistances Inversion

best practice is to dress the couple of electrodes after every cycle of work. However this strategy cannot always be used, in these cases the diameter degradation curve has to be evaluated in order to fix the proper dressing frequency. This curve reports the increase of the electrode diameter in function of the spot welds produced from the last to the next dressing, as shown in Figure 1.16.

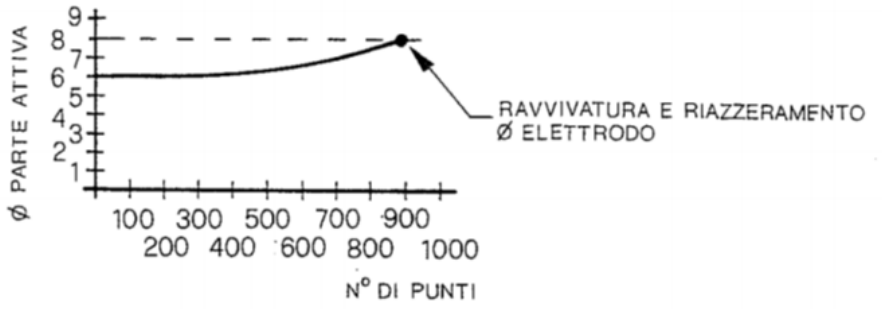

Figure 1.16: Electrode Wear

In the industrial practice two thresholds are set for the dressing

- A Threshold, the number of spot welds executed from a dressing to the next one; when this is reached the welder warn the operator but the work cycle is not arrested;
- B Threshold, the number of the executable spot welds after the previous

warning; when this is reached the welder arrest the work cycle.

The operator has to dress the electrodes and reset the spot weld counter as soon as possible, so after the work cycle in execution is finished without exceeding the threshold. The B Threshold has to be set so that the welder arrest does not interrupt the work cycle.

1.6.2 Positioning

The previous observations assume that the weld zone is produced on plain sheets and it is surrounded by a sufficient portion of metal sheets, however in the industrial context it is necessary to obtain spot welds also on curve sheets and on the edge of the sheets. In this conditions the electrode force application is difficult, influencing the contact resistance, therefore it is a factor which increases the variability of the weld structure's electrical resistance. To facilitate the positioning of the electrodes on curve sheets the electrode shape is dependent on the application. In edge welds the electrode force does not contain effectively the molten metal facilitating the occurrence of expulsion. The three images of Figure 1.17 shows the spot welding near the edge of a sheet, that of both sheets and where the sheets are curved.

Another implied hypothesis is that the electrodes are perfectly normal to the surface of the metal sheets and collinear, in reality they can be not aligned andor not normal to the surface because of erroneous positioning of the electrodes by the gun. This error can be caused by the electrodes' holders not being collinear or not normal to the surface, by the wrong programmed gun movement or by the sheets curvature despite the proper electrode shape, whose interaction with the electrodes during the squeezing can cause their deformation. This affects the pressure and the contact resistance, influencing the current density and the spot weld quality. The nonfunctional gun can cause different defects, shown in Figure 1.18, underlining the drift of the spot welding from the ideal condition, such as

- the "Z" deformation of the work pieces because of electrodes collinear, but not normal (the sheets are rotated from their original plane);

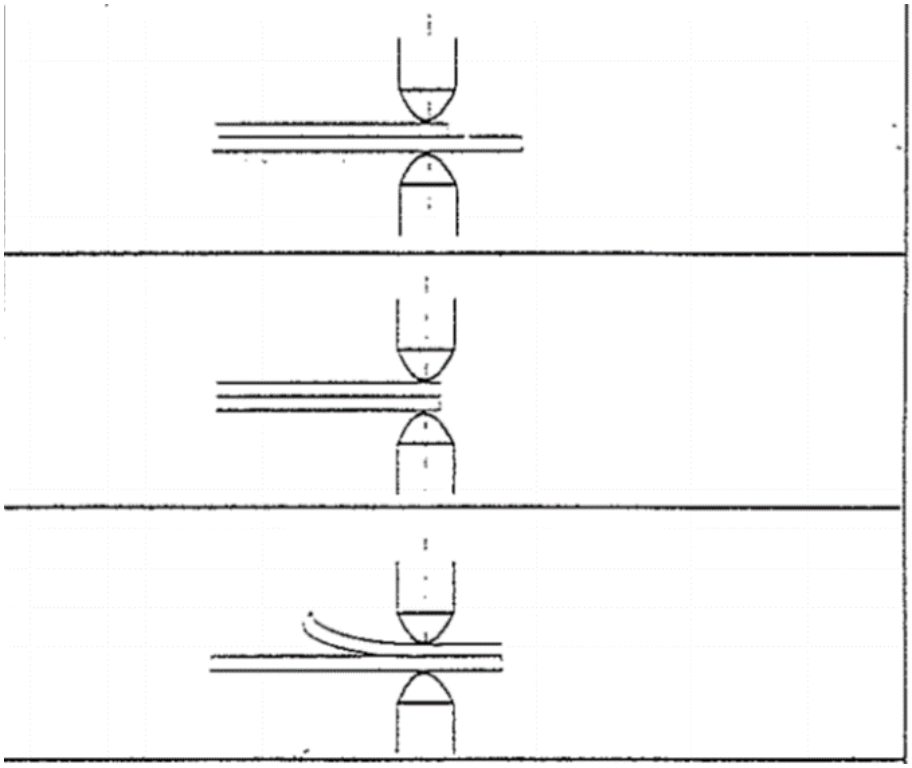

Figure 1.17: Edge Weld and Curvature

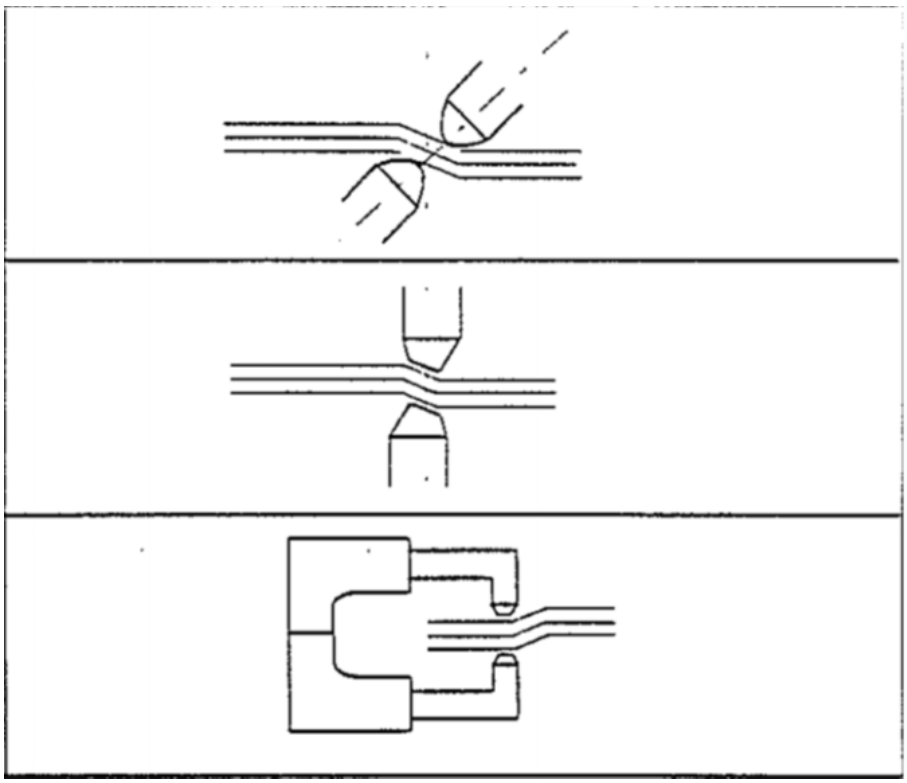

Figure 1.18: Z Deformation

- the "Z" deformation of the spot because of electrode normal but not collinear (the weld interface is out of the plane of the sheet metal);
- the "Z" deformation of the sheet towards the outside of the gun because of improper movement (the sheets are displaced from their original plane).

It is important to observe that the curvature of the sheet surface can be a characteristics of the nominal geometry or an effect of a previous spot weld which cause the spreading apart; in this case the quilting of the sheets can occur, as shown in Figure 1.19. This deformation of the weld area is associated to the thinning of the metal, so these spot welds could have an excessive indentation and to be weaker.

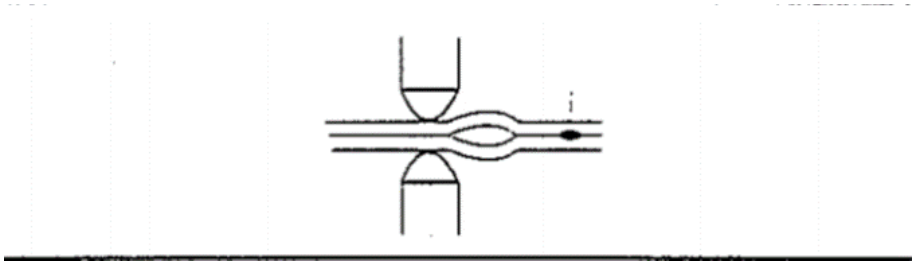

Figure 1.19: Quilting

1.6.3 Shunting

In RSW the shunting is the diversion of the welding current from the welding zone to a nearby existing weld. It is difficult to avoid in industrial applications because multiple welds are commonly arranged next to each other in the same region of a part for structural and handling purposes. However it is also experienced in the repairing applications when a broken spot weld is substituted with a new one. Therefore, one or more premade neighbouring welds often exist when making a new weld. The existing welds may divert the electric current from the new weld, indeed they are characterised by a lower resistance than the welding zone because of their metal continuity. As the applied electric current is shared by the weld to be made (the shunted weld) and the existing weld (the shunt weld), the heat generated in the shunted weld may not be sufficient for melting or for growing the weld nugget to the designated size. However during the process the shunting effect is reduced by the

melting of the metal in the welding zone, which reduces the resistance of the shunted weld. The distribution of welding current in shunting is illustrated in Figure 1.20 for two adjacent spot welds.

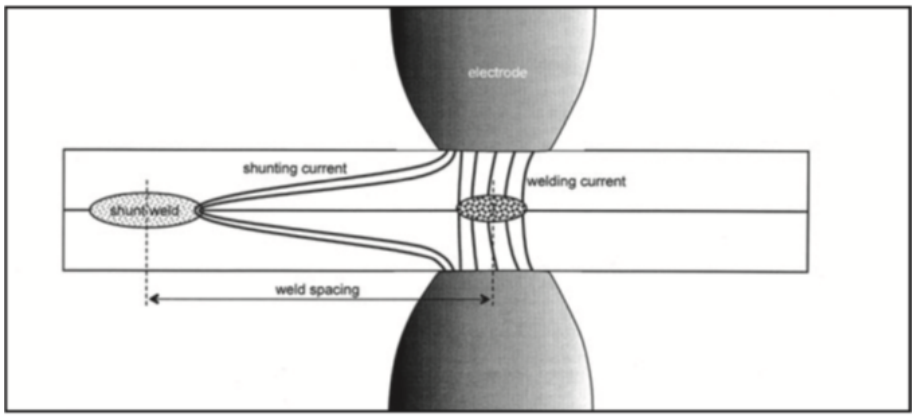

Figure 1.20: Shunting

The proportions of the electric current through the shunting and welding paths are determined by the relative values of electric resistances along these paths. These resistances can be categorised as follows

- Shunting path, composed by
  - R_C the contact resistances at the electrode-sheet interfaces at both the top and bottom surfaces;
  - R_{BS} the bulk resistance along the shunting path through the shunt weld.
- Welding path, composed by:
  - R_C contact resistances at the electrode-sheet interfaces at both the top and bottom surfaces;
  - R_{BW} bulk resistances of the top and bottom sheets;
  - R_{CW} the contact resistance at the faying interface where the shunted weld has to be made.

The contact resistances at the electrode contact can be assumed identical for these two paths, so it is indicated by R_C. The electric circuit of the welding setup can be

simplified as shown in Figure 1.21. In this diagram the total applied electric current, I, is split into a welding current and a shunting current, I_W and I_S, respectively. These two currents, and therefore the amounts of heat generated in the shunting and welding path by joule effect, are determined by the relative values of the resistances along the two paths. However, these resistances along these two paths are not constant during the process and evolve with increase of temperature and metallurgical changes. With melting the contact area at the faying interface along the welding path and the contact resistance R_{CW} disappear. In this moment the resistance in the welding path, strongly dominated by the contact resistance, is reduced and the resistance ratio of the two paths changes drastically. It is too complex studying the shunting accounting the evolution of these resistances, so their values at the beginning of the welding are used to understand this phenomenon. It is fundamental to stress that this choice is made possible by the fact that this welding is strongly influenced by these values determining the electrical current distribution and the heat generation for the whole weld time.

Both the resistance of the shunting path and of the welding one is positively related to the resistivity of the metal of the sheets to be welded. The resistivity of a material, with fixed chemical composition, is a function of temperature, so different metals exhibit different increases of resistivity with temperature. However the resistivity relation with temperature is affected by phase transformation in the solid state, increase of the crystal size and by fusion with temperature increase. Despite this function is not known for most of the metals beyond their melting points, it is known it is characterised by a sudden rise at the melting point. It is important to stress that generally in shunting the shunting path does not experience melting, so the resistance of the shunting path is influenced by the resistivity of the parent metal, that of the heat affected zone and that of the weld nugget. Instead, as previously observed in other sections, the resistance of the welding path is influenced also by the the resistivity of the molten metal, despite this influence the resistance does not simply increase with the increase of the volume of melted metal. From an industrial point of view it is important to note that aluminium and magnesium have significantly lower resistivity than iron from the room temperature to the melting point, so the

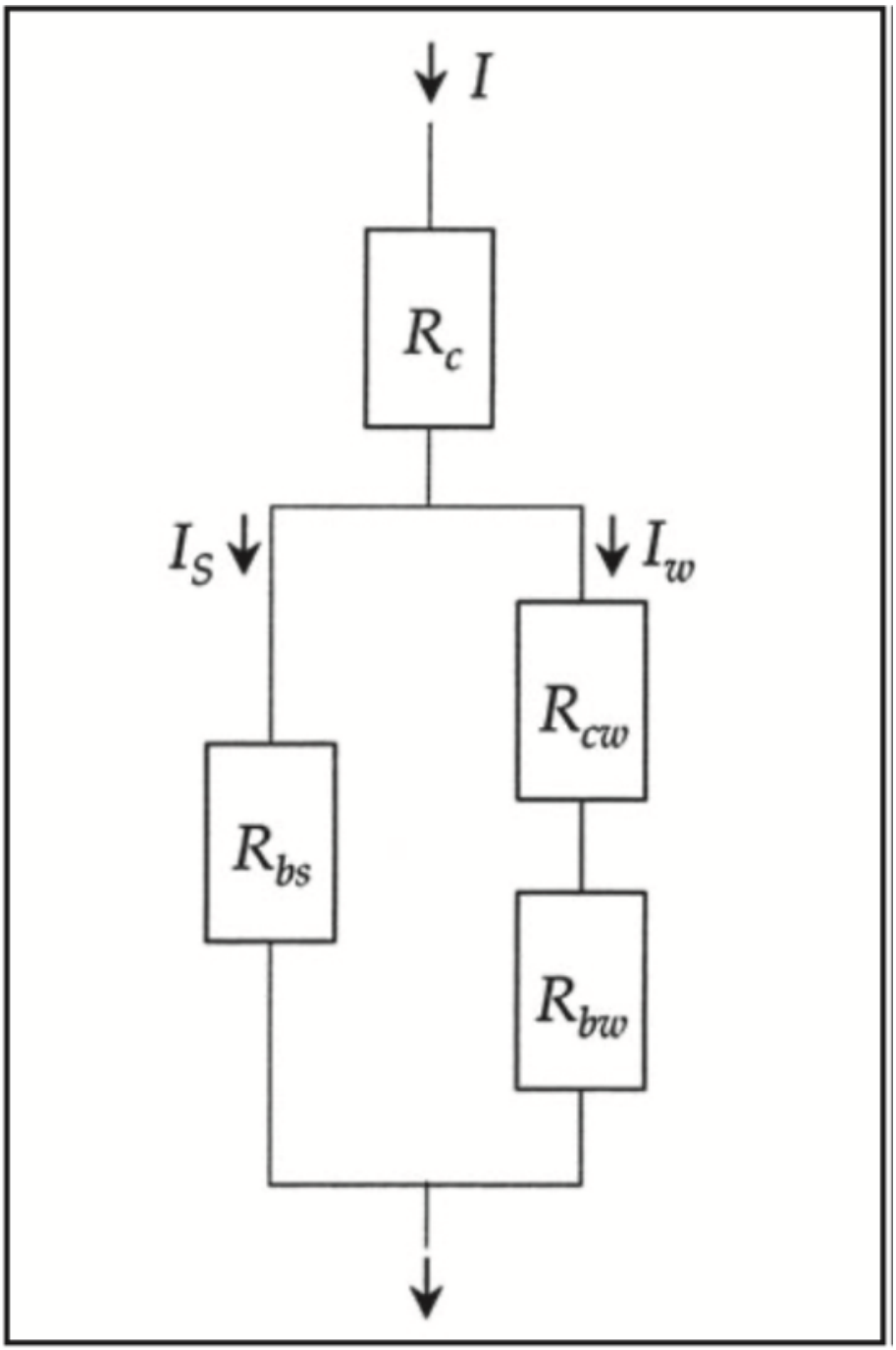

Figure 1.21: Shunting Resistance Scheme

spot welding of aluminium and magnesium sheets are more subject to shunting.

It is important to specify that the resistance of the shunting path is influenced by the extension of the shunting path in the parent metal, strictly correlated to the distance among the shunted weld and the shunt one, and the size of the shunt weld, referring to nugget and HAZ's sizes. It is clear that the geometry and material properties of the work pieces have a strong influence on shunting.

The resistance of the welding path is composed by the contact resistance, so it is influenced by the electrode force and by the surface condition of the faying surfaces of the weld, so asperities and surface contaminants such as oxides and greases. While some contaminants like zinc, used as coating for corrosion protection, significantly reduce the contact resistance, others increase it. From an industrial point of view hot-dipped galvanised (HDG) steels possess lower contact resistance than galvannealed sheets. This resistance will disappear gradually with the increase of temperature at the interface of the metal sheets because of asperities deformation under the electrode force and molten metal distribution among the asperities. So the contact resistance at the sheet faying interface influence the resistance of the welding path and the shunting mostly at the beginning of the process, however it controls the heat generation in the metal also after its disappearing dictating the resistivity distribution. The characterisation of these resistances is fundamental for understanding the influence of the process parameters on the shunting phenomenon, such as: welding current, welding time and electrode force.

Welding current and welding time does not influence the proportion of the shunted current on the welding current, but oppose the reduction of the heat generation in the weld zone caused by shunting. It is possible to increase the portion of the current flowing through the welding path increasing the welding current, so the heating of the metal at the interface of the metal sheets is raised. Also the growth of the weld time boost the heating of the material and homogenise the distribution of electrical resistivity in the sheet stack-up.

Electrode force directly influence the contact resistance at the faying interface along the welding path, indeed a large electrode force creates a larger contact area among

the sheets reducing the contact resistance. However this relation is affected by the mechanical behaviour of the metal sheets under the electrode force, dependent on the yield strength of the metal and the sheet thickness. The contact among the parts to be welded is reached by the deformation of the asperities, so a bigger yield strength andor sheet thickness reduce this deformation and increase the resistance of the welding path. Besides the influence of the electrode force on the shunting effect can be reversed by the sheet thickness, indeed a large force can reduce the gap among thin sheets characterised by a lower stiffness, proportional to the sheet thickness, reducing the shunting path, so its resistance.

Other geometric characteristics influencing the shunting are the electrode geometry (dimensions and shape), and the electrode alignment (axial and angular misalignment) which regulate the pressure application on the sheets, so the electrode force.

Shunting is affected by many variables whose effects depend on intensive interactions among them, therefore the effect of each factor cannot be studied simultaneously, but it has to be studied fixing the other factors. So this phenomenon is studied by mean of experimental studies which have to be projected opportunely, designing an experimental matrix where for each variable many values are selected. The effect of every factor is debated based on empirical observations which are accepted or rejected using statistical analyses combination of factors. However the conclusions have to be taken cautiously, taking into account the experimental conditions, because of the previously cited limitation. Another approach of study is the analytical analysis or numerical modelling, however these are difficult because of the metal resistivity function of temperature is not known after melting, so they are conducted assuming idealised behaviour for the material. For this reason the shunting prevision of these models are of qualitative nature and the experimental studies are preferred in the shunting study.

Often the the geometry and the material of the work pieces are fixed by the request of a project, so according to these the process parameters are established and the only factor to reduce shunting is the weld spacing, indeed increasing weld pitch the resistance for the shunt path grows. Welding a metal with large bulk resistivity, as

steel, requires a small weld spacing to avoid the shunting effect than the welding aluminium alloys.

The shunted welds are generally smaller in size than shunt welds free from shunting, however the shunted weld having another shunted weld for shunt weld is less affected by shunting, indeed this shunt weld has a reduced nugget size which increase the resistance of the shunting path and the existence of previous welds increase the contact area reducing the resistance of the welding path. However the first spot welds increases the the sheet separation and reduce the contact area for the subsequent spot weld, so the previous observation is true for a proper number of spot welds. The shunted welds are also characterised by an amplified burn mark, indeed the total current flows through the contact resistance at the electrode-sheet interfaces an can be increased by the reduction of the total resistance because of shunting according to the system control of the welder. This feature can be observed in visual inspections of spot welds. The shunted welds' Heat Affected Zone is asymmetric because of the asymmetry of the current density field [10] caused by the shunting, this can be observed by micrograph on the transverse section of the weld [32] [46], as in Figure 1.22

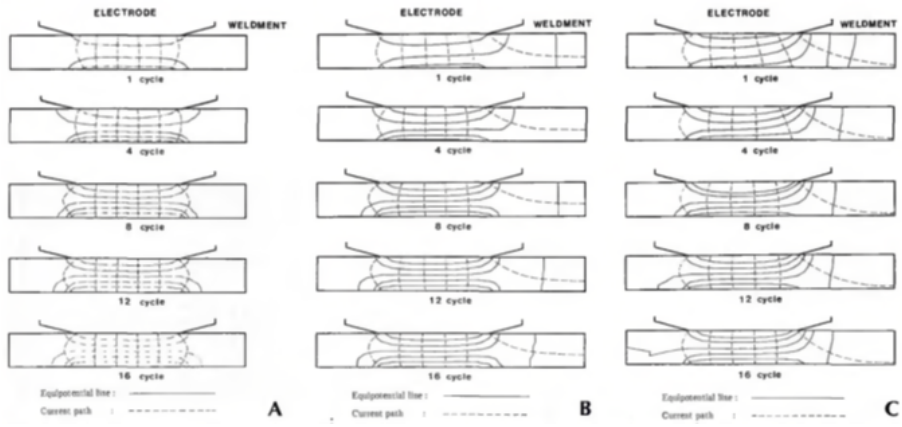

Figure 1.22: Shunting Current Density

1.7 Defects and Assessment

The spot welds produced can be classified in seven classes according to the defects.

1.7.1 Good Weld

A Good Spot guarantees the structural integrity of the welded assembly, its penetration (thickness) is adequate and the nugget, which joins the two metal sheets, has proper dimensions. Generally the joint thickness of a Good Spot is 70-90% of parent metal thickness, indeed the spot weld indentation is a technological signature. It does not contain internal flaws, such as: blowholes, porosities and cracks. The heat input and the the force are adequate [7] [35]. It is shown in Figure 1.23 [52].

1.7.2 No Weld

A No Weld spot is characterised by the separation of the metal sheets. Generally this defect is caused by an inadequate heat input which has not melted the metal, so it is characterised by the absence of the weld nugget. This low heat input can be associated to a high electrode force. Sometimes despite the melting of a proper volume of metal the sheets are still separated after the process because the electrodes have been removed before the solidification for a low hold time [7] [35]. It is shown in Figure 1.24 [6].

1.7.3 Stick Weld

A Stick Weld is similar to a No Weld, indeed it characterised by the absence of the weld nugget, however the metal sheets are held together by localised fusion at the welding interface. The thickness of the joint is approximately equal to the parent metal thickness because no significant parent metal melting with subsequent squeezing has occurred, however the thickness of the spot is lower than the total parent metal thickness [52]. The heat input is inadequate to create a nugget, but sufficient to melt the contact points at the interface. If the work pieces are coated, the coating can melt and freeze soldering the part together. The resulting bond among the sheets fails under low loads [7] [35]. It is shown in Figure 1.25 [52].

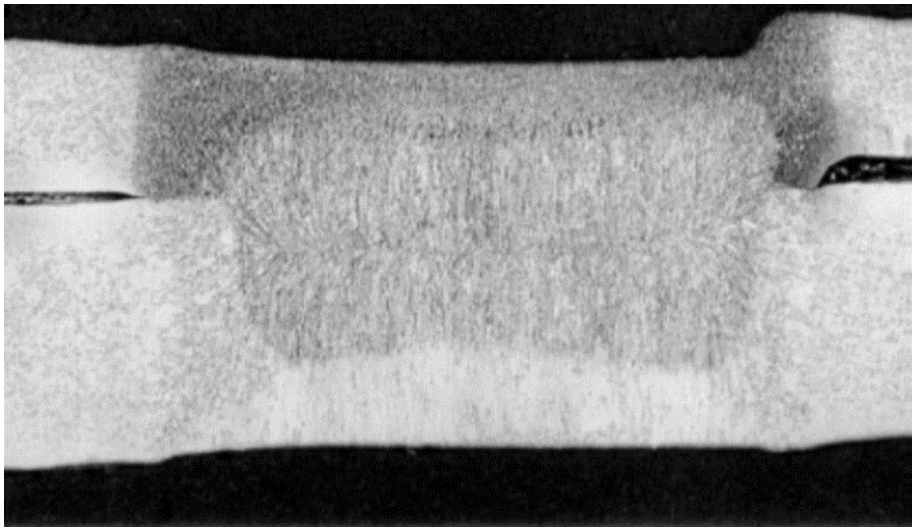

Figure 1.23: Good Weld

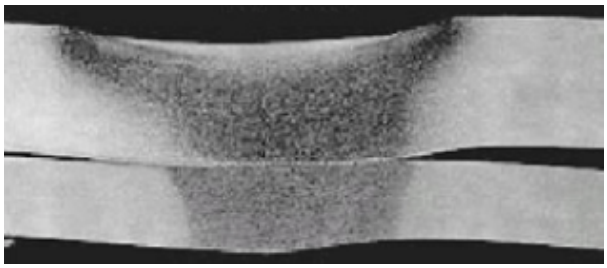

Figure 1.24: No Weld

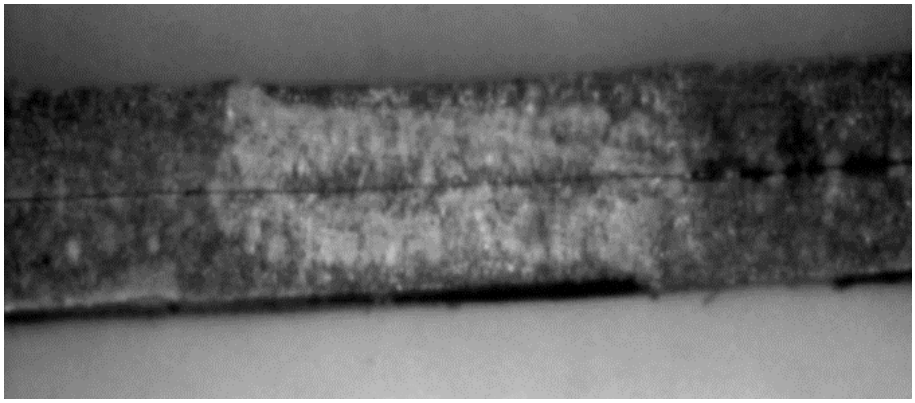

Figure 1.25: Stick Weld

1.7.4 Small nugget

A Small Nugget is characterised by a smaller weld nugget and joint thickness than in a Good Spot. The process has not melted enough metal at the interface of the metal sheets, so the nugget size is small in radial and transverse dimensions, actually its diameter is smaller than the nominal diameter of the spot weld and a circular ring around the center of the weld remains unwelded. The thickness of the joint is not very dissimilar from the parent metal thickness because little melting and squeezing have occurred [7] [35]. It is shown in Figure 1.26 [52].

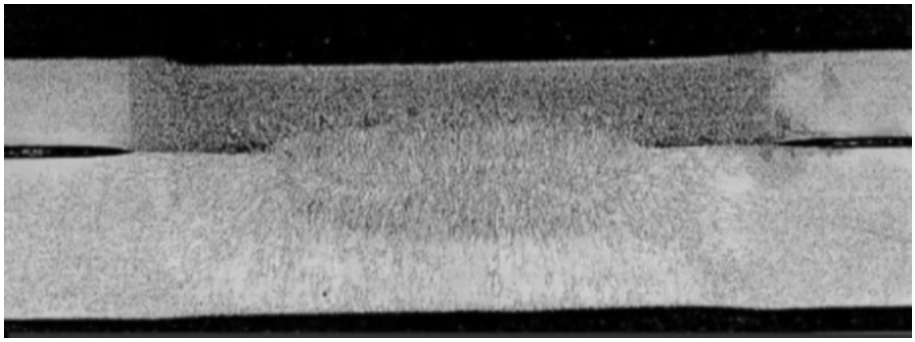

Figure 1.26: Small Nugget

1.7.5 Thin Spot

A Thin Spot is apparently similar to a Good Spot, but it is characterised by a lower total thickness of the weld nugget [34] and of the joint. The process has melted enough metal at the interface of the metal sheets to get a nugget with proper diameter, however it has not a proper penetration. The joint thickness, lower than a minimum percentage value of parent metal thickness, does not provide the necessary strength. In other words the indentation, an essential part of resistance welding, is excessive. It is due to an excessive electrode force related to the softening of the metal at high temperatures [7] [35].

1.7.6 Flaw in Spot

A Flaw in Spot is apparently similar to a Good Spot, but the weld nugget is characterised by internal defects such as: gas porosity, blowholes, cracks, voids, contraction cavities, non metallic inclusions. The formation of shrink holes or gas pockets with uncoated metal sheets is mostly caused by contamination of the sheet surface, for example with oil. In the case of zinc coated metal sheets the shrinkage cavitation is very likely with the hollowing of the electrodes, as the zinc deposit is burning, gas pockets or blowholes develop under high pressure and expand as soon as the sheet metal melts at the weld spot. Due to the hollow shape of the electrodes, the ring weld is formed at the outer edge and grows towards the inside. This means that the gas pocket is forced inward and finally forms the shrink hole or piping within the ring and mostly in the middle of the total thickness because this is where the pressure equilibrium is during the liquid phase [7]. Heavy interface expulsion, caused by excessive heat input and low electrode force, can generate cavities at the border of the nugget with the space among the sheets. Surface expulsion of molten metal can create cracks [35]. It is shown in Figure 1.27 [52].

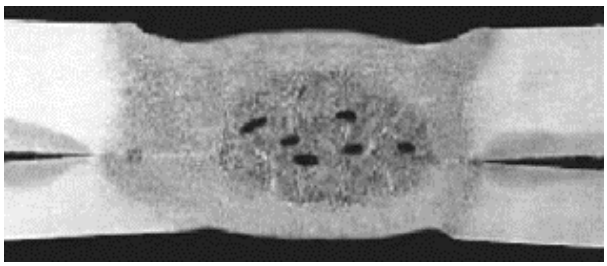

Figure 1.27: Internal Flaw

1.7.7 Burnt Spot

A Burnt Spot is characterised by a smaller joint thickness and a bigger weld nugget in all dimensions than in a Good Spot, indeed it extends to the surface of the joint. The process has melted too metal at the interface of the metal sheets, so the molten metal expulsion is very probable, therefore the nugget size is bigger in radial and transverse dimensions, but the joint thickness is lower because of metal expulsion.

This weld is brittle and often fracture at the interface of the sheets under a load with little or no stretching or yielding. It is fundamental to note that metal expulsion forms spikes of solidified molten metal, the so called whiskers or burrs, these are symptoms of the splash. When this last is to sever so much material is expelled that a hole is left that passes completely through the weld creating a Burn Through weld [7] [35]. It is shown in Figure 1.28 [52].

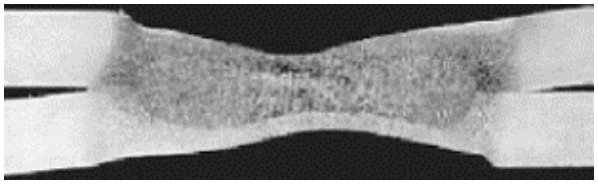

Figure 1.28: Burnt Spot

Chapter 2

Functional Data Analysis and Unsupervised Learning

2.1 Functional Data Analysis

Functional data analysis (FDA) deals with the analysis and theory of data that are in the form of functions, images and shapes, or more general objects. The atom of functional data is a function, where for each subject in a random sample one or several functions are recorded. Functional data are intrinsically infinite dimensional. This poses challenges both for theory and computation. These challenges vary based on how the functional data are sampled. On the other hand, the infinite dimensional structure of the data is a rich source of information, which brings many opportunities for research and data analysis [47].

The aims of functional data analysis are

- representing the data in ways that aid further analysis;
- displaying the data;
- studying sources of pattern and variation among the data;
- explaining variation in an outcome or dependent variable, by using input or

independent variable information;

- comparing two or more sets of data with respect to certain types of variation, where two sets of data can contain different sets of replicates of the same functions, or different functions for a common set of replicates.

The first task in a functional data analysis is to represent the data. Assuming that a functional datum for replication i arrives as a set of n discrete values $y_{i1}, ..., y_{in}$. The task is to convert these values into a function x_i with values $x_i(t)$. Interpolation is used when the discrete values are assumed to be errorless, while smoothing is involved if there is some observational error.

2.1.1 From scalar data to functional data

The basic idea is to think of observed data functions as single entities. In practice, functional data are usually observed and recorded discretely as n pairs (t_j, y_j), and y_j is a snapshot of the function at time t_j, possibly blurred by measurement error. From these observed data the existence of a function x arises. Also, we want the function to be smooth, so that a pair of adjacent data values, y_j and y_{j+1} are necessarily linked together to some extent and unlikely to be too different from each other. By smooth, we usually mean that function x possesses one or more derivatives, which we indicate by Dx, D^2x, and so on, so that $D^m x$ refers to the derivative of order m, and $D^m x(t)$ is the value of that derivative at argument t. The observed data, however, may not be at all smooth due to the presence of noise or measurement error. In general, we are concerned with a collection or sample of functional data, rather than just a single function x. Specifically, the record or observation of the function x_i might consist of n_i pairs (t_{ij}, y_{ij}), $j = 1, \ldots, n_i$. The argument values t_{ij} may be the same for each record or vary from record to record. It may be that the interval $\mathcal{T}$ over which data are collected also varies from record to record. Sometimes time t is considered cyclically, for instance when t is the time of year. This means that the functions satisfy periodic boundary conditions, where the function x at the beginning of the interval $\mathcal{T}$ picks up smoothly from the values of x at the end. Data for functions which do not naturally wrap around in this way are called non-periodic.

Smoothness is a property of the latent function x, and may not be at all obvious in the raw data vector $y = (y_1, \ldots, y_n)$ owing to the presence of observational error or noise. We express this in notation as

$$y_j = x(t_j) + \epsilon_j \tag{2.1}$$

where the noise ϵ_j contributes a roughness to the raw data. It is possible to express the "signal plus noise" model with vector notation as

$$y = x(t) + e \tag{2.2}$$

where y, $x(t)$, t and e are all column vectors of length n. The variance-covariance matrix for the vector of observed values y is equal to the variance-covariance matrix for the corresponding vector ϵ of residual values since the values $x(t_j)$ are here considered fixed effects with variance 0. Let Σ_e be our notation for residual variance-covariance matrix, which expresses how the residuals vary over repeated samples that are identical except for noise or error variation.

The sampling rate or resolution of the raw data is essentially a local property of the data and can be described as the density of the argument values t_j relative to the amount of curvature in the data, rather than simply the number n of argument values. The curvature of a function x at argument t is usually measured by the size of the second derivative, as reflected in either $|D^2x(t)|$ or $[D^2x(t)]^2$. Where curvature is high, it is essential to have enough points to estimate the function effectively. What is enough? This depends on the amount of error ϵ_j ; when the error level is small and the curvature is mild, it is acceptable a low sampling rate.

2.1.2 Representing functional data by basis functions

A basis function system is a set of known functions Φ_k that are mathematically independent of each other and that have the property that we can approximate arbitrarily well any function by taking a weighted sum or linear combination of a sufficiently large number K of these functions. The most familiar basis function

system is the collection of monomials that are used to construct power series,

$$1, t, t^2, t^3, ..., t^k, ...$$

Right behind the power series there is the Fourier series system,

$$1, sin(\omega t), cos(\omega t), sin(2\omega t), cos(2\omega t), sin(3\omega t), cos(3\omega t), ..., sin(k\omega t), cos(k\omega t), ...$$

Basis function procedures represent a function x by a linear expansion

$$x(t) = \sum_{k=1}^{K} c_k \Phi_k(t) \tag{2.3}$$

in terms of K known basis functions Φ_k. By letting c indicate the vector of length K of the coefficients c_k and Φ as the functional vector whose elements are the basis functions Φ_k, we can also express the linear expansion in matrix notation as

$$x = c'\Phi = \Phi' c \tag{2.4}$$

In effect, basis expansion methods represent the potentially infinite-dimensional world of functions within the finite-dimensional framework of vectors like c. The dimension of the expansion is therefore K. It would be a mistake, though, to conclude that functional data analysis in this case simply reduces to multivariate data analysis; a great deal also depends on how the basis system, Φ, is chosen. When $K = n$ an exact representation or interpolation is achieved, in the sense that we can choose the coefficients c_k to yield $x(t_j) = y_j$ for each j. Therefore, the degree to which the data y_j are smoothed is determined by the number K of basis functions. Consequently, a basis system is not defined by a fixed number K of parameters, but K is itself a parameter that has to be chosen according to the characteristics of the data. Ideally, basis functions should have similar features to those known to belong to the functions being estimated. So it is easier to achieve a satisfactory approximation using a comparatively small number K of basis functions. The smaller K, the better the basis functions reflect certain characteristics of the data

- the more degrees of freedom we have to test hypotheses and compute accurate confidence intervals,
- the less computation is required,
- the more likely it is that the coefficients themselves can become interesting descriptors of the data from a substantive perspective.

Consequently, there is no such thing as a universally good basis. The choice of basis is particularly important for a derivative estimate

$$D\hat{x}(t) = \sum_{k=1}^{K} \hat{c}_k D\Phi_k(t) = \hat{c}' D\Phi(t) \tag{2.5}$$

Bases that work well for function estimation may give rather poor derivative estimates.

2.1.3 The spline basis system for non-periodic data

Spline functions are the most common choice of approximation system for non-periodic functional data or parameters. Splines combine the fast computation of polynomials with substantially greater flexibility, often achieved with only a modest number of basis functions. Moreover, basis systems have been developed for spline functions that require an amount of computation that is proportional to n, a vital property since many applications involve thousands or millions of observations.

The first step in defining a spline is to divide the interval over which a function is to be approximated into L subintervals separated by values τ_l, $l = 1, \ldots, L-1$ that are called breakpoints or knots. Over each interval, a spline is a polynomial of specified order m. The order of a polynomial is the number of constants required to define it, and is one more than its degree, its highest power. Adjacent polynomials join up smoothly at the breakpoint which separates them for splines of order greater than one, so that the function values are constrained to be equal at their junction. Moreover, derivatives up to order $m-2$ must also match up at these junctions. The total number of degrees of freedom in the fit equals the order of the polynomials plus the the number of interior breakpoints. If there are no interior knots, the spline

reverts to being a simple polynomial.

The main way to gain flexibility in a spline is to increase the number of breakpoints. In general, we want more breakpoints over regions where the function exhibits the most complex variation, and fewer where the function is only mildly nonlinear. Breakpoints are not quite the same thing as knots. In fact, we can have two or more breakpoints that move together to coalesce or be coincident. When this happens, there is a loss of continuity condition for each additional coincident breakpoint. In this way, we can engineer abrupt changes in a derivative or even a function value at pre-specified breakpoints. Thus, the term breakpoint refers to the number of unique knot values, while the term knot refers to the sequence of values at breakpoints, so some breakpoints can be associated with multiple knots. The number of parameters required to define a spline function in the usual situation of one knot per breakpoint is the order plus the number of interior knots, $m + L - 1$.

2.1.4 The B-spline basis for spline functions

To construct a spline function we specify a system of basis functions $\Phi_k(t)$, with these following essential properties:

- Each basis function $\Phi_k(t)$ is itself a spline function as defined by an order m and a knot sequence τ;
- Since a multiple of a spline function is still a spline function, and since sums and differences of splines are also splines, any linear combination of these basis functions is a spline function;
- Any spline function defined by m and τ can be expressed as a linear combination of these basis functions.

Although there are many ways that such systems can be constructed, the B-spline basis system developed by de Boor (2001) is the most popular, and code for working with B-splines is available in a wide range of programming languages, including R, S-PLUS and MATLAB. Other spline basis systems are truncated power functions, M-splines and natural splines, and these and others are discussed by de Boor (2001)

and Schumaker (1981).

An order m B-spline basis function has the compact support property, that is it is positive over no more than m adjacent intervals. This property has a great importance for efficient computation, in fact if there are K B-spline basis functions then the order K matrix of inner products of these functions will be band-structured, with only $m-1$ sub-diagonals above and below the main diagonal containing nonzero values. So, the computation of spline function can be organized so as to increase only linearly with K.

Considering the domain over our B-spline system is defined, as we move from the left boundary towards the center, the intervals over which the basis functions are positive increase, while their transition to the left boundary varies in smoothness, with the left-most spline being discontinuous, the next being continuous only, and the third being once-differentiable. That we lose differentiability at the boundaries makes good sense, since we normally have no information about what the function we are estimating is doing beyond the interval on which we collect data. We therefore are allowing for the possibility that the function may be discontinuous beyond the boundaries. This boundary behavior of B-spline basis functions is achieved by placing m knots at the boundaries. That is, when B-splines are actually computed, the knot sequence τ is extended at each end to add an additional $m - 1$ replicates of the boundary knot value. There are some applications where we do not want $m - 2$ continuous derivatives at certain fixed points in the interior of the interval. This can be readily accommodated by B-splines placing a knot at such fixed points, and then for each reduction in differentiability an additional knot is placed at that location as well. The notation $B_k(t, \tau)$ is often used to indicate the value at t of the Bspline basis function defined by the breakpoint sequence τ. Here k refers to the number of the largest knot at or to the immediate left of value t. The $m-1$ knots added to the initial breakpoint are also counted in this scheme, and this is consistent with the fact that the first m B-spline basis functions all have supports all beginning at the left boundary. This notation gives us $m + L - 1$ basis functions, as required in the usual case where all interior knots are discrete. According to this notation, a spline

function $S(t)$ with discrete interior knots is defined as

$$S(t) = \sum_{k=1}^{m+L-1} c_k B_k(t, \tau). \tag{2.6}$$

About positioning the interior breakpoints, many applications suggest to equal spacing them, if the data are relatively equally spaced. Otherwise, it may be wiser to place a knot at every j-th data point, were j is a pre-fixed number. A special case is that of smoothing splines where a breakpoint is placed at each argument value. Finally, one can depart from either of these simple procedures to place more knots in regions known to contain high curvature, and fewer where there is less.

2.1.5 Smoothing

2.1.5.1 Smoothing functional data by least squares

Our goal is to fit the discrete observations y_j, $j = 1, \ . \ . \ . \ , n$ using the model $y_j = x(t_j) + \epsilon_j$, and a basis function expansion for $x(t)$ of the form

$$x(t) = \sum_{k}^{K} c_k \Phi_k(t) = c'\Phi \tag{2.7}$$

The vector c of length K contains the coefficients c_k. Let us define the n by K matrix Φ as containing the values $\Phi_k(t_j)$.

Unweighted least squares fits

A simple linear smoother is obtained if we determine the coefficients of the expansion c_k by minimizing the least squares criterion

$$SMSSE(y|c) = \sum_{j=1}^{n} [y_j - \sum_{k}^{K} c_k \Phi_k(t_j)]^2 \tag{2.8}$$

The criterion is expressed more cleanly in matrix terms as

$$SMSSE(y|c) = (y - \Phi c)'(y - \Phi c) \tag{2.9}$$

The right side is also often written in functional notation as $||y - \Phi c||^2$. Taking the derivative of criterion SMSSE(y|c) with respect to c yields the equation

$$2\Phi\Phi' c - 2\Phi' y = 0 \tag{2.10}$$

and solving this for c provides the estimate $\hat{c}$ that minimizes the least squares solution,

$$\hat{c} = (\Phi'\Phi)^{-1}\Phi' y \tag{2.11}$$

The vector $\hat{y}$ of fitted values is

$$\hat{y} = \Phi\hat{c} = \Phi(\Phi'\Phi)^{-1}\Phi' y \tag{2.12}$$

Simple least squares approximation is appropriate in situations where we assume that the residuals ϵ_j about the true curve are independently and identically distributed with mean zero and constant variance σ^2. That is, we prefer this approach when we assume the standard model for error.

Weighted least squares fits

The standard model for error will often not be realistic. To deal with nonstationary and/or autocorrelated errors, we may need to bring in a differential weighting of residuals by extending the least squares criterion to the form

$$SMSSE(y|c) = (y - \Phi c)' W (y - \Phi c) \tag{2.13}$$

where W is a symmetric positive definite matrix that allows for unequal weighting of squares and products of residuals. If the variance-covariance matrix Σ_e for the residuals ϵ_j is known, we get W in this way:

$$W = \Sigma_e^{-1}$$

When an estimate of the complete Σ_e is not feasible, the covariances among errors are often assumed to be zero, and then W is diagonal with reciprocals of the error variance associated with the y_j's in the diagonal. We can always set it to I if the standard model is assumed. The weighted least squares estimate $\hat{c}$ of the coefficient vector c is

$$\hat{c} = (\Phi' W \Phi)^{-1} \Phi' W y \tag{2.14}$$

Least squares fits as linear transformations of the data

We consider all smoothing methods having the property of being linear. This property simplifies computational issues considerably, so we need to consider what linearity in a smoothing procedure means.

A linear smoother estimates the function value $\hat{y}_j = \hat{x}(t_j)$ by a linear combination of the discrete observations

$$\hat{x}(t_j) = \sum_{l=1}^{n} S_j(t_l) y_l \tag{2.15}$$

where $S_j(t_l)$ weights the lth discrete data value in order to generate the fit to y_j . In matrix terms,

$$\hat{x}(t) = Sy \tag{2.16}$$

where $\hat{x}(t)$ is a column vector containing the values of the estimate of function x at each sampling point t_j. In the unweighted least squares case, for example, we see that

$$S = \Phi(\Phi'\Phi)^{-1}\Phi' \tag{2.17}$$

In regression analysis this matrix is called the "hat matrix" because it converts the dependent variable vector y into its fit $\hat{y}$. In the context of least squares estimation, the smoothing matrix has the property of being a projection matrix. This means that it creates an image of data vector y on the space spanned by the columns of matrix Φ such that the residual vector $e = y - \hat{y}$ is orthogonal to the fit vector $\hat{y}$,

$$(y - \hat{y})'\hat{y} = 0 \tag{2.18}$$

This in turn implies that the smoothing matrix has the idempotency property $SS = S$ The corresponding smoothing matrix for weighted least squares smoothing is

$$S = \Phi(\Phi' W \Phi)^{-1} \Phi' W \tag{2.19}$$

Matrix S is still an orthogonal projection matrix, but now the residual and fit vectors are orthogonal in the sense that

$$(y - \hat{y})' W \hat{y} = 0 \tag{2.20}$$

The linearity of a smoother is an important feature for various reasons: The linearity property

$$S(ay + bz) = aSy + bSz$$

is important for working out various properties of the smooth representation and, also, it implies relatively fast computation. On the other hand, some nonlinear smoothers may be more adaptive to different behavior in different parts of the range of observation, and may be robust to outlying observations.

The model for observed data offers an image of the data that has fewer degrees of freedom than those of the original data. The concept of the degrees of freedom of a fit means simply the number of parameters estimated from the data that are required to define the model. The same notion of degrees of freedom is used to data smoothing using least squares, where the number of parameters is the length K of the coefficient vector c. The number of degrees of freedom for error is therefore $n - K$. However, there is a more general way of computing the effective degrees of freedom of a smooth fit to the data and consequently the corresponding degrees of freedom for error. That is, to use the "hat" matrix S by defining the degrees of freedom of the smooth fit to be

$$df = \text{trace}(S) \tag{2.21}$$

where the trace of a square matrix means the sum of its diagonal elements. This

more general definition yields exactly K for least squares fits, and therefore does not represent anything new.

Choosing the number K of basis functions

The larger the order of the expansion K, the better the fit to the data, but of course we risk also fitting noise or variation that we wish to ignore. On the other hand, if K is too small, we may miss some important aspects of the smooth function x that we are trying to estimate. This trade-off can be expressed in another way. For large values of K and n the bias in estimating $x(t)$, that is

$$Bias[\hat{x}(t)] = x(t) - E[\hat{x}(t)] \tag{2.22}$$

is small. In fact, if the notion of additive errors having expectation zero holds, the bias will be zero for $K = n$. But, one of the main reasons of doing smoothing is to reduce the influence of noise or ignorable variation on the estimate $\hat{x}$. Consequently, we are also interested in the variance of estimate

$$Var[\hat{x}(t)] = E[\hat{x}(t) - E[\hat{x}(t)]^2] \tag{2.23}$$

If $K = n$, this will be unacceptably high. So, to reduce variance we have to look for smaller values of K, but of course not so small as to make the bias unacceptable. Mean-squared error expresses what we want to achieve:

$$MSE[\hat{x}(t)] = E[\hat{x}(t) - x(t)^2] \tag{2.24}$$

also called the L^2 loss function. In most applications we can't actually minimize this since we have no way of knowing what $x(t)$ is without using the data. However, by the simple additive decomposition we obtain one of the most important equation in statistics:

$$MSE[\hat{x}(t)] = Bias^2[\hat{x}(t)] + Var[\hat{x}(t)] \tag{2.25}$$

So, it would be worthwhile to tolerate a little bias if the result is a big reduction in sampling variance. In fact, this is almost always the case, and is the fundamental reason for smoothing data in order to estimate functions. Sampling variance increases rapidly when we use too many basis functions, but squared bias tends to decay more gently to zero at the same time. So, the best results for totaled mean squared error are obtained at the minimum.

Computing sampling variances and confidence limits

The estimation of the coefficient vector c of the basis function expansion $x = c'\Phi$ by minimizing least squares defines a linear mapping from the raw data vector y to the estimate. With this mapping in hand, it is relatively simple to compute the sampling variance of the coefficient vector and of anything that is linearly related to it. If a random variable y is normally distributed with a variance-covariance matrix Σ_y, then the random variable Ay defined by any matrix A has the variance-covariance matrix

$$Var[Ay] = A\Sigma_y A'$$

Now in this linear modelling situation, the model for the data vector y, in this case $x(t)$, is regarded as a fixed effect having zero variance. Consequently, the variance-covariance matrix of y using the model $y = x(t) + \epsilon$ is the variance-covariance matrix Σ_e of the residual vector σ. So, we have to use the information in the actual residuals to replace the population quantity Σ_e by a reasonable sample estimate $\hat{\Sigma}_e$.

When we are smoothing a single curve, it is important to use methods which produce relatively unbiased estimate of variance in order to avoid underestimating sampling variance. For example, if the standard model for error is accepted,

$$s^2 = \frac{1}{n-K}\sum_{j}^{n}(y_j - \hat{y}_j)^2$$

is much preferred as an estimate of σ^2 than the maximum likelihood estimate that involves dividing by n. So, one reasonable strategy for choosing K is to add basis

functions until s^2 fails to decrease substantially.

Confidence limits are typically computed by adding and subtracting a multiple of the standard errors, that is, the square root of the sampling variances, to the actual fit. This kind of confidence limits are called point-wise because they reflect confidence regions for fixed values of t rather than regions for the whole curve. But, they may be problematic for two important ways. First, it is implicitly assumed that K is a fixed constant, but K for smoothing problems is more like a parameter estimated from the data, and consequently the size of these confidence limits does not reflect the uncertainty in our knowledge of K. Secondly, the smooth curve to which we add and subtract multiples of the standard error to get point-wise limits is itself subject to bias, and especially in regions of high curvature. Thus, the confidence limits calculated in this way are themselves biassed, and the region covered by them may not be quite as advertised.

2.1.5.2 Smoothing functional data with a roughness penalty

A more powerful option for approximating discrete data by a function is the roughness penalty or regularization approach that retains the advantages of the basis function and local expansion smoothing techniques, but circumvents some of their limitations. Finally, it often produces better results, and especially in the estimation of derivatives. Roughness penalty methods are based on optimizing a fitting criterion that defines what a smooth of the data is trying to achieve. But here the precise meaning of "smooth" is expressed explicitly at the level of the criterion being optimized, rather than implicitly in terms of the number of basis functions being used.

Mean squared error trade off

The spline smoothing method estimates a curve x from observations $y_j = x(t_j) + \epsilon_j$ by making explicit two conflicting goals in curve estimation. On the one hand, we want that the estimated curve gives a good fit to the data, for example in terms of the residual sum of squares $\sum[y_j - x(t_j)]^2$. On the other hand, we do not want the

fit to be too good if this results in a curve x that is excessively "wiggly" or locally variable. These competing aims correspond to the elements of the basic principle of statistics

$$\text{Mean squared error} = \text{Bias}^2 + \text{Sampling variance}$$

A completely unbiased estimate of the function value $x(t_j)$ can be produced by a curve fitting y_j exactly, since this observed value is itself an unbiased estimate of $x(t_j)$ according to our error model. But any such curve must have high variance, manifested in the rapid local variation of the curve. In spline smoothing the mean squared error, MSE, is one way of capturing the quality of estimate. MSE can often be dramatically reduced by sacrificing some bias in order to reduce sampling variance, and this is a key reason for imposing smoothness on the estimated curve. By requiring that the estimate varies gently from one value to another, we are "borrowing information" from neighboring data values, thereby expressing our faith in the regularity of the underlying function x that we are trying to estimate. The roughness penalty makes explicit what we sacrifice in bias to achieve an improvement MSE or some other loss function.

Quantifying roughness

We can quantify the notion "roughness" of a function by using the square of the second derivative $[D^2x(t)]^2$ of the function at t. This represents its curvature at t, since a straight line, that has no curvature, also has a zero second derivative. Consequently, a natural measure of a function's roughness is the integrated squared second derivative

$$PEN_2(x) = \int [D^2x(s)]^2 ds \tag{2.26}$$

Highly variable functions can be expected to yield high values of $PEN_2(x)$ because their second derivatives are large over at least some of the range of interest. We may need to generalize the roughness penalty by allowing a derivative $D^m x$ of arbitrary

order, so we can work with the penalty

$$PEN^m(x) = \int [D^m x(s)]^2 ds \tag{2.27}$$

The penalized sum of squared errors fitting criterion

Let $x(t)$ be the vector resulting from function x being evaluated at the vector t of argument values. We define the penalized residual sum of squares as

$$PENSSE\lambda(x|y) = [y - x(t)]'W[y - x(t)]^2 + \lambda PEN^2(x) \tag{2.28}$$

The estimate of the function is obtained by finding the function x that minimizes $PENSSE\lambda(x)$ over the space of functions x for which $PEN^2(x)$ is defined. The parameter λ is a smoothing parameter that measures the rate of exchange between fit to the data, as measured by the residual sum of squares in the first term, and variability of the function x, as quantified by $PEN^2(x)$ in the second term. As λ becomes larger, not linear functions must incur a more substantial roughness penalty through the term $PEN^2(x)$, and consequently the composite criterion $PENSSE\lambda(x)$ must place more emphasis on the smoothness of x and less on fitting the data. For this reason, as $\lambda \to \infty$ the fitted curve x must approach the standard linear regression to the observed data, where $PEN^2(x) = 0$. On the other hand, for small λ the curve tends to become more variable since there is less penalty placed on its roughness, and as $\lambda \to 0$ the curve x approaches an interpolant to the data, satisfying $x(t_j) = y_j$ for all j.

The structure of a smoothing spline

Assuming that x has a second derivative and that the sampling points t_j, $j = 1, ..., n$ are distinct, what kind of function minimizes this penalized error sum of squares? A remarkable theorem, found in de Boor (2002) and other more advanced texts on smoothing, states that the curve x that minimizes $PENSSE\lambda(x|y)$ is a cubic spline with knots at the data points t_j. In this way the issue of locating the knots

disappears. Smoothing splines adapt naturally to unequal spacing of sampling points, and thus automatically take advantage of regions where data density is high, and at the same time are especially smooth over regions where there are few observations. The most common computational technique for spline smoothing is to use an order four B-spline basis function expansion with knots at the sampling points, and to minimize the $PENSSE_\lambda(x|y)$ with respect to the coefficients of the expansion.

Without a roughness penalty, the coefficient vector c in the expansion

$$x(t) = \sum_k^K c_k \Phi_k(t) = c'\Phi(t) = \Phi'(t)c \tag{2.29}$$

where c is the K-vector of coefficients and Φ is the K-vector of basis functions, has the solution

$$\hat{c} = (\Phi' W \Phi)^{-1} \Phi' W' y \tag{2.30}$$

where n by K matrix Φ contains the values of the K basis functions at the n sampling points, W is a weight matrix to allow for possible covariance structure among residuals, and where y is the vector of discrete data to be smoothed. The corresponding expression for the vector of fits to the data is

$$\hat{y} = \Phi(\Phi' W \Phi)^{-1} \Phi' W y = S_\Phi y \tag{2.31}$$

where S_Φ is the projection operator

$$S_\Phi = \Phi(\Phi' W \Phi)^{-1} \Phi' W \tag{2.32}$$

corresponding to the basis system Φ. We can re-express the roughness penalty $PEN_m(x)$ in matrix terms as follows.

$$PEN_m(x) = \int [D^m x(s)]^2 ds = \int [D^m c'\Phi(s)]^2 ds = \int c' D^m \Phi(s) D^m \Phi'(s) c ds =$$

$$= c'[\int D^m \Phi(s) Dm \Phi'(s) ds] c = c' R c \tag{2.33}$$

where

$$R = \int D^m \Phi(s) D^m \Phi'(s) ds = \int D^m \Phi D^m \Phi' \tag{2.34}$$

By adding the error sum of squares $SSE(y|c)$ and $PEN_m(x)$ multiplied by a smoothing parameter λ, we obtain

$$PENSSE_m(y|c) = (y - \Phi c)' W (y - \Phi c) + \lambda c' R c \tag{2.35}$$

Taking the derivative with respect to parameter vector c, we obtain

$$-2\Phi' W y + \Phi' W \Phi c + \lambda R c = 0 \tag{2.36}$$

from which we obtain the expression for the estimated coefficient vector

$$\hat{c} = (\Phi' W \Phi + \lambda R)^{-1} \Phi' W y \tag{2.37}$$

The expression for the data-fitting vector $\hat{y}$ is

$$\hat{y} = \Phi(\Phi' W \Phi + \lambda R)^{-1} \Phi' W y = S_{\Phi,\lambda} y \tag{2.38}$$

where the order n symmetric "hat" matrix is

$$S_{\Phi,\lambda} = \Phi(\Phi' W \Phi + \lambda R)^{-1} \Phi' W \tag{2.39}$$

Computing the matrix R will generally require approximating the integral by a numerical quadrature scheme, although exact expressions are possible where both B-spline and Fourier bases are involved. In fact, it is seldom necessary to have very high accuracy in the approximation.

$S_{\Phi,\lambda}$ is also useful for computing a degrees of freedom value for a spline smooth

$$df(\lambda) = trace S_{\Phi,\lambda} \tag{2.40}$$

Choosing the smoothing parameter

When we fit data using a roughness penalty, we switch from defining the smooth in terms of degrees of freedom K to defining the smooth in terms of the smoothing parameter λ. Although from a mathematical perspective we can contemplate any positive values of λ, the realities of floating point computation actually impose some severe limits due to the need to solve a system of linear equations with the coefficient matrix

$$M(\lambda) = \Phi' W \Phi + \lambda R \tag{2.41}$$

The two matrices $\Phi' W \Phi$ and R can have elements of radically different sizes. In particular, the size of

$$||R|| = \sqrt{\sum_k \sum_l r_{kl}^2} \tag{2.42}$$

increases by roughly an order of magnitude for each increase in the order m of derivative that is used to define the roughness penalty. Now R itself has rank $K - m$, and so cannot itself be useful as a coefficient matrix. This implies that we cannot have λR so large as to overwhelm $\Phi' W \Phi$; otherwise, attempting to invert $M(\lambda)$ will either produce an error message or a result full of rounding error that can lead us to seriously wrong results. One possibility is that the size of λR should not be more than 10^10 times the size of $\Phi' W \Phi$. On the lower limit side, we cannot always use $\lambda = 0$; Again, a rule of thumb can be proposed: Choose λ at least large enough to ensure that the size of λR is at least with ten orders of magnitude of the size of $\Phi' W \Phi$.

The cross-validation (CV) and the generalized cross-validation (GCV)

The basic idea behind cross-validation is to set part of the data to one side, calling it a validation sample, and fit the model to the balance of the data, called the training sample. In that way, we see how well the model fits data that were not used to estimate the model. A versatile technique for choosing a smoothing parameter is to leave only one observation out as the validation sample, fitting the data to the rest, and then estimating the fitted value for the left out data value. If this procedure is repeated for each observation in turn, and the resulting error sum of squares summed

over all values, the result is the cross-validated error sum of squares. After computing this criterion over a range of values of λ, we choose the value that yields its minimum. Cross-validation can be used in a lot of situations, and rests only on the assumption that observations are relatively independent of one another. However, the method has two problems. First, it is usually computationally intensive. The second problem is that minimizing CV can lead to undersmoothing the data because the method tends too often to favor fitting noisy or high-frequency types of variation that we would prefer to ignore.

GCV was originally developed as a simpler version of the cross-validation procedure that avoided the need to re-smooth n times. But it is also more reliable than cross-validation, in fact it have less tendency to under-smooth. The criterion is usually expressed as

$$GCV(\lambda) = \frac{n^{-1}SSE}{[n^{-1}trace(I - S_{\Phi,\lambda})]^2} \tag{2.43}$$

where df is the equivalent degrees of freedom measure and S_λ is the smoothing operator. But it can be more revealing to use the equivalent expression

$$GCV(\lambda) = \frac{n}{n - df(\lambda)} \frac{SSE}{n - df(\lambda)} \tag{2.44}$$

Notice that this is a twice-discounted mean squared error measure. The right factor is the unbiased estimate of error variance σ^2, and thus represents some discounting by subtracting $df(\lambda)$ from n. The left factor further discounts this estimate by multiplying by $\frac{n}{(n-df(\lambda))}$. The minimization of GCV with respect to λ will inevitably involve trying a large number of values of λ, whether grid-search or a numerical optimization algorithm is used. A preliminary generalized eigenanalysis can speed up the computation of GCV(λ). Criterion GCV can be expressed in terms of the n by N data matrix Y, the $n \times K$ matrix Φ of basis function values and the order K penalty matrix R as follows

$$GCV(\lambda) = \frac{ntraceY'[I - S_{\Phi,\lambda}]^{-2}Y}{trace[I - S_{\Phi,\lambda}]^2} \tag{2.45}$$

where the "hat" matrix $S_{\Phi,\lambda}$ has the expression

$$S_{\Phi,\lambda} = \Phi M(\lambda)^{-1}\Phi' W$$

and where

$$M(\lambda) = \Phi' W \Phi + \lambda R$$

We actually don't need to invert $M(\lambda)$ each time we change *lambda*, but we need to solve a linear system of equations for which it is the coefficient matrix, and this is that we want to avoid. This can be achieved if we first solve the generalized eigenvalue problem

$$RV = \Phi' W \Phi V D \tag{2.46}$$

where D is the matrix of eigenvalues of R in the metric defined by $\Phi' W \Phi$ and V, the columns of which are the corresponding eigenvectors of R, satisfy the orthogonality condition

$$V' \Phi' W \Phi V = I \tag{2.47}$$

We now express, for any new value of λ, the required inverse very efficiently as

$$M(\lambda)^{-1} = V(I + \lambda D)^{-1} V' \tag{2.48}$$

since the matrix now being inverted is diagonal. Moreover, taking the derivative of GCV(λ) involves calculating the matrix

$$M(\lambda)^{-1}\Phi' W \Phi M(\lambda)^{-1} = V(I + \lambda D)^{-2} V' \tag{2.49}$$

so that providing a derivative value to a numerical optimization algorithm is also computationally efficient and likely to decrease the number of evaluations of GCV(λ) substantially.

Confidence intervals for function values and functional probes

We now want to see how to compute confidence limits on some useful quantities that

depend on an estimated function x that has been computed by smoothing a vector of discrete data y with a roughness penalty . For example, we can construct a pair of confidence limits such that the probability that the true value of $x(t)$ lies within these limits is a specified value, such as 0.95. We wish to construct an upper and a lower curve such that the probability that the entire true curve lies between these functional limits, and this can be done with computationally intensive methods such as bootstrapping (Efron and Tibshirani, 1993).

More generally, we may wish to examine quantities of the form

$$\rho_\xi(x) = \int \xi(t)x(t)dt \tag{2.50}$$

We use the term functional probe for the quantity $\rho_\xi(x)$ and linear probe function for the weighting function ξ that defines it. The probe function, in turn, is chosen so as to highlight some interesting feature, such as a peak, valley, or difference between function values over two non-overlapping regions. A probe is a generalization of the notion of a contrast in analysis of variance, used there to probe a set of treatment effects for specific types of variation. However, there is no need for the values of ξ to integrate to zero.

scrivere qualcosa sui metodi supervised e unsupervised

2.2 Unsupervised Learning

In the supervised learning setting, we typically have access to a set of p features $X_1, X_2, ..., X_p$, measured on n observations, and a response Y also measured on those same n observations. The aim is then to predict Y using $X_1, X_2, ..., X_p$. On unsupervised learning, we have only a set of features $X_1, X_2, ..., X_p$ measured on n observations and we are not interested in prediction, because we do not have an associated response variable Y to predict. Rather, the goal is to discover interesting things about the measurements on $X_1, X_2, ..., X_p$. There are two particular types of unsupervised learning: principal components analysis, a tool used for data

visualization or data pre-processing before applying supervised techniques, and clustering, a broad class of methods for discovering unknown subgroups in data. [27]

2.2.1 Principal components analysis for functional data

PCA for multivariate data

The central concept in multivariate statistic is that of taking a linear combination of variable values,

$$f_i = \sum_{j=1}^{p} \beta_j x_{ij}, i = 1, ..., N \tag{2.51}$$

where β_j is a weighting coefficient applied to the observed values x_{ij} of the jth variable. It can be expressed as

$$f_i = \beta' x_i, i = 1, ..., N \tag{2.52}$$

where β is the vector $(\beta_1, ..., \beta_p)'$ and x_i is the vector $(x_{i1}, ..., x_{ip})'$. It is possible to define principal components analysis in terms of stepwise procedure, which defines sets of normalized weights that maximize variation in the f_i's

1. Find the weight vector $\xi_1 = (\xi_{11}, ..., \xi_{p1})'$ for which the linear combination values

$$f_{i1} = \sum_j \xi_{j1} x_{ij} = \xi_1' x_i \tag{2.53}$$

have the largest possible mean square $N^{-1} \sum_i f_{i1}^2$ subject to the constraint

$$\sum_j \xi_{j1}^2 = ||\xi_1||^2 = 1 \tag{2.54}$$

2. Execute second and subsequent steps, possibly up to a limit of the number of variables p. On the mth step, compute a new weight vector ξ_m with components ξ_{jm} and new values $f_{im} = \xi_m' x_i$. Thus, the values f_{im} have maximum mean square,

subject to the constraint $||\xi_m||^2 = 1$ and the $m-1$ additional constraint(s)

$$\sum_j \xi_{jk}\xi_{jm} = \xi_k'\xi_m = 0, k < m \tag{2.55}$$

In the first step, maximizing the mean square, we are identifying the strongest and most important mode of variation in the variables. The unit sum of squares constraint on the weights is essential for having a well defined problem. In fact, without it, the mean squares of the linear combination values could be made arbitrarily large. On second and subsequent steps, we look for the most important modes of variation requiring the weights defining them to be orthogonal to those identified previously, so that they are indicating something new. Of course, the amount of variation measured in terms of $N^{-1}\sum_i f_{im}^2$ will decline on each step. At some point, usually well short of the maximum index p, we expect to lose interest in modes of variation thus defined. The values of the linear combinations f_{im} are called principal component scores and are very useful for describing what these important components of variation mean in terms of the characteristics of specific cases or replicates. To be sure, the mean is a very important aspect of the data, but we already have an easy technique for identifying it. Therefore, we usually subtract the mean for each variable from corresponding variable values before doing PCA. When this is done, maximizing the mean square of the principal component scores corresponds to maximizing their sample variance.

Defining PCA for functional data

In functional PCA, instead of variable values we have function values $x_i(s)$, so the continuos index s has replaced the discrete index j of the multivariate context. When we were considering vectors, the appropriate way of combining a weight vector β with a data vector x was to calculate the inner product

$$\beta' x = \sum_j \beta_j x_j \tag{2.56}$$

. When β and x are functions $\beta(s)$ and $x(s)$, summations over j are replaced by integrations over s to define the inner product

$$\int \beta x = \int \beta(s)x(s)ds \tag{2.57}$$

Within the principal components analysis, the weights β_j become functions with values $\beta_j(s)$. The principal component scores corresponding to weight β are now

$$f_i = \int \beta x_i = \int \beta(s)x_i(s)ds \tag{2.58}$$

In the first functional PCA step, the weight function $\xi_1(s)$ has to maximize $N^{-1}\sum_i f_{i1}^2 = N^{-1}\sum_i(\int \xi_1 x_i)^2$ subject to the continuous analogue $\int \xi_1(s)^2 ds = 1$ of the unit sum of squares constraint. This time, the notation $||\xi 1||^2$ is used to mean the squared norm $\int \xi_1(s)^2 ds = \int \xi_1^2$ of the function ξ_1. As for multivariate PCA, the weight function ξ_m is also required to satisfy the orthogonality constraint(s) $\int \xi_k \xi_m = 0, k < m$ on subsequent steps. Each weight function has the task of defining the most important mode of variation in the curves subject to each mode being orthogonal to all modes defined on previous steps.

Defining an optimal empirical orthonormal basis

We want to find a set of exactly K orthonormal functions ξ_m so that the expansion of each curve in terms of these basis functions approximates the curve as closely as possible. Since these basis functions are orthonormal, it follows that the expansion will be of the form

$$\hat{x}_i(t) = \sum_{k=1}^{K} f_{ik}\xi_k(t) \tag{2.59}$$

where f_{ik} is the principal component value $\int x_i \xi_k$. As a fitting criterion for an individual curve, consider the integrated squared error

$$||xi - \hat{x}_i||^2 = \int [x(s) - \hat{x}(s)]^2 ds \tag{2.60}$$

and as a global measure of approximation,

$$PCASSE = \sum_{i=1}^{N} ||x_i - \hat{x}_i||^2 \tag{2.61}$$

For minimizing the error criterion, we use the same set of principal component weight functions that maximize variance components as defined above as basis. For this reason, these functions ξ_m are referred to in some fields as empirical orthonormal functions, because they are determined by the data they are used to expand.

PCA and eigenanalysis

Assume that our observed values, x_{ij} in the multivariate context and $x_i(t)$ in the functional situation, result from subtracting the mean variable or function values, so that their sample means $N^{-1}\sum_i x_{ij}$, or cross-sectional means $N^{-1}\sum_i x_i(t)$, respectively, are zero. In multivariate data analysis the principal components analysis is described as the task of finding the eigenvalues and eigenvectors of the covariance or correlation matrix. The logic for this is as follows. Let the $N \times p$ matrix X contain the values x_{ij} and the vector ξ of length p contain the weights for a linear combination. Then the mean square criterion for finding the first principal component weight vector can be written as

$$\max_{\xi'\xi=1} N^{-1}\xi' X' X \xi\xi \tag{2.62}$$

since the vector of principal component scores f_i can be written as $X\xi$. Use the p×p matrix V to indicate the sample variance-covariance matrix $V = N^{-1}X'X$. This criterion can now be expressed

$$\max_{\xi'\xi=1} \xi' V \xi \tag{2.63}$$

This maximization problem is solved by finding the solution with largest eigenvalue ρ of the eigenvector problem or eigenequation

$$V\xi = \rho\xi \tag{2.64}$$

There is a sequence of different eigenvalue-eigenvector pairs (ρ_j, ξ_j) satisfying this equation, and the eigenvectors ξ_j are orthogonal. Because the mean of each column of X is usually subtracted from all values in that column as a preliminary to principal components analysis, the rank of X is $N-1$ at most, and hence the $p \times p$ matrix V has, at most, $\min p, N-1$ nonzero eigenvalues ρ_j. For each j, the eigenvector ξ_j satisfies the maximization problem subject to the additional constraint of being orthogonal to all the eigenvectors $\xi_1, \xi_2, ..., \xi_{j-1}$ found so far. This is exactly what was required of the principal components in the second step laid out. Therefore the multivariate PCA problem is equivalent to the algebraic and numerical problem of solving the eigenequation. Now consider the functional version of PCA. Define the covariance function $v(s,t)$ by

$$v(s,t) = N^{-1} \sum_{i=1}^{N} x_i(s) x_i(t) \tag{2.65}$$

The principal component weight functions $\xi_j(s)$. Each of these satisfies the equation

$$\int v(s,t)\xi(t)dt = \rho\xi(s) \tag{2.66}$$

for an appropriate eigenvalue ρ. The left side is an integral transform V of the weight function ξ defined by

$$V\xi = \int v(\mathring{\mathrm{u}}, t)\xi(t)dt \tag{2.67}$$

This integral transform is called the covariance operator V . Therefore we may also express the eigenequation directly as

$$V\xi = \rho\xi \tag{2.68}$$

where ξ is now an eigenfunction rather than an eigenvector.

Computational methods for functional PCA

To express each function x_i as a linear combination of known basis functions Φ_k is a good way of reducing the eigenequation to discrete or matrix form. The number K

of basis functions used depends on many considerations: how many discrete sampling points n were in the original data, whether some level of smoothing was to be imposed by using $K < n$, how efficient or powerful the basis functions are in reproducing the behavior of the original functions, and so forth. Now suppose that each function has basis expansion

$$x_i(t) = \sum_{k=1}^{K} c_{ik}\Phi_k(t) \tag{2.69}$$

We may write this more compactly by defining the vector-valued function x to have components $x_1, ..., x_N$, and the vector-valued function Φ to have components $\Phi_1, ..., \Phi_K$. We may then express the simultaneous expansion of all N curves as

$$x = C\Phi \tag{2.70}$$

where the coefficient matrix C is $N \times K$. In matrix terms the variance-covariance function is

$$v(s,t) = N^{-1}\Phi(s)'C'C\Phi(t) \tag{2.71}$$

Define the order K symmetric matrix W to have entries

$$w_{k_1,k_2} = \int \Phi_{k_1}\Phi_{k_2} \tag{2.72}$$

or $W = \int \Phi\Phi'$. Now suppose that an eigenfunction ξ for the eigenequation has an expansion

$$\xi(s) = \sum_{k=1}^{K} b_k\Phi_k(s) \tag{2.73}$$

or, in matrix notation, $\xi(s) = \Phi(s)'b$. This yields

$$\int v(s,t)\xi(t)dt = \int N^{-1}\Phi(s)'C'C\Phi(t)\Phi(t)'bdt = \Phi(s)'N^{-1}C'CWb \tag{2.74}$$

Therefore the eigenequation can be expressed as

$$\Phi(s)'N^{-1}C'CWb = \rho\Phi(s)'b \tag{2.75}$$

Since this equation must hold for all s, this implies the purely matrix equation

$$N^{-1}C'CWb = \rho b \tag{2.76}$$

But note that $||\xi|| = 1$ implies that $b'Wb = 1$ and, similarly, two functions ξ_1 and ξ_2 will be orthogonal if and only if the corresponding vectors of coefficients satisfy $b_1'Wb_2 = 0$. To get the required principal components, we define $u = W^{1/2}b$, solve the equivalent symmetric eigenvalue problem

$$N^{-1}W^{1/2}C'CW^{1/2}u = \rho u \tag{2.77}$$

and compute $b = W^{-1/2}u$ for each eigenvector.

2.2.2 Clustering methods for functional data

Clustering is a set of techniques used for finding subgroups, called clusters, in a data set. When we cluster the observations of a data set $x_1, ..., x_n$, our aim is to partition them into M distinct groups so that the observations within each group are quite similar to each other, while observations between different groups are quite dissimilar from each other, based on definition of a dissimilarty measure for two or more observations. There are different approaches for functional clustering that can be grouped in *raw-data clustering*, *filtering methods*, *adaptive methods*, and *distance-based methods*.

Raw-data clustering

The raw-data clustering method consists in the clustering of discretized version X_{ij} of the functional observations X_i by means of classical multivariate methods. So, we do not need to reconstruct the functional data, rather we can use well-established multivariate algorithms. The k-mean method is one of the most popular clustering algorithm. In the functional setting, k-means aims to partition the observations into

M clusters $C_1^*, ..., C_M^*$ such that the within-cluster sum of squares is minimized,

$$[C_1^*, ..., C_M^*] = \underset{C_1,...,C_M}{\operatorname{argmin}} \sum_{m=1}^{M} \sum_{X_i \in C_m} (X_i - \mu_m)'(X_i - \mu_m) \tag{2.78}$$

where $C_1, ..., C_M$ are all the possible observation partitions in M groups, $X_i = (X_{i1}, ..., X_{im_i})'$, and μ_m is the mean vector of the observations in C_m.

Instead, hierarchical clustering [20] produces a division in which clusters at each level of the hierarchy are formed by all and only the clusters of the lower levels, without overlapping. There are two main approaches for hierarchical clustering: agglomerative (bottom-up) and divisive (top-down). The former starts at the bottom (i.e., each observation in one different cluster) and at each level recursively merges a selected pair of clusters into a single cluster. The latter starts at the top (i.e., all observations in one cluster) and at each level recursively splits one of the existing clusters at that level into two new clusters. The choice of the intergroup dissimilarity metric, like single linkage, complete linkage, centroid linkage, average linkage gives life to different versions of agglomerative methods. Ward (1963) [49] considered hierarchical clustering procedures based on minimizing the loss of information from joining two groups. Finally, model-based clustering assumes that the data in each cluster is generated from a given probabilistic distribution and the combined data stems from a convex combination of these distributions. The number of clusters M has to be determined. For k-means and hierarchical clustering, this can be done based on many indices [11] like the silhouette width [39], the gap statistic [43] and the Dunn index [18]; whereas, for model-based clustering, information criteria, such as the Akaike information criterion (AIC), Bayesian information criterion (BIC) and likelihood (ICL) could be used. But, analysis of raw data through classical multivariate techniques has several problems, because of the high number of evaluation points and the strong correlation. Furthermore, raw-data clustering approach has the drawback of losing the functional aspect of the data and is not suited for curves observed at different evaluation points. So, we will not take into account this method for our studies.

Filtering methods

Filtering methods are based on the reconstruction of the functional observations X_i from the discrete points X_{ij}. The most common approach [36] is to assume that the functional observations are embedded in a finite dimensional functional space spanned by a finite set of basis functions. Each Xi can be written as

$$X_i(t) = \sum_{k=1}^{K} c'_{ik}\Phi_k(t) = c'_i\Phi(t), t \in \mathcal{T}, i = 1, \ldots, n \tag{2.79}$$

where $\Phi = (\Phi_1, ..., \Phi_K)'$ is the vector of basis functions that span the K-dimensional subset of $L^2(\mathcal{T})$, $c_i = (c_{i1}, ..., c_{iK})'$ is the K-dimensional coefficient vector. The basis functions Φ_j can be either pre-specified, e.g. B-spline [14] and Fourier [36] or data-adaptive, e.g. obtained using functional principal component analysis (FPCA) [24]. In case of pre-specified basis functions, if the X_{ij} are observed with measurement error, then the coefficient vector c_i is usually estimated as $\hat{c}_i$ via penalized least-squares, even though standard least-squares could be used as well [36], that is

$$\hat{c}_i = \underset{c_i \in R^K}{\operatorname{argmin}} \sum_{j=1}^{m_i} (X_{ij} - c'_i\Phi(t_{ij})^2 + \lambda \int_{\mathcal{T}} [D^2 c'_i\Phi(t)]^2 dt \tag{2.80}$$

where D^2 is the second order differential operator and $\lambda > 0$ is a smoothing parameter. The reconstructed functional observation is

$$\hat{X}_i^{PS}(t) = \hat{c}'_i\Phi(t), t \in \mathcal{T}, i = 1, ..., n \tag{2.81}$$

The FPCA provides a data-adaptive basis to obtain the functional data. The functional observations are reconstructed, for $i = 1, ..., n$, as

$$\hat{X}_i^{DA}(t) = \sum_{l=1}^{L} \xi_{il}\psi_l(t) = \xi'_i\psi(t), t \in \mathcal{T}, i = 1, ..., n \tag{2.82}$$

where $\xi_i = (\xi_{i1}, ..., \xi_{iL})'$ is the vector of principal component scores defined as $\xi_{il} = \int_{\mathcal{T}} \psi(t)X_i(t)dt$, and $\psi = (\psi_1, ..., \psi_L)'$ is the vector whose elements are weight functions referred to as principal components. Principal components are defined by

an iterative algorithm which at each step finds the weight function that maximizes the mean square of the scores, or their sample variance, that is

$$\psi_l = \underset{\psi}{\operatorname{argmax}} \sum_{i=1}^{n} \xi_{il}^2 = \sum_{i=1}^{n} (\int_{\mathcal{T}} \psi(t) X_i(t) dt)^2, l = 1..., L \tag{2.83}$$

under the constraints: $\int_{\mathcal{T}} \psi_l(t)^2 dt = 1$ and $\int_{\mathcal{T}} \psi_i(t)\psi_j(t)dt = 0$, for $i \neq j$.

By reconstructing of the functional observations we can reduce the dimensionality of the data by summarizing each curve through a finite set of parameters, that is $\hat{c}_i$ or ξ_i depending on whether basis functions used are pre-specified or data-adaptive. Then, the finite set of parameters are clusterized by means of standard multivariate clustering techniques, such as k-means, hierarchical clustering or model-based clustering. The choice of the number M of clusters can be made by the use of several indices [11].

Adaptive methods

Adaptive methods relies on a finite dimensional representation of the functional data through basis functions as filtering approaches, with the exception that the basis expansion coefficients are treated as random variables with cluster-specific probability distributions. Similarly to the filtering approaches, if the functional observation X_i belongs to the mth cluster among the M clusters, it is modeled through basis functions as

$$X_i(t) = \eta'_{im}\Phi(t), t \in \mathcal{T}, i = 1, ..., n \tag{2.84}$$

where $\Phi = (\Phi_1, ..., \Phi_K)'$ are natural cubic splines, and η_{ik} is a vector of spline normal random coefficients defined as

$$\eta_{im} = \mu_m + \gamma_i \tag{2.85}$$

with μ_m the coefficient vector of the m-th cluster mean, and $\gamma_i \sim N(0, \Gamma)$ the subject-specific random effects for the i-th curve. Then, the vector of discretized values $X_i = (X_{i1}, ..., X_{im_i})'$ is modelled as

$$X_i = S_i(\mu_m + \gamma_i) + \epsilon_i \tag{2.86}$$

where $S_i = (\Phi(t_{i1}), ..., \Phi(t_{im_i}))'$ is the realization matrix of the vector Φ, and $\epsilon_i \sim N(0, R)$ is the measurement error random vector. The covariance matrix R is usually set equal $\sigma^2 I_{m_i}$, where I_{m_i} is the size m_i identity matrix. The unknown parameters $\mu_m,\ m = 1, ..., M, \mathcal{T}$ and σ are estimated by maximizing the mixture likelihood, where the cluster membership vector is modeled as a multinomial random variable with parameters $(\pi_1, ..., \pi_M)$, with π_m the probability of an observation to belong to the m-th cluster. Thus, the mixture likelihood is defined as

$$L(\mu_1, ..., \mu_M, \Gamma, \sigma, \pi_1, ..., \pi_M) = \prod_{i=1}^{N} \sum_{m=1}^{M} \pi_m f_m(X_i) \tag{2.87}$$

where $f_m(X_i)$ is the conditional density function of X_i belonging to the m-th cluster, that is $X_i|m \sim N(S_i\mu_m, \Sigma_i)$, with $\Sigma_i = \sigma^2 I_{m_i} + S_i \Gamma S_i'$. The maximization is often carried out by means of the expected maximization (EM) algorithm. After the estimate of the unknown parameters, each curve X_i is assigned to the cluster whose estimated posterior probability of cluster membership $\pi_{m|i} = \hat{f}_m(X_i)\hat{\pi}_m / \sum_{j=1}^{M} \hat{f}_j(X_i)\hat{\pi}_j$ is maximum. Moreover, the cluster mean coefficients μ_m could be further optimally parameterized to produce useful low-dimensional representations of the curves [28]. Information criteria, such as AIC and BIC, are used to select the number M of clusters and the basis dimension K [28]. The use of spline basis has two main drawbacks: (i) they are inappropriate when dealing with functions that show peaks and irregularities, (ii) they require heavy computational efforts and so are not suitable to represent high dimensional data. For these reason, we will not take into account this method for our studies.

Distance-based methods

Distance-based method is the functional extension of classical geometric clustering algorithm to functional data, such as k-means [13] and hierarchical clustering [36], that basically rely on the definition of proximity or dissimilarity among observations. Therefore, we have to introduce an appropriate functional measure of proximity or dissimilarity. We will use the following measure of proximity between the curves X_i

and X_j

$$d_l(X_i, X_j) = \left(\int_{\mathcal{T}} \left(X_i^{(l)} - X_j^{(l)} \right)^2 dt \right)^{1/2} \tag{2.88}$$

where $X(l)$ denotes the l-th derivative of X. In this case the number of clusters could be suitably chosen through the silhouette index [39].

Chapter 3

Application of functional clustering of spot welding dynamic resistance curves

The thesis activity is based on a large data set pertaining to spot welds on Maserati Levante's body collected during laboratory test. At first, only a small part has been made available. A first analysis has been conducted with this part of the data set with the aim of finding clusters of similar welding points. Then, when the remaining part of the data has been made available the analysis have been conducted and benefited from the conclusion of the analysis already drawn.

An initial distinction of the spot welds in 22 groups was made according to their own technological characteristics, such as sheet thickness, number of sheets, sheet material and number of current pulses. The analysis is based on the point belonging to the group 6 that is characterized by two galvanized sheets of the same thickness (0.8 mm), but different material (FEE340F and FEE220BH), and one current pulse. This group consists of 25 spot names collocated on the left side of the Maserati Levante's body, with the last 2 spot names (53702 and 53703) not always presented on the body. The welding schedule is not the same for all the spot names. In fact, the weld time is not the same for welding points of the same spot name. The process

parameters, current intensity, electrical resistance, voltage and power, have been sampled with a sampling period of 1 ms for all the welding time. The voltage among the electrode tips is measured using dressed copper wires attached to the electrodes, the dressing avoids the voltage induction in them. These are checked to ensure their integrity at the beginning of every welding cycle. The welding current is measured using an air-core toroid in the primary circuit of the welder. The sampled values of current and voltage are used to evaluate resistance and power according respectively with the first Ohm's law and electric power definition. These values are used to plot the profiles of the dynamic resistance curves, as shown in Figure 3.1.

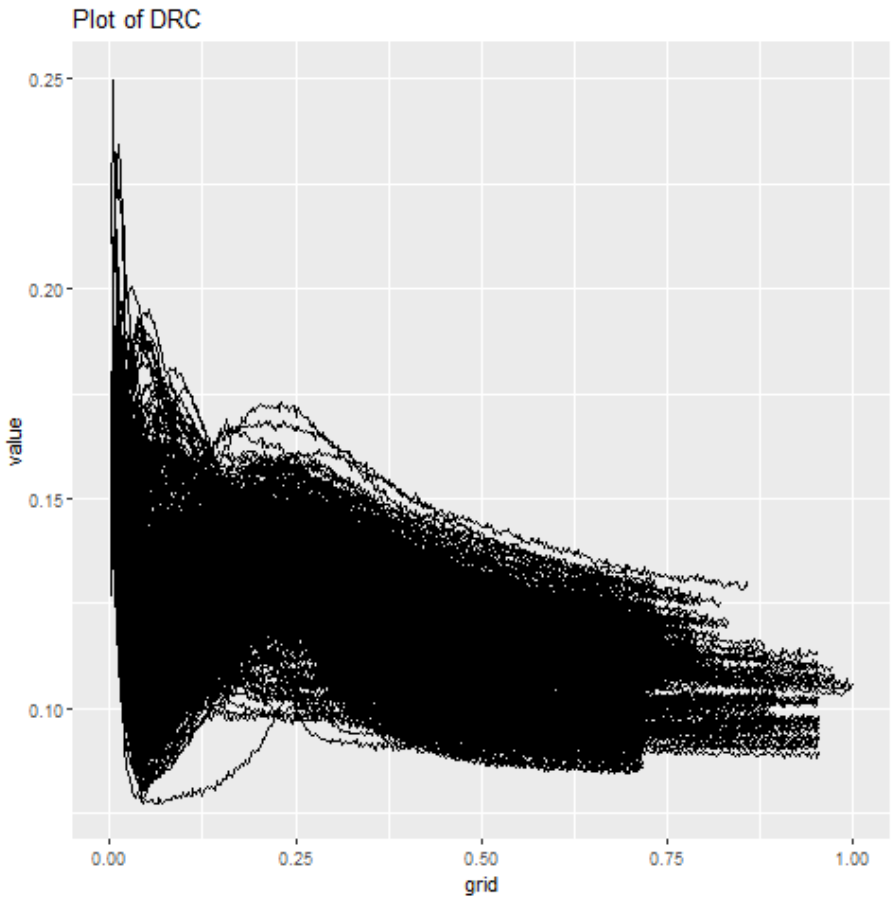

Figure 3.1: Dynamic resistance curves

The first part of data set is made by 1210 welding points. Four of these points are been deleted from the data set, because they were considered as outliers. In fact, they present some sampled values that are larger than two or three orders of magnitude with respect to the rest of the sampled values. Then, the dynamic

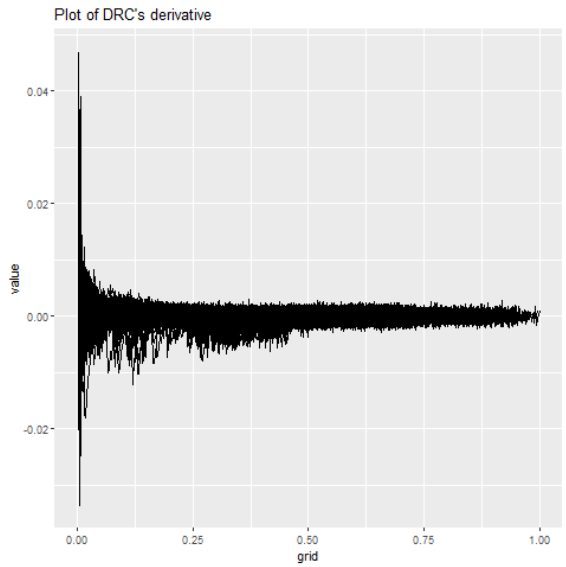

Figure 3.2: Dynamic resistance curves' derivative

resistance curves were divided into two groups, according to their welding time. The former is characterized by a welding time equal to 220 ms, the latter by a welding time equal to 300 ms. These two welding time values have been selected because most of the curves ends at those values, while the others have a slightly larger weld time. So, the curves with a welding time larger than the group threshold have been cut at threshold values. This operation was needed because, the different duration of the welding time does not influence the clustering process. In fact, without taking off these curves, the clustering methods produce clusters that differ form each other only for the welding time. In this way the cluster analysis would not have brought some useful information about the process. Furthermore, this cutting operation is also justified by the fact that the final part of the dynamic resistance curve does not bring the same amount of information than the initial and the middle parts. In fact, it is possible to observe from Figure 3.1 and by plotting the first derivative of the DRC, as shown in Figure 3.2, that the curve variation is very small in the final part and more important in the initial and middle parts. After this division in two groups, other three curves belonging to the 220 ms group have been removed from the data set because of their very short welding time.

After these preliminary actions, the two groups have been functional-clustered

separately in order to discover new information about the process nature. Therefore, the various functional clustering methods illustrated in Chapter 2 have been applied to the two groups, and the most satisfactory results from a technological point of view have been found by applying the distance-based method. In fact, by giving in input [2,4] as the interval in which to search for the optimal number of clusters, the algorithm has returned for the 220 ms group 3 clusters, as shown in Figure 3.3, that might represent, because of their shapes, 3 different welding quality levels according to the literature. It is possible to associate cluster 1 with small nugget welding point, cluster 2 with cold weld, and cluster 3 with good weld. For the 300ms group, by giving in input [2,4] as the interval in which to search for the optimal number of clusters, the clustering procedure gave back 4 clusters, as shown in Figure 3.4. Clusters 1 and 4 might refer to good welds, while clusters 2 and 3 might refer to welds affected by spatter.

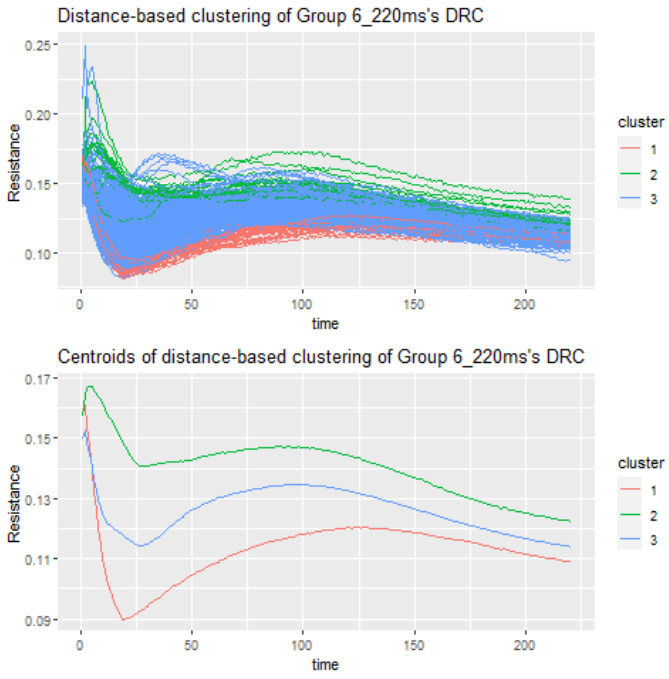

Figure 3.3: Distance based clustering of 220ms group. The panel above shoes the individual curves coloured by cluster assignment, while the panel below shows the clusters centroids.

These observations can be done by looking to the centroids, that show the average profile of the respective clusters. However, by looking at the single curves, it is possible to note that not all have the shapes that are similar to the respective

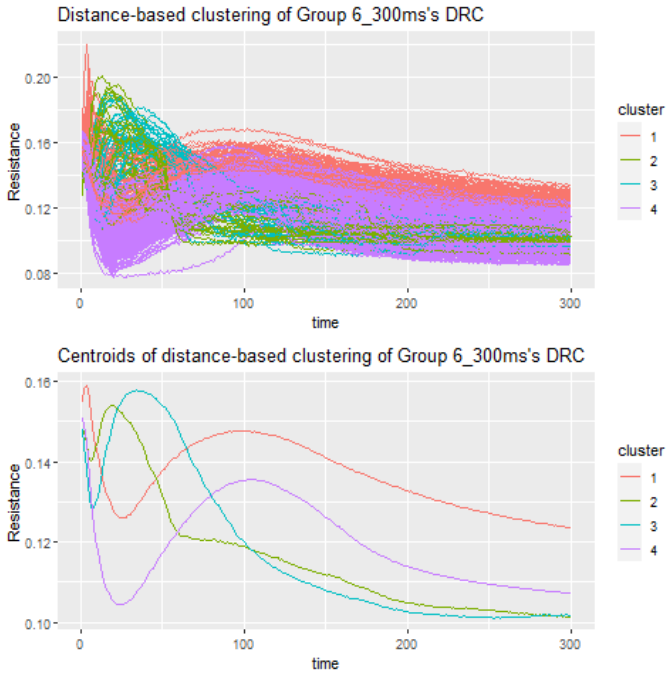

Figure 3.4: Distance based clustering of 300ms' group. The panel above shoes the individual curves coloured by cluster assignment, while the panel below shows the clusters centroids.

centroids. In fact, the lack of a response variable is a limit in this case, where we have an unsupervised learning problem, so this activity may be not adequate for detecting a cluster of only spatter welds. For a better interpretation of these results, contingency tables have been constructed. In this way, it is possible to observe how the clusters are structured and, thus, if there are some clusters built up only by some few spot names. For the 220ms group, as shown in Table 3.1, cluster 1 is composed by only two spot names (53003 and 530023) and cluster 2 by only seven spot names, while the third cluster is composed by welding point of all spot names. Considering that the process has been pointed by experts as a reference good process, this table strengthens the technological interpretation made before. In fact, by looking at the intra spot name frequency, Table 3.2 shows most of the welding points for all spot names to fall into cluster 3. Thus, most of the welding points are good weld. Similar results can be observed for the 300ms group. In fact, as shown in Table 3.3, only cluster 4 is build up by all spot names, while the other three clusters are composed by only few spot names. Furthermore, as shown in Table 3.4, most of the welding points for all spot names fall into cluster 4, which might represent good welds.

Table 3.1: Contingency tables for 220ms' clusters

	1	2	3	TOTAL
53003_0_00	3	1	26	30
53009_0_00	0	3	11	14
53021_0_00	0	0	44	44
53023_0_00	30	0	25	55
53029_0_00	0	8	37	45
53035_0_00	0	2	30	32
53037_0_00	0	0	46	46
53039_0_00	0	0	17	17
53041_0_00	0	0	33	33
53043_0_00	0	0	21	21
53045_0_00	0	1	30	31
53049_0_00	0	0	40	40
53051_0_00	0	2	42	44
53056_0_00	0	12	31	43

Table 3.2: Contingency frequency tables for 220ms' group

	1	2	3
53003_0_00	0.1000000	0.0333333	0.8666667
53009_0_00	0.0000000	0.2142857	0.7857143
53021_0_00	0.0000000	0.0000000	1.0000000
53023_0_00	0.5454545	0.0000000	0.4545455
53029_0_00	0.0000000	0.1777778	0.8222222
53035_0_00	0.0000000	0.0625000	0.9375000
53037_0_00	0.0000000	0.0000000	1.0000000
53039_0_00	0.0000000	0.0000000	1.0000000
53041_0_00	0.0000000	0.0000000	1.0000000
53043_0_00	0.0000000	0.0000000	1.0000000
53045_0_00	0.0000000	0.0322581	0.9677419
53049_0_00	0.0000000	0.0000000	1.0000000
53051_0_00	0.0000000	0.0454545	0.9545455
53056_0_00	0.0000000	0.2790698	0.7209302

Table 3.3: Contingency tables for 300ms' group

	1	2	3	4	TOTALE
53001_0_00	12	2	2	34	50
53003_0_00	0	4	0	15	19
53007_0_00	9	1	2	27	39
53009_0_00	2	0	0	10	12
53011_0_00	9	5	15	47	76
53013_0_00	3	10	2	44	59
53015_0_00	3	2	10	45	60
53019_0_00	3	0	2	49	54
53021_0_00	0	1	0	8	9
53025_0_00	14	1	2	33	50
53027_0_00	11	0	0	37	48
53029_0_00	0	0	1	1	2
53033_0_00	4	8	1	45	58
53035_0_00	0	1	1	16	18
53037_0_00	0	0	0	4	4
53039_0_00	0	0	1	33	34
53041_0_00	0	3	1	16	20
53043_0_00	0	1	1	28	30
53045_0_00	0	0	0	19	19
53049_0_00	0	0	1	13	14
53051_0_00	1	0	1	7	9
53056_0_00	0	1	1	8	10
53702_0_00	1	0	1	6	8
53703_0_00	2	1	0	3	6

Table 3.4: Contingency frequency tables for 300ms' group

	1	2	3	4
53001_0_00	0.2400000	0.0400000	0.0400000	0.6800000
53003_0_00	0.0000000	0.2105263	0.0000000	0.7894737
53007_0_00	0.2307692	0.0256410	0.0512821	0.6923077
53009_0_00	0.1666667	0.0000000	0.0000000	0.8333333
53011_0_00	0.1184211	0.0657895	0.1973684	0.6184211
53013_0_00	0.0508475	0.1694915	0.0338983	0.7457627
53015_0_00	0.0500000	0.0333333	0.1666667	0.7500000
53019_0_00	0.0555556	0.0000000	0.0370370	0.9074074
53021_0_00	0.0000000	0.1111111	0.0000000	0.8888889
53025_0_00	0.2800000	0.0200000	0.0400000	0.6600000
53027_0_00	0.2291667	0.0000000	0.0000000	0.7708333
53029_0_00	0.0000000	0.0000000	0.5000000	0.5000000
53033_0_00	0.0689655	0.1379310	0.0172414	0.7758621
53035_0_00	0.0000000	0.0555556	0.0555556	0.8888889
53037_0_00	0.0000000	0.0000000	0.0000000	1.0000000
53039_0_00	0.0000000	0.0000000	0.0294118	0.9705882
53041_0_00	0.0000000	0.1500000	0.0500000	0.8000000
53043_0_00	0.0000000	0.0333333	0.0333333	0.9333333
53045_0_00	0.0000000	0.0000000	0.0000000	1.0000000
53049_0_00	0.0000000	0.0000000	0.0714286	0.9285714
53051_0_00	0.1111111	0.0000000	0.1111111	0.7777778
53056_0_00	0.0000000	0.1000000	0.1000000	0.8000000
53702_0_00	0.1250000	0.0000000	0.1250000	0.7500000
53703_0_00	0.3333333	0.1666667	0.0000000	0.5000000

Table 3.5: Spot names of Group 1

Group 1's spotnames
53019_0_00
53021_0_00
53035_0_00
53037_0_00
53039_0_00
53041_0_00
53043_0_00
53045_0_00
53049_0_00
53051_0_00
53702_0_00

Table 3.6: Spot names of Group 2

Group 1's spotnames
53003_0_00
53009_0_00
53023_0_00
53027_0_00
53029_0_00
53056_0_00
53703_0_00

Table 3.7: Spot names of Group 3

Group 1's spotnames
53001_0_00
53007_0_00
53011_0_00
53013_0_00
53015_0_00
53025_0_00
53033_0_00

Three new groups of spot names have been established according to the frequency of the spot names in the clusters, i.e., spot names of both groups roughly belonging to one cluster have been grouped together. Tables 3.5, 3.6 and 3.7 show the group conformation. Furthermore, the plots of these curves of the three groups are shown in Figures 3.5, 3.6, 3.7 and it is important to underline that group profiles have been cut at 240ms, 220ms, 300ms respectively, for groups 1, 2 and 3. Moreover, note that profiles inside the same group appear to be more similar to each other than those of the other groups. It is possible to locate these three groups on the Maserati Levante's body, as shown in Figure 3.8.

After this grouping operation, the three groups have been clustered in order to obtain subgroups corresponding to a different weld quality more characteristic for the various type of dynamic resistance curves. For this purpose, only distance based, filtering B-spline and filtering FPCA clustering methods have been used. The results point out that there is no cluster made only of spatter welding points and there is no cluster with good welds only, except for what has been found by applying filtering

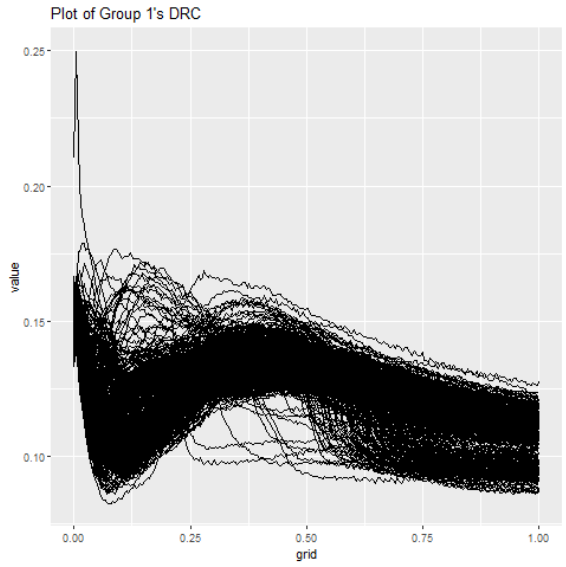

Figure 3.5: Group 1's DRCs

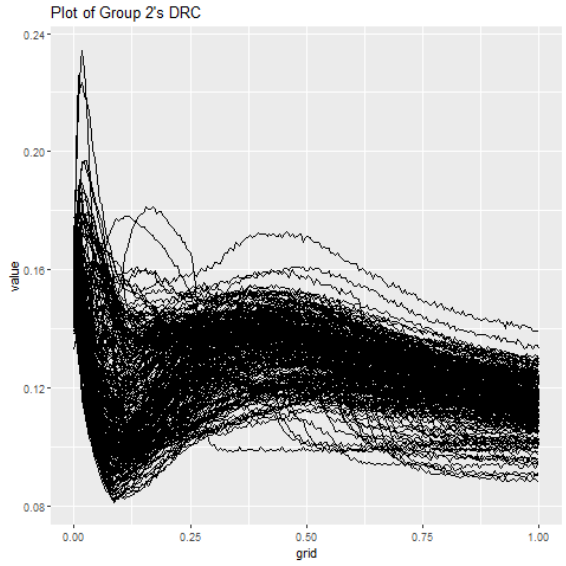

Figure 3.6: Group 2's DRCs

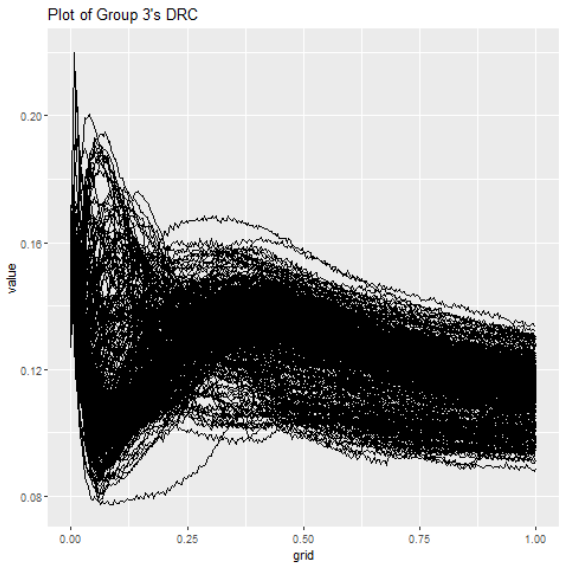

Figure 3.7: Group 3's DRCs

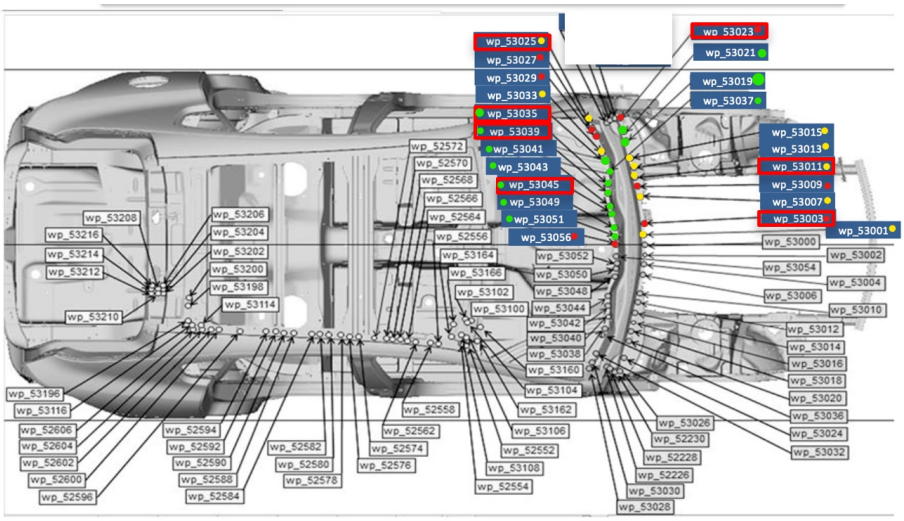

Figure 3.8: Maserati Levante's body. Green: Group 1; Red: Group 2; Yellow: Group 3

B-spline hierarchical (group 1) and filtering FPCA model-based (group 2). Some of these results are shown in Figures 3.9 and 3.10.

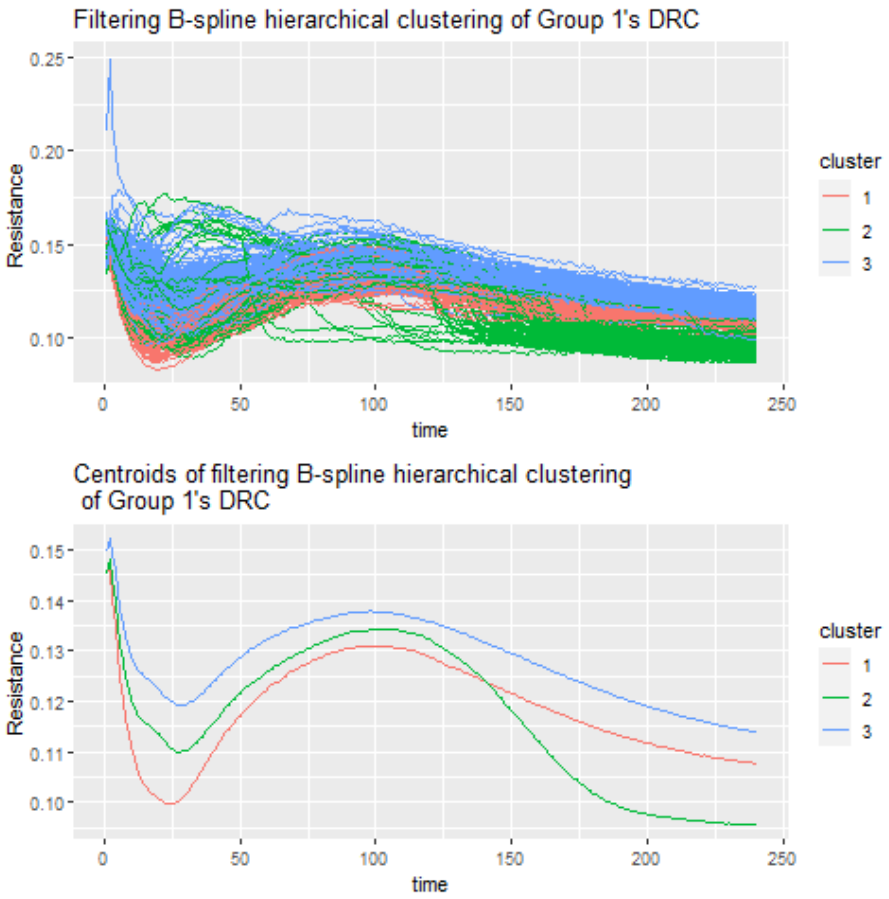

Figure 3.9: Filtering B-spline hierarchical of group 1

With the provision of the remaining part of the data set, 38883 welding point data were available. 67 points have been removed from the data set based on engineering and statistical considerations. Thanks to the huge amount of data, it was possible to cluster curves grouped by spot name. In this way, detecting clusters of only spatter welding might be easier, because we are clustering separately DRCs for each spot name, then curves are more similar to each other. Before clustering, the dynamic resistance curves of each spot name have been cut at the their common welding time. Figure 3.11 shows the threshold values for all the spot names.

For this analysis distance-based and filtering methods have been used. Other methods (Raw data and Adaptive) have not been deemed as adequate. The former

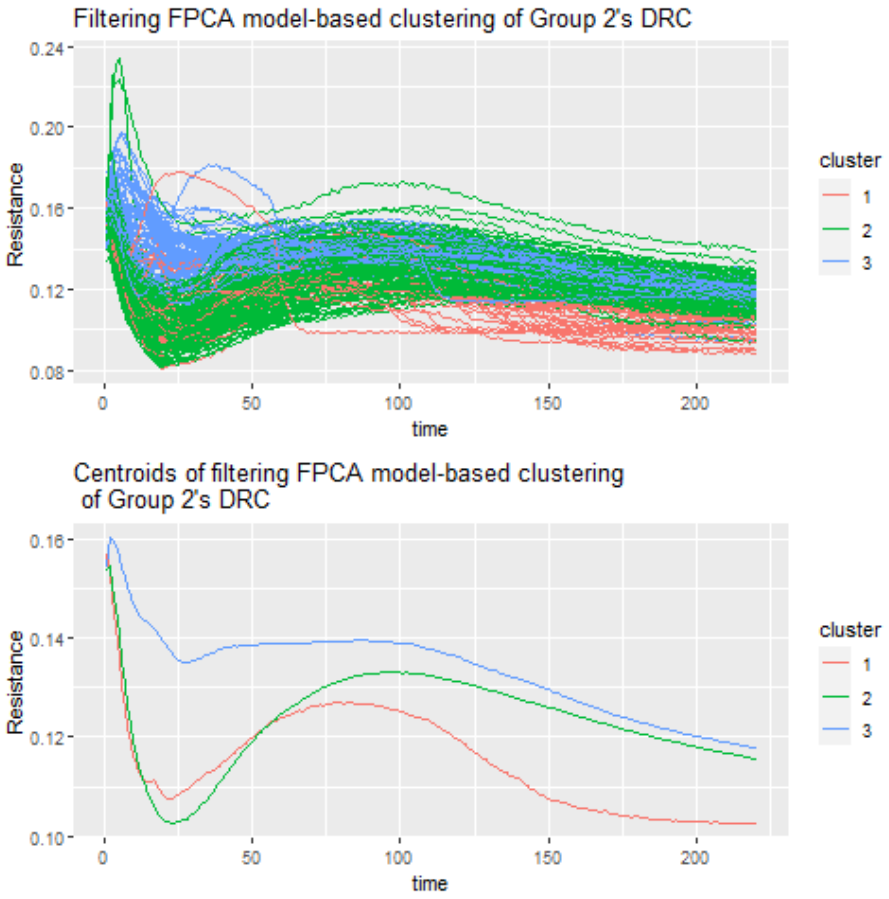

Figure 3.10: Filtering FPCA model-based of group 2

SPOTNAME	WELD TIME [ms]
53001	300
53003	240
53007	320
53009	280
53011	300
53013	320
53015	320
53019	300
53021	240
53023	220
53025	320
53027	320
53029	240
53033	340
53035	240
53037	240
53039	240
53041	240
53043	240
53045	240
53049	240
53051	240
53056	240
53702	320
53703	320

Figure 3.11: Spot names welding time thresholds

does not take in account the functional nature of the data, the latter requires long computational times. For distance-based clustering method, the required number of clusters has been set equal to 2, to obtain the separation between spattered and good welds, while for filtering methods the optimal number of cluster has been chosen among 2, 3 and 4 by means of several indices mentioned in Chapter 2. Some of the results of these clustering methods are shown in Figures 3.12, 3.13 3.14 3.15. It is possible to see that some clusters contain only spatter welds, while others contain both spatter and good welds.

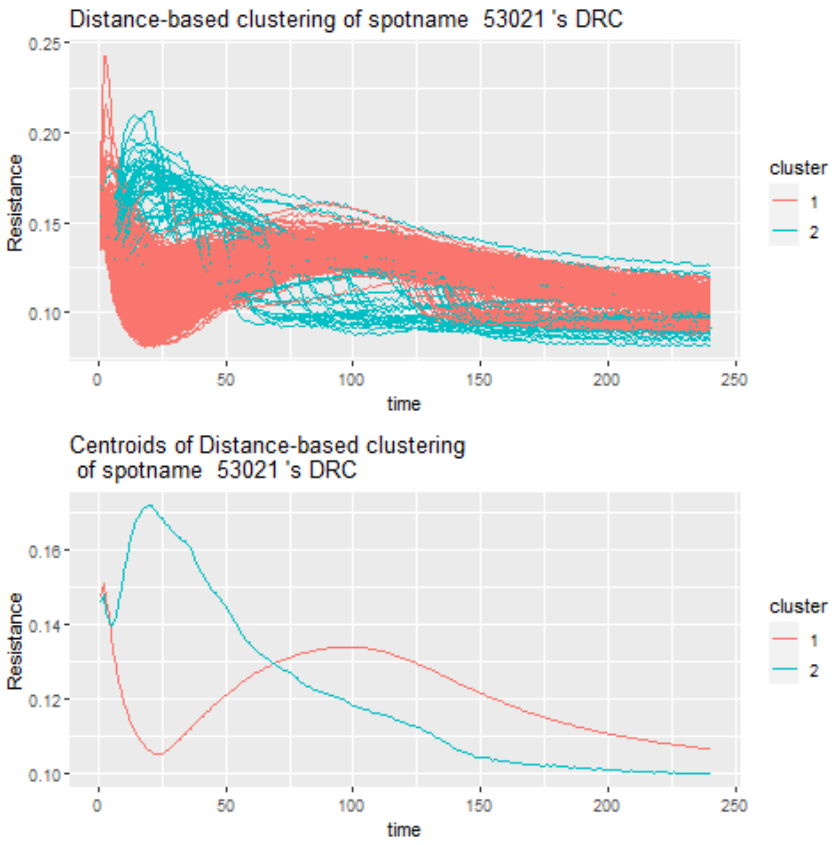

Figure 3.12: Distance based clustering of Spotname 53021's DRC

Also in this case, contingency tables have been constructed in order to visualize the distribution in the clusters of the many spot names' welding points. Considering that the clusters of only spatter welds have not been found immediately for all spot names, a hierarchical approach for the filtering methods has been carried out in order to divide the found clusters in sub-clusters. Each cluster has been thus clustered again until it was possible to find a technological interpretation. Some results are

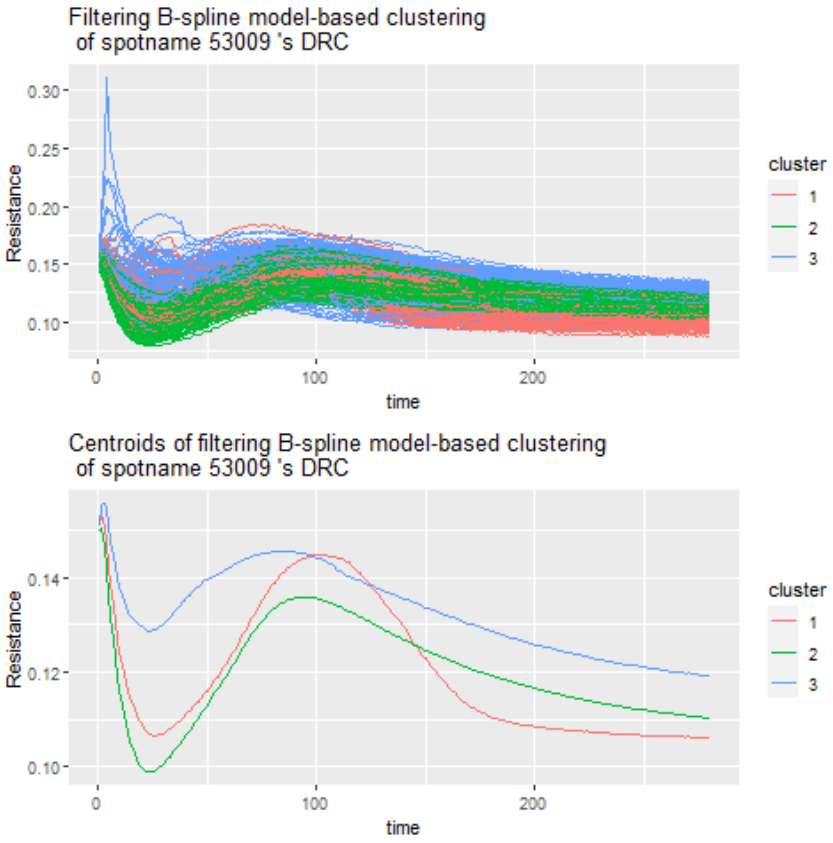

Figure 3.13: Filtering B-spline model-based clustering of spotname 53009's DRC

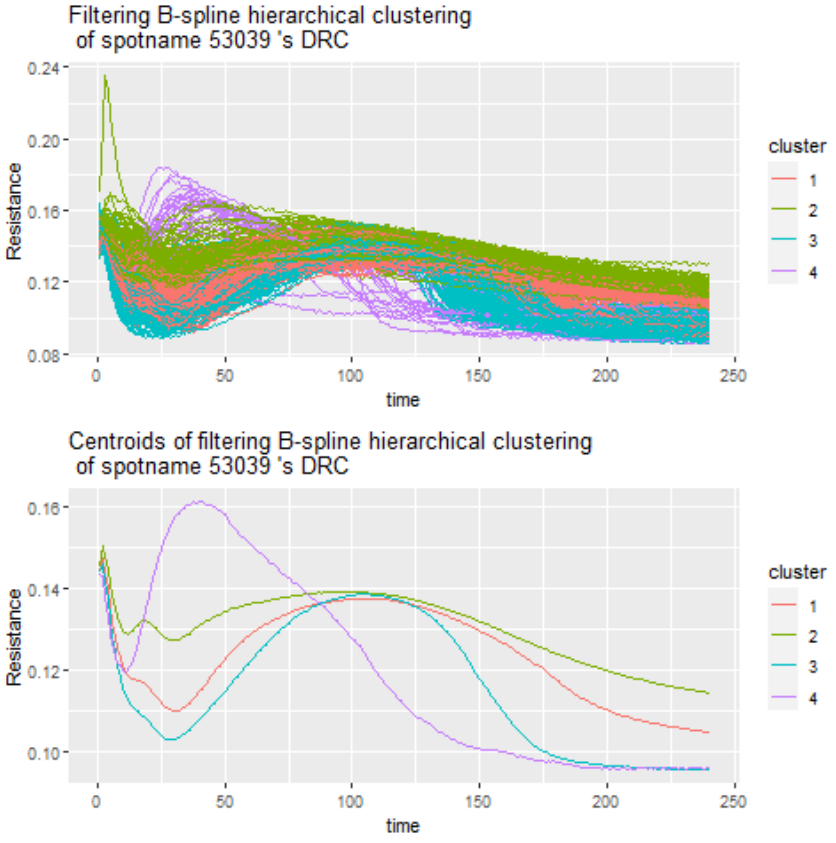

Figure 3.14: Filtering B-spline hierarchical clustering of spotname 53039's DRC

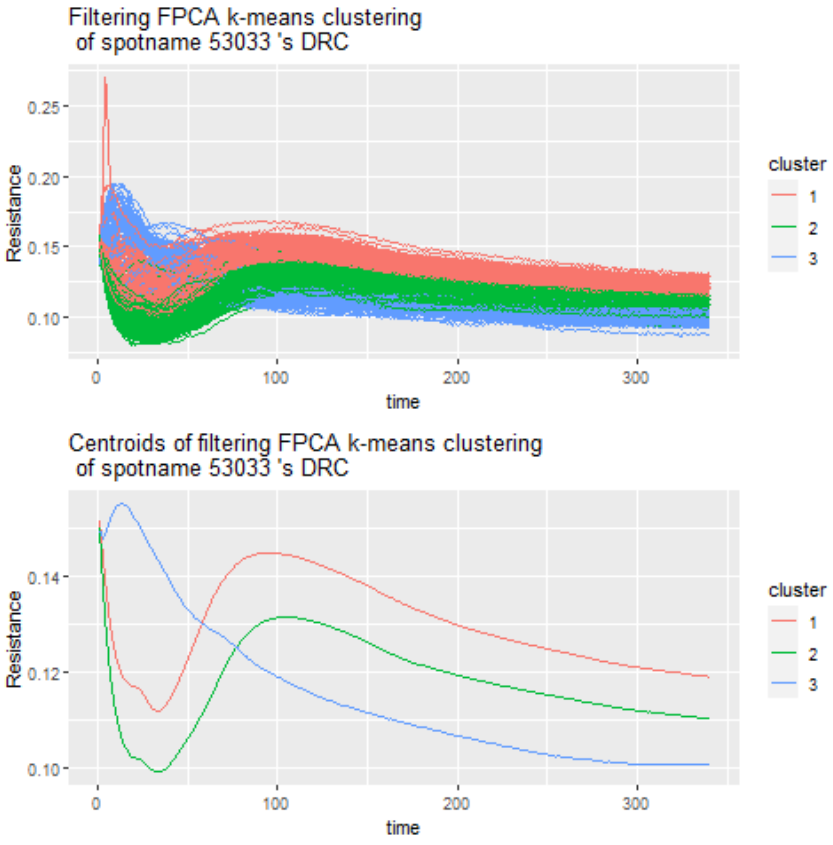

Figure 3.15: Filtering FPCA k-means profili-centroidi of spotname 53033's DRC

shown in the following Figures 3.16 and 3.17.

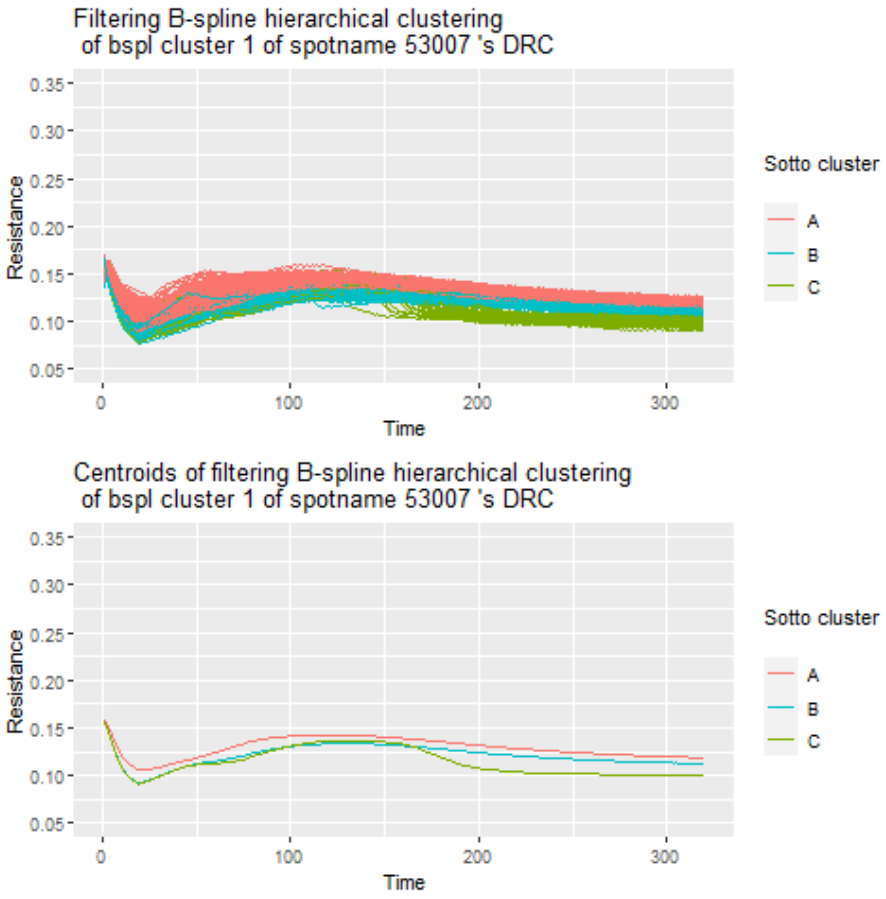

Figure 3.16: Filtering B-spline profili-centroidi of bspl cluster 1 of spotname 53007 's DRC

After this hierarchical clustering, contingency tables have been constructed again. By visualizing them, it is possible to note that most of the welding points fall in one or more clusters, that represent good weld, while a small amount of them falls in to cluster representing spatter welds.

Conclusions and future development

With this first analysis through functional data analysis approach, the spot names of group 6 have been grouped in three new subgroups, not only from a technological point of view, but also on the basis of how they are divided into the various clusters. A possible future activity might be a deeper analysis of the welding process of these

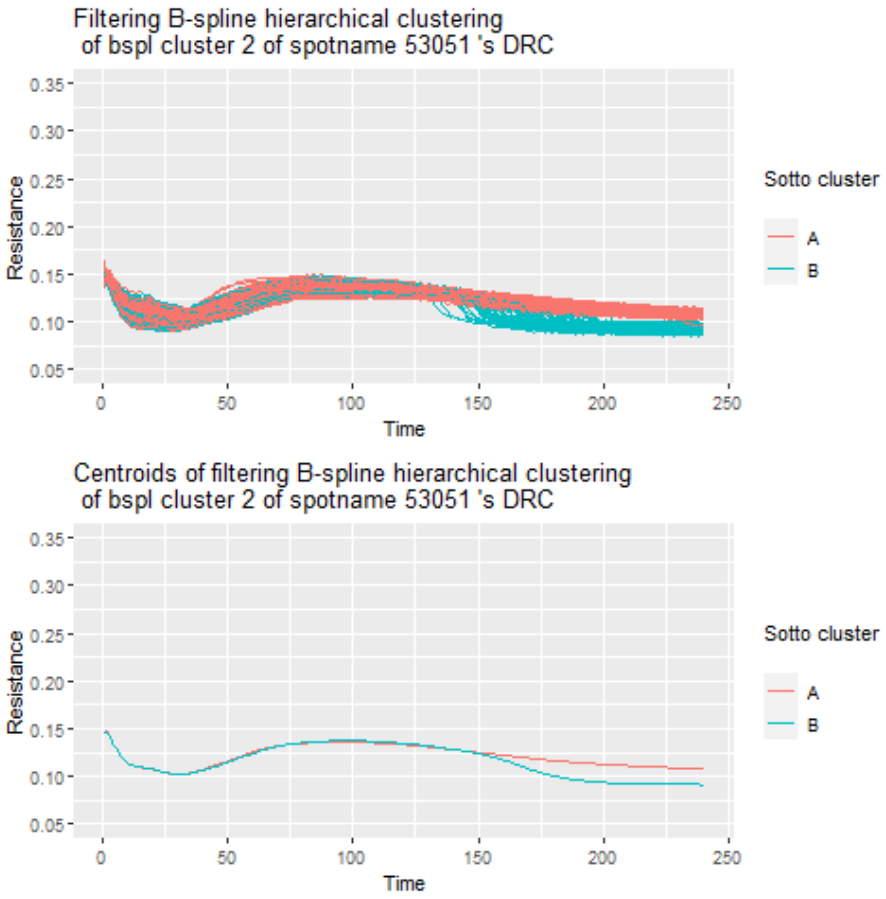

Figure 3.17: Filtering B-spline profili-centroidi of bspl cluster 2 of spotname 53051 's DRC

three groups aimed at the discovering of the technological reasons of the clustering with the use of more interpretable clustering methods as that recently published in [9].

Instead, the second analysis led to the creation of clusters representative of different welding qualities for each spot name. In this way, a smarter quality control can be conducted. In fact, having identified clusters corresponding to a profiles with a higher probability of being spattered, the ultrasonic control can be restricted to the profiles that fall into those specific clusters. This may guarantee a huge savings in the quality control activities. In order to do this, a control charting scheme should to be designed by calculating a dissimilarity index between a new profile and the clusters, a new profile can be assigned to the closer cluster. Furthermore, by imposing control limits, DRCs that present too high dissimilarity could be marked as produced by an out-of-control process.

Finally, by acquiring ultrasonic quality data, it will be possible to assign to each cluster the actual welding quality and a functional regression model could be built up for mapping a more accurate relationship between welding process data and the final quality.

All the plots and tables are available on this GitHub repository:

https://github.com/GianlucaNapoli/immagini-tesi.git

Acknowledgements

The author is extremely grateful to CRF (www.crf.it) Engineer Gianmarco Genchi for his technological insights in the interpretation of the results and to Engineer Alessandro Agizza for his help with R code.

Bibliography

[1] M. Acebes, R. D. de Molina, I. Gauna, N. Thorpe, and J. C. Guerro. Development of an automated ultrasonic inspection device for quality control of spot welds. In *19th World Conference on Non-Destructive Testing*, 2016.

[2] D. W. Adams, C. D. Summerville, B. M. Voss, J. Jeswiet, and M. C. Doolan. Correlating variations in the dynamic resistance signature to weld strength in resistance spot welding using principal component analysis. *Journal of Manufacturing Science and Engineering*, 139(4), 2017.

[3] R. M. Alliance. Resistance welding manual, 2003.

[4] A. Ambroziak, R. G. Maev, M. Korzeniowski, and P. Kustroń. Ultrasonic quality control methods for spot-welded joints. *Welding International*, 25(12):927–932, 2011.

[5] G. Avezzani. Prontuario per il tecnico di saldatura a punti, 1993.

[6] J. Buckley and R. Servent. Improvements in ultrasonic inspection of resistance spot welds. *Insight-Non-Destructive Testing and Condition Monitoring*, 51(2): 73–77, 2009.

[7] R. M. Buffington, P. Kaminski, and E. A. Larson. Selectable focus ultrasonic transducers for diagnostic imaging. *US Patent 4*, 557:146, 12 1985.

[8] R. M. Buffington, P. Kaminski, and E. A. Larson. Selectable focus ultrasonic transducers for diagnostic imaging, Dec. 10 1985. US Patent 4,557,146.

[9] F. Centofanti, A. Lepore, and B. Palumbo. Sparse and smooth functional data clustering, 2021.

[10] H. Chang and H. Cho. A study on the shunt effect in resistance spot welding. *Welding Journal*, 69(8):308–316, 1990.

[11] M. Charrad, N. Ghazzali, V. Boiteau, and A. Niknafs. Nbclust: an r package for determining the relevant number of clusters in a data set. *Journal of statistical software*, 61:1–36, 2014.

[12] Y. Cho and S. Rhee. Experimental study of nugget formation in resistance spot welding. *Welding journal*, 82(8):195–201, 2003.

[13] J. A. Cuesta-Albertos and R. Fraiman. Impartial trimmed k-means for functional data. *Computational Statistics & Data Analysis*, 51(10):4864–4877, 2007.

[14] C. De Boor and C. De Boor. *A practical guide to splines*, volume 27. springer-verlag New York, 1978.

[15] A. Dennison, D. Toncich, and S. Masood. Control and process-based optimisation of spot-welding in manufacturing systems. *The International Journal of Advanced Manufacturing Technology*, 13(4):256–263, 1997.

[16] D. Dickinson, J. Franklin, A. Stanya, et al. Characterization of spot welding behavior by dynamic electrical parameter monitoring. *Welding Journal*, 59(6): 170, 1980.

[17] D. Dickinson, J. Franklin, A. Stanya, et al. Characterization of spot welding behavior by dynamic electrical parameter monitoring. *Welding Journal*, 59(6): 170, 1980.

[18] J. C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. 1973.

[19] M. El-Banna, D. Filev, and R. B. Chinnam. Online qualitative nugget classification by using a linear vector quantization neural network for resistance spot welding. *The International Journal of Advanced Manufacturing Technology*, 36 (3-4):237–248, 2008.

[20] B. Everitt, S. Landau, M. Leese, and D. Stahl. *Cluster Analysis*. John Wiley & Sons, 2011.

[21] F. Ferraty and P. Vieu. *Nonparametric functional data analysis: theory and practice*. Springer Science & Business Media, 2006.

[22] S. Z. Gavidel, S. Lu, and J. L. Rickli. Performance analysis and comparison of machine learning algorithms for predicting nugget width of resistance spot welding joints. *The International Journal of Advanced Manufacturing Technology*, 105(9):3779–3796, 2019.

[23] G. Genutis, E. Jasiūnienė, and R. Sanderson. An algorithm for the estimation of the quality of the spot welds. *Russian Journal of Nondestructive Testing*, 50 (6):335–342, 2014.

[24] P. Hall and M. Hosseini-Nasab. On properties of functional principal components analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):109–126, 2006.

[25] O. Ighodaro, E. Biro, and Y. Zhou. Study and applications of dynamic resistance profiles during resistance spot welding of coated hot-stamping steels. *Metallurgical and Materials Transactions A*, 48:1–14, 12 2016. doi: 10.1007/s11661-016-3899-3.

[26] M. Jafari Vardanjani, A. Araee, J. Senkara, J. Jakubowski, and J. Godek. Theoretical analysis of shunting effect in resistance spot welding (rsw) of aa2219. *Journal of the Chinese Institute of Engineers*, 39(8):907–918, 2016.

[27] e. a. James, Gareth. *An introduction to statistical learning*, volume 112. New York: springer, 2013.

[28] G. M. James and C. A. Sugar. Clustering for sparsely sampled functional data. *Journal of the American Statistical Association*, 98(462):397–408, 2003.

[29] M. Kimchi and D. H. Phillips. Resistance spot welding: fundamentals and applications for the automotive industry. *Synthesis Lectures on Mechanical Engineering*, 1(2):115, 2017.

[30] P. Kokoszka and M. Reimherr. *Introduction to functional data analysis*. Chapman and Hall/CRC, 2017.

[31] C. Lane, C. Sorensen, G. Hunter, S. Gedeon, and T. Eagar. Cinematography of resistance spot welding of galvanized steel sheet. *Weld. J.*, 66(9):260, 1987.

[32] Y. Li, B. Wang, Q. Shen, M. Lou, and H. Zhang. Shunting effect in resistance spot welding steels—part 2: theoretical analysis. *Welding Journal*, 92(8):231–238, 2013.

[33] Ó. Martín, M. Pereda, J. I. Santos, and J. M. Galán. Assessment of resistance spot welding quality based on ultrasonic testing and tree-based techniques. *Journal of Materials Processing Technology*, 214(11):2478–2487, 2014.

[34] H. P. Moers. Ultrasonic testing as a means for quality assurance in resistance spot welding'. *Krautkrämer, Sonderdruck*, 297, 1999.

[35] S. Rabinovich, K. Jassby, O. Livni, and R. Aharoni. Progress in spotweld test and classification tools. In *15th World Conference on Nondestructive Testing. Roma*, 2000.

[36] J. Ramsay and B. Silverman. *Functional data analysis*. Springer, 2005.

[37] R. Raoelison, A. Fuentes, P. Rogeon, P. Carre, T. Loulou, D. Carron, and F. Dechalotte. Contact conditions on nugget development during resistance spot welding of zn coated steel sheets using rounded tip electrodes. *Journal of Materials Processing Technology*, 212(8):1663–1669, 2012.

[38] D. Roberts, J. Mason, and C. Lewis. Ultrasonic spot weld testing with automatic classification. *Science and technology of welding and joining*, 7(1):47–50, 2002.

[39] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[40] W. Roye. Ultrasonic testing of spot welds in the automotive industry. *Krautkrämer GmbH & Co. oHG, nr SD*, 298, 2003.

[41] Y. Song, L. Hua, X. Wang, B. Wang, and Y. Liu. Research on the detection model and method for evaluating spot welding quality based on ultrasonic a-scan analysis. *Journal of Nondestructive Evaluation*, 35(1):4, 2016.

[42] R. Stout and J. Koh. Detectability and significance of buried tears in welded t-joints. *Welding journal*, 66(1):11s–18s, 1987.

[43] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[44] C. L. Tsai, W. L. Dai, and D. W. Dickinson. Analysis and development of a real-time control methodology in resistance spot welding. *SAE transactions*, pages 158–176, 1991.

[45] M. Tumuluru. Resistance spot weld performance and weld failure modes for dual phase and trip steels. In *Failure Mechanisms of Advanced Welding Processes*, pages 43–64. Elsevier, 2010.

[46] B. Wang, M. Lou, Q. Shen, Y. Li, and H. Zhang. Shunting effect in resistance spot welding steels—part 1: experimental study. *Welding Journal*, 92(6):182s–189s, 2013.

[47] J.-L. Wang, J.-M. Chiou, and H.-G. Müller. Functional data analysis. *Annual Review of Statistics and Its Application*, 3:257–295, 2016.

[48] S. Wang and P. Wei. Modeling dynamic electrical resistance during resistance spot welding. *J. Heat Transfer*, 123(3):576–585, 2001.

[49] J. H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[50] G. Xu, J. Wen, C. Wang, and X. Zhang. Quality monitoring for resistance spot welding using dynamic signals. In *2009 International Conference on Mechatronics and Automation*, pages 2495–2499. IEEE, 2009.

[51] H. Zhang and J. Senkara. Resistance welding. *CRC Press*, 2012.

[52] X. Zhao, Y. Zhang, and G. Chen. Research for ultrasonic fast-identification of the stick-weld defect. In *2006 IEEE Instrumentation and Measurement Technology Conference Proceedings*, pages 81–85. IEEE, 2006.

[53] K. Zhou and L. Cai. Online nugget diameter control system for resistance spot welding. *The International Journal of Advanced Manufacturing Technology*, 68, 10 2013. doi: 10.1007/s00170-013-4886-0.

[54] K. Zhou and L. Cai. Study on effect of electrode force on resistance spot welding process. *Journal of applied physics*, 116(8), 2014.

Appendix A

R code

```
#INSTALLAZIONE E CARICAMENTO PACKAGES
###############
#installazione packages

# install.packages("wavethresh")
# if (!("curvclust" %in% installed.packages())) {
#   install.packages("D:/Gianluca/Desktop/TESI/curvclust_0.0.1 (2).tar.gz",
# repos = NULL, type="source")
# }
# install.packages("dplyr")
# install.packages("GGally")
# install.packages("ggplot2")
# install.packages("ggthemes")
# install.packages("ggvis")
# install.packages("plotly")
# install.packages("rio")
# install.packages("rmarkdown")
# install.packages("stringr")
# install.packages("tidyr")
# install.packages("curvclust")
# install.packages("tidyverse")
# install.packages("parallel")
# install.packages("fda.usc")
# install.packages("clValid")
# install.packages("funHDDC")
# install.packages("NbClust")
# install.packages("mclust")
# install.packages("factoextra")

# install.packages("ggpubr")
# install.packages("knitr")

#carico packages

library(wavethresh)
library(curvclust)
library(tidyverse)
library(parallel)
library(fda.usc)
library(clValid)
library(funHDDC)
library(NbClust)
library(mclust)
library(factoextra)
library(ggpubr)
#per latex
library(knitr)

source("D:/Gianluca/Desktop/TESI/funclustRSW-master/funclustRSW-master/fclust.R")
source("D:/Gianluca/Desktop/TESI/funclustRSW-master/funclustRSW-master/functions.R")

###############


#Prima parte data set
```

```r
#PREPARAZIONE DATI

#Carico data set
##################
data_profili <-
  read.csv("D:/Gianluca/Desktop/TESI/DATI/SCC100R01_Profili_Febbraio2021_Associazione_NonFault.csv")
scalari_associazione <-
  read.csv("D:/Gianluca/Desktop/TESI/DATI/SCC100R01_Scalari_Febbario2021_Associazione_NonFault.csv")
gruppi_spotname <- import("D:/Gianluca/Desktop/TESI/DATI/MaseratiSCC100R01SpotNameGruppo.csv")

scalari_associazione <- merge(scalari_associazione,gruppi_spotname,by = ("spotName"))
#ora abbiamo i vari spotName collegati alle varie famiglie tecnologiche
#il clustering verrà effettuato tra i profili delle varie famiglie
#Mi focalizzo sul gruppo 6

#creazione matrice con i valori di resistenza dei profili registrati
dateTime_scalari <- scalari_associazione$dateTime[scalari_associazione$Gruppo == 6]
resistance_rawdata <- data_profili %>%
  filter(id %in% dateTime_scalari) %>%
  pivot_wider(names_from = time,
              values_from = ResistanceCurve,
              id_cols = id,
              values_fill = NA)
##################

#Manipolazione dati
####################
#elimino profili anomali
resistance_rawdata <- resistance_rawdata[-c(165,371,551,619,725,750,753),]

#Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
resistance_rawdata <- t(resistance_rawdata)

#Divido le curve in due gruppi:
#il primo con le curve che sono state registrate non oltre i 300 ms (e poi tagliate a 220ms)
#il secondo con le curve che sono state registrate oltre i 300 ms (e poi tagliate a 300ms)
X_220 <- list()
X_300 <- list()
j <- 1
k <- 1
for (i in 1:ncol(resistance_rawdata)) {
  if (is.na(resistance_rawdata[301,i])) {          #301 perché la prima riga sono gli id dei profili
    X_220[[j]] <- resistance_rawdata[,i];
    j <- j+1;
  }
  else {
    X_300[[k]] <- resistance_rawdata[,i];
    k <- k+1;
  }
}

#prendo gli elementi della lista per creare le matrici, ogni elemento della lista è un profilo
#e verrà messo per colonna
X_220_matrice <- matrix(nrow=437,ncol=495)
X_300_matrice <- matrix(nrow=437,ncol=708)    #436 ms+ 1 di id

for (i in 1:708) {
  X_300_matrice[,i] <- X_300[[i]]
}
for (i in 1:495) {
  X_220_matrice[,i] <- X_220[[i]]
}

#la prima riga rappresenta gli id dei profili, quindi li setto come nomi delle colonne
colnames(X_220_matrice) <- X_220_matrice[1,]
X_220_matrice <- X_220_matrice[-1,]
storage.mode(X_220_matrice) <- "numeric"

colnames(X_300_matrice) <- X_300_matrice[1,]
X_300_matrice <- X_300_matrice[-1,]
storage.mode(X_300_matrice) <- "numeric"
####################


#INIZIO ANALISI
#creazione dei profili
#####################
X_220_matrice <- X_220_matrice[seq(from = 1, to = 220),]  #time x profili
storage.mode(X_220_matrice) <- "numeric"
n_obs_220 <- ncol(X_220_matrice)
n_point_220 <- nrow(X_220_matrice)
grid_220 <- seq(from = 0, to = 1, length.out = n_point_220)
data_220 <- list(X = X_220_matrice, grid = grid_220)

df_long_220 <- data_220$X %>%
  as.data.frame %>%
  mutate(grid = data_220$grid) %>%
  pivot_longer(-grid, names_to = "observation")

profili_220<- ggplot(df_long_220) +
  geom_line(mapping = aes(x = grid, y = value, group = observation)) +
  ggtitle("Plot of Group 6 220ms's DRC")
ggexport(profili_220, filename = "ggPlot group 6 220ms's DRC.png")
```

```r
X_300_matrice <- X_300_matrice[seq(from = 1, to = 300),]  #time x profili
storage.mode(X_300_matrice) <- "numeric"
n_obs_300 <- ncol(X_300_matrice)
n_point_300 <- nrow(X_300_matrice)
grid_300 <- seq(from = 0, to = 1, length.out = n_point_300)
data_300 <- list(X = X_300_matrice, grid = grid_300)

df_long_300 <- data_300$X %>%
  as.data.frame %>%
  mutate(grid = data_300$grid) %>%
  pivot_longer(-grid, names_to = "observation")

profili_300 <- ggplot(df_long_300) +
  geom_line(mapping = aes(x = grid, y = value, group = observation)) +
  ggtitle("Plot of Group 6 300ms's DRC")
ggexport(profili_300, filename = "ggPlot group 6 300ms's DRC.png")
######################

num_cluster_seq <- 2:4
ncores <- 1


#ADAPTIVE CLUSTERING

#fclust
######################
#Inizio procedura per gruppo 6 tagliato a 220 ms
X_6_220_adaptive <- X_220_matrice[seq(from = 1, to = 220, by = 10),
                                  seq(from = 1, to = 495, by = 5)]  #time x profili
storage.mode(X_6_220_adaptive) <- "numeric"
n_obs_6_adaptive <- ncol(X_6_220_adaptive)
n_point_6_220_adaptive <- nrow(X_6_220_adaptive)
grid_6_220_adaptive <- seq(from = 0, to = 1, length.out = n_point_6_220_adaptive)
data_6_220_adaptive <- list(X = X_6_220_adaptive, grid = grid_6_220_adaptive)

mod1_220 <- fit_fclust_ms(data = data_6_220_adaptive,
                          num_cluster_seq = num_cluster_seq,
                          dim_seq = c(5, 10),
                          ncores = ncores)
mod1_220$class_opt_BIC
cl_1_220 <- mod1_220$class_opt_BIC

clustered_funs1_220 <- get_clustered_funs_df(cl = cl_1_220,
                                             data = data_6_220_adaptive,
                                             method = "adaptive\nfunHDDC")
centroids1_220 <- get_centroids_df(cl = cl_1_220,
                                   data = data_6_220_adaptive,
                                   method = "adaptive\nfunHDDC")

adapt_fclust_220 <- ggplot(clustered_funs1_220) +
  geom_line(aes(time, Resistance, group = obs, col = cluster))  +
  ggtitle("Adaptive fclust clustering of Group 6_220's DRC")
adapt_fclust_220
adapt_fclust_cen_220 <- ggplot(centroids1_220) +
  geom_line(aes(time, Resistance, col = cluster))  +
  ggtitle("Centroids of Adaptive fclust clustering of Group 6_220's DRC")
adapt_fclust_cen_220

full_adapt_fclust_220 <- ggarrange(adapt_fclust_220,adapt_fclust_cen_220, ncol = 1, nrow = 2)
ggexport(full_adapt_fclust_220, filename = "Adaptive fclust clustering 220ms.png")


#Inizio procedura per gruppo 6 tagliato a 300 ms
X_6_300_adaptive <- X_300_matrice[seq(from = 1, to = 300, by = 10),
                                  seq(from = 1, to = 708, by = 5)]  #time x profili
storage.mode(X_6_300_adaptive) <- "numeric"
n_obs_6_adaptive <- ncol(X_6_300_adaptive)
n_point_6_300_adaptive <- nrow(X_6_300_adaptive)
grid_6_300_adaptive <- seq(from = 0, to = 1, length.out = n_point_6_300_adaptive)
data_6_300_adaptive <- list(X = X_6_300_adaptive, grid = grid_6_300_adaptive)

mod1_300 <- fit_fclust_ms(data = data_6_300_adaptive,
                          num_cluster_seq = num_cluster_seq,
                          dim_seq = c(5, 10),
                          ncores = ncores)
mod1_300$class_opt_BIC
cl_1_300 <- mod1_300$class_opt_BIC

clustered_funs1_300 <- get_clustered_funs_df(cl = cl_1_300,
                                             data = data_6_300_adaptive,
                                             method = "adaptive\nfunHDDC")
centroids1_300 <- get_centroids_df(cl = cl_1_300,
                                   data = data_6_300_adaptive,
                                   method = "adaptive\nfunHDDC")

adapt_fclust_300 <- ggplot(clustered_funs1_300) +
  geom_line(aes(time, Resistance, group = obs, col = cluster))  +
  ggtitle("Adaptive fclust clustering of Group 6_300's DRC")
adapt_fclust_300
adapt_fclust_cen_300 <- ggplot(centroids1_300) +
  geom_line(aes(time, Resistance, col = cluster))  +
  ggtitle("Centroids of Adaptive fclust clustering of Group 6_300's DRC")
adapt_fclust_cen_300
```

```
full_adapt_fclust_300 <- ggarrange(adapt_fclust_300,adapt_fclust_cen_300,
                                   ncol = 1, nrow = 2)
ggexport(full_adapt_fclust_300, filename = "Adaptive fclust clustering 300ms.png")
#####################


#curvclust
####################
parameters <- expand.grid(
  structures = c(
    "constant",
    "group",
    "scale.location",
    "group.scale.location",
    "none"
  ),
  mixed = c(
    TRUE,
    FALSE
  ),
  reduction = c(
    TRUE,
    FALSE
  )
) %>%
  filter(!(structures == "none" & mixed),
         !(structures != "none" & !mixed)) %>%
  mutate(structures = as.character(structures))

#Inizio procedura per gruppo 6 tagliato a 220 ms
mod2_220 <- vector(mode = "list", length = nrow(parameters))
names(mod2_220) <- sapply(1:nrow(parameters), function(ii) {
  parameters[ii,] %>%
    mutate(structures = paste0("structure_", structures),
           mixed = paste0("mixed_", mixed),
           reduction = paste0("reduction_", reduction)) %>%
    paste0(collapse = " ")
})
mod2_220

for (ii in 1:nrow(parameters)) {
  mod2_220[[ii]] <- curvclust_ms(data=data_6_220_adaptive,
                                 num_cluster_seq = c(1, num_cluster_seq),
                                 structure = parameters$structures[ii],
                                 mixed = parameters$mixed[ii],
                                 reduction = parameters$reduction[ii])
}

bic_220 <- sapply(mod2_220, function(x) x$BIC)
bic_220

bic_df_long_220 <- bic_220 %>%
  as.data.frame %>%
  mutate(K = c(1, num_cluster_seq)) %>%
  pivot_longer(- K, values_to = "BIC", names_to = "model")
bic_df_long_220

ggplot(bic_df_long_220) +
  geom_line(aes(K, BIC, col = model))

bic_opt_220 <- filter(bic_df_long_220, BIC == max(BIC))
Kopt_220 <- bic_opt_220$K
Kopt_220

mod2_opt_220 <- mod2_220[[bic_opt_220$model]]
cl_2_220 <- mod2_opt_220$class[[Kopt_220]]


clustered_funs2_220 <- get_clustered_funs_df(cl = cl_2_220,
                                             data = data_6_220_adaptive,
                                             method = "adaptive\ncurvclust")
centroids2_220 <- get_centroids_df(cl = cl_2_220,
                                   data = data_6_220_adaptive,
                                   method = "adaptive\ncurvclust")

adapt_curvclust_220 <- ggplot(clustered_funs2_220) +
  geom_line(aes(time, Resistance, group = obs, col = cluster))  +
  ggtitle("Adaptive curvclust clustering of Group 6_220's DRC")
adapt_curvclust_220
adapt_curvclust_cen_220 <- ggplot(centroids2_220) +
  geom_line(aes(time, Resistance, col = cluster))  +
  ggtitle("Centroids of Adaptive curvclust clustering of Group 6_220's DRC")
adapt_curvclust_cen_220

full_adapt_curvclust_220 <- ggarrange(adapt_curvclust_220,adapt_curvclust_cen_220,
                                      ncol = 1, nrow = 2)
ggexport(full_adapt_curvclust_220,
         filename = "Adaptive curvclust clustering 220ms.png")


# Inizio procedura per gruppo 6 tagliato a 300 ms
mod2_300 <- vector(mode = "list", length = nrow(parameters))
names(mod2_300) <- sapply(1:nrow(parameters), function(ii) {
  parameters[ii,] %>%
    mutate(structures = paste0("structure_", structures),
```

```
          mixed = paste0("mixed_", mixed),
          reduction = paste0("reduction_", reduction)) %>%
    paste0(collapse = " ")
})
mod2_300

for (ii in 1:nrow(parameters)) {
  mod2_300[[ii]] <- curvclust_ms(data=data_6_300_adaptive,
                                 num_cluster_seq = c(1, num_cluster_seq),
                                 structure = parameters$structures[ii],
                                 mixed = parameters$mixed[ii],
                                 reduction = parameters$reduction[ii])
}

bic_300 <- sapply(mod2_300, function(x) x$BIC)
bic_300

bic_df_long_300 <- bic_300 %>%
  as.data.frame %>%
  mutate(K = c(1, num_cluster_seq)) %>%
  pivot_longer(- K, values_to = "BIC", names_to = "model")
bic_df_long_300

ggplot(bic_df_long_300) +
  geom_line(aes(K, BIC, col = model))

bic_opt_300 <- filter(bic_df_long_300, BIC == max(BIC))
Kopt_300 <- bic_opt_300$K
Kopt_300

## mod2 curvclust
mod2_opt_300 <- mod2_300[[bic_opt_300$model]]
cl_2_300 <- mod2_opt_300$class[[Kopt_300]]


clustered_funs2_300 <- get_clustered_funs_df(cl = cl_2_300,
                                             data = data_6_300_adaptive,
                                             method = "adaptive\ncurvclust")
centroids2_300 <- get_centroids_df(cl = cl_2_300,
                                   data = data_6_300_adaptive,
                                   method = "adaptive\ncurvclust")


adapt_curvclust_300 <- ggplot(clustered_funs2_300) +
                       geom_line(aes(time, Resistance, group = obs, col = cluster))  +
                       ggtitle("Adaptive curvclust clustering of Group 6_300's DRC")
adapt_curvclust_300
adapt_curvclust_cen_300 <- ggplot(centroids2_300) +
                       geom_line(aes(time, Resistance, col = cluster))  +
                       ggtitle("Centroids of Adaptive curvclust clustering of Group 6_300's DRC")
adapt_curvclust_cen_300


full_adapt_curvclust_300 <- ggarrange(adapt_curvclust_300,adapt_curvclust_cen_300,
                                      ncol = 1, nrow = 2)
ggexport(full_adapt_fclust_300, filename = "Adaptive curvclust clustering 300ms.png")
######################


#funHDDC
######################
#Inizio procedura per gruppo 6 tagliato a 220 ms
mod3_220 <- fit_funHDDC_ms(data = data_6_220_adaptive,
                           num_cluster_seq = num_cluster_seq,
                           model = c('AkjBkQkDk',
                                     'AkjBQkDk',
                                     'AkBkQkDk',
                                     'ABkQkDk',
                                     'AkBQkDk',
                                     'ABQkDk'),
                           threshold_seq = c(.5, .9),
                           nb.rep = 20)
mod3_220$mod_opt$K
cl_3_220 <- mod3_220$mod$class

clustered_funs3_220 <- get_clustered_funs_df(cl = cl_3_220,
                                             data = data_6_220_adaptive,
                                             method = "adaptive\nfunHDDC")
centroids3_220 <- get_centroids_df(cl = cl_3_220,
                                   data = data_6_220_adaptive,
                                   method = "adaptive\nfunHDDC")

adapt_funHDDC_220 <- ggplot(clustered_funs3_220) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Adaptive funHDDC clustering of Group 6_220's DRC")
adapt_funHDDC_220
adapt_funHDDC_cen_220 <- ggplot(centroids3_220) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle("Centroids of Adaptive funHDDC clustering of Group 6_220's DRC")
adapt_funHDDC_cen_220

full_adapt_funHDDC_220 <- ggarrange(adapt_funHDDC_220,adapt_funHDDC_cen_220,
                                    ncol = 1, nrow = 2)
ggexport(full_adapt_funHDDC_220, filename = "Adaptive funHDDC clustering 220ms.png")
```

```
#Inizio procedura per gruppo 6 tagliato a 300 ms
mod3_300 <- fit_funHDDC_ms(data = data_6_300_adaptive,
                           num_cluster_seq = num_cluster_seq,
                           model = c('AkjBkQkDk',
                                     'AkjBQkDk',
                                     'AkBkQkDk',
                                     'ABkQkDk',
                                     'AkBQkDk',
                                     'ABQkDk'),
                           threshold_seq = c(.5, .9),
                           nb.rep = 20)
mod3_300$mod_opt$K
cl_3_300 <- mod3_300$mod$class

clustered_funs3_300 <- get_clustered_funs_df(cl = cl_3_300,
                                             data = data_6_300_adaptive,
                                             method = "adaptive\nfunHDDC")
centroids3_300 <- get_centroids_df(cl = cl_3_300,
                                   data = data_6_300_adaptive,
                                   method = "adaptive\nfunHDDC")

adapt_funHDDC_300 <- ggplot(clustered_funs3_300) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Adaptive funHDDC clustering of Group 6_300's DRC")
adapt_funHDDC_300
adapt_funHDDC_cen_300 <- ggplot(centroids3_300) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle("Centroids of Adaptive funHDDC clustering of Group 6_300's DRC")
adapt_funHDDC_cen_300

full_adapt_funHDDC_300 <- ggarrange(adapt_funHDDC_300,adapt_funHDDC_cen_300,
                                    ncol = 1, nrow = 2)
ggexport(full_adapt_funHDDC_300, filename = "Adaptive funHDDC clustering 300ms.png")
######################

#confronto tecniche adaptive
######################
adatptive_clustering_220 <- ggarrange(adapt_fclust_220,adapt_curvclust_220,adapt_funHDDC_220,
                                      adapt_fclust_cen_220,adapt_curvclust_cen_220,
                                      adapt_funHDDC_cen_220, ncol = 1, nrow = 3)
ggexport(adatptive_clustering_220, filename = "Adaptive clustering technique's comparison 220ms.png")

adatptive_clustering_300 <- ggarrange(adapt_fclust_300,adapt_curvclust_300,adapt_funHDDC_300,
                                      adapt_fclust_cen_300,adapt_curvclust_cen_300,
                                      adapt_funHDDC_cen_300, ncol = 1, nrow = 3)
ggexport(adatptive_clustering_300, filename = "Adaptive clustering technique's comparison 300ms.png")
######################



#DISTANCE-BASED CLUSTERING
######################
#Inizio procedura per gruppo 6 tagliato a 220 ms
mod4_220 <- distance_ms(data = data_220, num_cluster_seq = num_cluster_seq, met = "other")
mod4_220$sil_opt
cl_4_220 <- mod4_220$mod_opt_sil$clus

#clusterizzo i profili
clustered_funs4_220 <- get_clustered_funs_df(cl = cl_4_220,
                                             data = data_220,
                                             method = "distance-based")

#ottengo i centroidi dei cluster
centroids4_220 <- get_centroids_df(cl = cl_4_220,
                                   data = data_220,
                                   method = "distance-based")

#plotto i profili clusterizzati
dis_bas_220 <- ggplot(clustered_funs4_220) +
  geom_line(aes(time, Resistance, group = obs, col = cluster))  +
  ggtitle("Distance-based clustering of Group 6_220ms's DRC")
dis_bas_220
#plotto i centroidi dei cluster
dis_bas_cen_220 <- ggplot(centroids4_220) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle("Centroids of distance-based clustering", "\n", "of Group 6_220ms's DRC")
dis_bas_cen_220

#salvo png profili clusterizzati e centroidi cluster
full_distance_based_220 <- ggarrange(dis_bas_220,dis_bas_cen_220, ncol = 1, nrow = 2)
ggexport(full_distance_based_220, filename = "Distance based clustering 220ms.png")


#Inizio procedura per gruppo 6 tagliato a 300ms
mod4_300 <- distance_ms(data = data_300, num_cluster_seq = num_cluster_seq, met = "other")
mod4_300$sil_opt
cl_4_300 <- mod4_300$mod_opt_sil$clus

clustered_funs4_300 <- get_clustered_funs_df(cl = cl_4_300,
                                             data = data_300,
                                             method = "distance-based")
centroids4_300 <- get_centroids_df(cl = cl_4_300,
                                   data = data_300,
                                   method = "distance-based")
```

```r
dis_bas_300 <- ggplot(clustered_funs4_300) +
  geom_line(aes(time, Resistance, group = obs, col = cluster))  +
  ggtitle("Distance-based clustering of Group 6_300ms's DRC")
dis_bas_300
dis_bas_cen_300 <- ggplot(centroids4_300) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of distance-based clustering", "\n", "of Group 6_300ms's DRC"))
dis_bas_cen_300

full_distance_based_300 <- ggarrange(dis_bas_300,dis_bas_cen_300, ncol = 1, nrow = 2)
ggexport(full_distance_based_300, filename = "Distance based clustering 300ms.png")
#####################



#FILTERING METHODS

#B-spline basis
####################
#Inizio procedura per gruppo 6 tagliato a 220 ms
mod5_220 <- fil_bspline_ms_nclust(data = data_220, num_cluster_seq = num_cluster_seq, nbasis = 28)
mod5_220$mod_opt$ind_hc$nbclust ## hierarchical
max(mod5_220$mod_opt$ind_km$cluster) ## k-means
mod5_220$mod_opt$mod_opt$G ## model-based

cl_5_hc_220 <- mod5_220$mod_opt$ind_hc$cluster
cl_5_km_220 <- mod5_220$mod_opt$ind_km$cluster
cl_5_mb_220 <- mod5_220$mod_opt$mod_opt$classification

clustered_funs5_hc_220 <- get_clustered_funs_df(cl = cl_5_hc_220,
                                                data = data_220,
                                                method = "filtering B-spline\nhierarchical")
clustered_funs5_km_220 <- get_clustered_funs_df(cl = cl_5_km_220,
                                                data = data_220,
                                                method = "filtering B-spline\nk-means")
clustered_funs5_mb_220 <- get_clustered_funs_df(cl = cl_5_mb_220,
                                                data = data_220,
                                                method = "filtering B-spline\nmodel-based")
centroids5_hc_220 <- get_centroids_df(cl_5_hc_220, data_220, "filtering B-spline\nhierarchical")
centroids5_km_220 <- get_centroids_df(cl_5_km_220, data_220, "filtering B-spline\nk-means")
centroids5_mb_220 <- get_centroids_df(cl_5_mb_220, data_220, "filtering B-spline\nmodel-based")

filt_bspl_hc_220 <- ggplot(clustered_funs5_hc_220) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering B-spline hierarchical clustering of Group 6 220ms's DRC")
filt_bspl_hc_220
filt_bspl_km_220 <- ggplot(clustered_funs5_km_220) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering B-spline k-means clustering of Group 6 220ms's DRC")
filt_bspl_km_220
filt_bspl_mb_220 <- ggplot(clustered_funs5_mb_220) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering B-spline model-based clustering of Group 6 220ms's DRC")
filt_bspl_mb_220
filt_bspl_hc_cen_220 <- ggplot(centroids5_hc_220) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering B-spline hierarchical clustering", "\n", "of Group 6 220ms's DRC"))
filt_bspl_hc_cen_220
filt_bspl_km_cen_220 <- ggplot(centroids5_km_220) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering B-spline k-means clustering", "\n", "of Group 6 220ms's DRC"))
filt_bspl_km_cen_220
filt_bspl_mb_cen_220 <- ggplot(centroids5_mb_220) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering B-spline modelbased clustering", "\n", "of Group 6 220ms's DRC"))
filt_bspl_mb_cen_220

full_filt_bspl_220 <- ggarrange(filt_bspl_hc_220,filt_bspl_hc_cen_220,filt_bspl_km_220,filt_bspl_km_cen_220,
                                filt_bspl_mb_220,filt_bspl_mb_cen_220,ncol=1,nrow=2)
ggexport(full_filt_bspl_220, filename = "Filtering B-spline profili-centroidi 220ms.png")

confr_filt_bspl_220 <- ggarrange(filt_bspl_hc_220,filt_bspl_km_220,filt_bspl_mb_220,filt_bspl_hc_cen_220,
                                 filt_bspl_km_cen_220,filt_bspl_mb_cen_220,ncol=1,nrow=3)
ggexport(confr_filt_bspl_220, filename = "Filtering B-spline technique's comparison 220ms.png")


#Inizio procedura per gruppo 6 tagliato a 300 ms
mod5_300 <- fil_bspline_ms_nclust(data = data_300, num_cluster_seq = num_cluster_seq, nbasis = 28)
mod5_300$mod_opt$ind_hc$nbclust ## hierarchical
max(mod5_300$mod_opt$ind_km$cluster) ## k-means
mod5_300$mod_opt$mod_opt$G ## model-based

cl_5_hc_300 <- mod5_300$mod_opt$ind_hc$cluster
cl_5_km_300 <- mod5_300$mod_opt$ind_km$cluster
cl_5_mb_300 <- mod5_300$mod_opt$mod_opt$classification

clustered_funs5_hc_300 <- get_clustered_funs_df(cl = cl_5_hc_300,
                                                data = data_300,
                                                method = "filtering B-spline\nhierarchical")
clustered_funs5_km_300 <- get_clustered_funs_df(cl = cl_5_km_300,
                                                data = data_300,
                                                method = "filtering B-spline\nk-means")
clustered_funs5_mb_300 <- get_clustered_funs_df(cl = cl_5_mb_300,
                                                data = data_300,
```

```
                                          method = "filtering B-spline\nmodel-based")
centroids5_hc_300 <- get_centroids_df(cl_5_hc_300, data_300, "filtering B-spline\nhierarchical")
centroids5_km_300 <- get_centroids_df(cl_5_km_300, data_300, "filtering B-spline\nk-means")
centroids5_mb_300 <- get_centroids_df(cl_5_mb_300, data_300, "filtering B-spline\nmodel-based")

filt_bspl_hc_300 <- ggplot(clustered_funs5_hc_300) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering B-spline hierarchical clustering of Group 6 300ms's DRC")
filt_bspl_hc_300
filt_bspl_km_300 <- ggplot(clustered_funs5_km_300) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering B-spline k-means clustering of Group 6 300ms's DRC")
filt_bspl_km_300
filt_bspl_mb_300 <- ggplot(clustered_funs5_mb_300) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering B-spline model-based clustering of Group 6 300ms's DRC")
filt_bspl_mb_300
filt_bspl_hc_cen_300 <- ggplot(centroids5_hc_300) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering B-spline hierarchical clustering", "\n", "of Group 6 300ms's DRC"))
filt_bspl_hc_cen_300
filt_bspl_km_cen_300 <- ggplot(centroids5_km_300) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering B-spline k-means clustering", "\n", "of Group 6 300ms's DRC"))
filt_bspl_km_cen_300
filt_bspl_mb_cen_300 <- ggplot(centroids5_mb_300) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering B-spline model-based clustering", "\n", "of Group 6 300ms's DRC"))
filt_bspl_mb_cen_300

full_filt_bspl_300 <- ggarrange(filt_bspl_hc_300,filt_bspl_hc_cen_300,filt_bspl_km_300,filt_bspl_km_cen_300,
                                filt_bspl_mb_300,filt_bspl_mb_cen_300,ncol=1,nrow=2)
ggexport(full_filt_bspl_300, filename = "Filtering B-spline profili-centroidi 300ms.png")

confr_filt_bspl_300 <- ggarrange(filt_bspl_hc_300,filt_bspl_km_300,filt_bspl_mb_300,filt_bspl_hc_cen_300,
                                 filt_bspl_km_cen_300,filt_bspl_mb_cen_300,ncol=1,nrow=3)
ggexport(confr_filt_bspl_300, filename = "Filtering B-spline technique's comparison 300ms.png")
#####################


#FPCA basis
####################
#Inizio procedura per gruppo 6 tagliato a 220 ms
mod6_220 <- fil_fpca_ss_nbclust(data = data_220, num_cluster_seq = num_cluster_seq, per_comp = 0.8)
mod6_220$mod_opt$ind_hc$nbclust ## hierarchical
max(mod6_220$mod_opt$ind_km$cluster) ## k-means
mod6_220$mod_opt$mod_opt$G ## model-based

cl_6_km_220 <- mod6_220$mod_opt$ind_km$cluster
cl_6_hc_220 <- mod6_220$mod_opt$ind_hc$cluster
cl_6_mb_220 <- mod6_220$mod_opt$mod_opt$classification

clustered_funs6_hc_220 <- get_clustered_funs_df(cl = cl_6_hc_220,
                                          data = data_220,
                                          method = "filtering FPCA\nhierarchical")
clustered_funs6_km_220 <- get_clustered_funs_df(cl = cl_6_km_220,
                                          data = data_220,
                                          method = "filtering FPCA\nk-means")
clustered_funs6_mb_220 <- get_clustered_funs_df(cl = cl_6_mb_220,
                                          data = data_220,
                                          method = "filtering FPCA\nmodel-based")
centroids6_hc_220 <- get_centroids_df(cl_6_hc_220, data_220, "filtering FPCA\nhierarchical")
centroids6_km_220 <- get_centroids_df(cl_6_km_220, data_220, "filtering FPCA\nk-means")
centroids6_mb_220 <- get_centroids_df(cl_6_mb_220, data_220, "filtering FPCA\nmodel-based")

filt_fpca_hc_220 <- ggplot(clustered_funs6_hc_220) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering FPCA hierarchical clustering of Group 6 220ms's DRC")
filt_fpca_hc_220
filt_fpca_km_220 <- ggplot(clustered_funs6_km_220) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering FPCA k-means clustering of Group 6 220ms's DRC")
filt_fpca_km_220
filt_fpca_mb_220 <- ggplot(clustered_funs6_mb_220) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering FPCA model-based clustering of Group 6 220ms's DRC")
filt_fpca_mb_220
filt_fpca_hc_cen_220 <- ggplot(centroids6_hc_220) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering FPCA hierarchical clustering", "\n", "of Group 6 220ms's DRC"))
filt_fpca_hc_cen_220
filt_fpca_km_cen_220 <- ggplot(centroids6_km_220) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering FPCA k-means clustering", "\n", "of Group 6 220ms's DRC"))
filt_fpca_km_cen_220
filt_fpca_mb_cen_220 <- ggplot(centroids6_mb_220) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering FPCA model-based clustering", "\n", "of Group 6 220ms's DRC"))
filt_fpca_mb_cen_220

full_filt_fpca_220 <- ggarrange(filt_fpca_hc_220,filt_fpca_hc_cen_220,filt_fpca_km_220,filt_fpca_km_cen_220,
                                filt_fpca_mb_220,filt_fpca_mb_cen_220,ncol=1,nrow=2)
ggexport(full_filt_fpca_220, filename = "Filtering fPCA profili-centroidi 220ms.png")

confr_filt_fpca_220 <- ggarrange(filt_fpca_hc_220,filt_fpca_km_220,filt_fpca_mb_220,filt_fpca_hc_cen_220,
```

```
                                filt_fpca_km_cen_220,filt_fpca_mb_cen_220,ncol=1,nrow=3)
ggexport(confr_filt_fpca_220, filename = "Filtering fPCA technique's comparison 220ms.png")


#Inizio procedura per gruppo 6 tagliato a 300 ms
mod6_300 <- fil_fpca_ss_nbclust(data = data_300, num_cluster_seq = num_cluster_seq, per_comp = 0.8)
mod6_300$mod_opt$ind_hc$nbclust ## hierarchical
max(mod6_300$mod_opt$ind_km$cluster) ## k-means
mod6_300$mod_opt$mod_opt$G ## model-based

cl_6_km_300 <- mod6_300$mod_opt$ind_km$cluster
cl_6_hc_300 <- mod6_300$mod_opt$ind_hc$cluster
cl_6_mb_300 <- mod6_300$mod_opt$mod_opt$classification

clustered_funs6_hc_300 <- get_clustered_funs_df(cl = cl_6_hc_300,
                                                data = data_300,
                                                method = "filtering FPCA\nhierarchical")
clustered_funs6_km_300 <- get_clustered_funs_df(cl = cl_6_km_300,
                                                data = data_300,
                                                method = "filtering FPCA\nk-means")
clustered_funs6_mb_300 <- get_clustered_funs_df(cl = cl_6_mb_300,
                                                data = data_300,
                                                method = "filtering FPCA\nmodel-based")
centroids6_hc_300 <- get_centroids_df(cl_6_hc_300, data_300, "filtering FPCA\nhierarchical")
centroids6_km_300 <- get_centroids_df(cl_6_km_300, data_300, "filtering FPCA\nk-means")
centroids6_mb_300 <- get_centroids_df(cl_6_mb_300, data_300, "filtering FPCA\nmodel-based")

filt_fpca_hc_300 <- ggplot(clustered_funs6_hc_300) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering FPCA hierarchical clustering of Group 6 300ms's DRC")
filt_fpca_hc_300
filt_fpca_km_300 <- ggplot(clustered_funs6_km_300) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering FPCA k-means clustering of Group 6 300ms's DRC")
filt_fpca_km_300
filt_fpca_mb_300 <- ggplot(clustered_funs6_mb_300) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering FPCA model-based clustering of Group 6 300ms's DRC")
filt_fpca_mb_300
filt_fpca_hc_cen_300 <- ggplot(centroids6_hc_300) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering FPCA hierarchical clustering", "\n", "of Group 6 300ms's DRC"))
filt_fpca_hc_cen_300
filt_fpca_km_cen_300 <- ggplot(centroids6_km_300) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering FPCA k-means clustering", "\n", "of Group 6 300ms's DRC"))
filt_fpca_km_cen_300
filt_fpca_mb_cen_300 <- ggplot(centroids6_mb_300) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering FPCA model-based clustering", "\n", "of Group 6 300ms's DRC"))
filt_fpca_mb_cen_300

full_filt_fpca_300 <- ggarrange(filt_fpca_hc_300,filt_fpca_hc_cen_300,filt_fpca_km_300,filt_fpca_km_cen_300,
                                filt_fpca_mb_300,filt_fpca_mb_cen_300,ncol=1,nrow=2)
ggexport(full_filt_fpca_300, filename = "Filtering fPCA profili-centroidi 300ms.png")

confr_filt_fpca_300 <- ggarrange(filt_fpca_hc_300,filt_fpca_km_300,filt_fpca_mb_300,filt_fpca_hc_cen_300,
                                 filt_fpca_km_cen_300,filt_fpca_mb_cen_300,ncol=1,nrow=3)
ggexport(confr_filt_fpca_300, filename = "Filtering fPCA technique's comparison 300ms.png")
#####################



#RAW-DATA CLUSTERING
#data preparation
#####################
#Per 220ms
X_220_raw <- X_220_matrice[seq(from = 1, to = 220, by = 5),seq(from = 1, to = 495)]  #time x profili
storage.mode(X_220_raw) <- "numeric"
n_obs_220_raw <- ncol(X_220_raw)
n_point_220_raw <- nrow(X_220_raw)
grid_220_raw <- seq(from = 0, to = 1, length.out = n_point_220_raw)
data_220_raw <- list(X = X_220_raw, grid = grid_220_raw)

df_long_220_raw <- data_220_raw$X %>%
  as.data.frame %>%
  mutate(grid = data_220_raw$grid) %>%
  pivot_longer(-grid, names_to = "observation")

print(ggplot(df_long_220_raw) +
        geom_line(mapping = aes(x = grid, y = value, group = observation)) +
        ggtitle("Plot of Group 6 220's DRC for Raw-data clustering")
)


#Per 300ms
X_300_raw <- X_300_matrice[seq(from = 1, to = 300, by = 5),seq(from = 1, to = 708)]  #time x profili
storage.mode(X_300_raw) <- "numeric"
n_obs_300_raw <- ncol(X_300_raw)
n_point_300_raw <- nrow(X_300_raw)
grid_300_raw <- seq(from = 0, to = 1, length.out = n_point_300_raw)
data_300_raw <- list(X = X_300_raw, grid = grid_300_raw)

df_long_300_raw <- data_300_raw$X %>%
  as.data.frame %>%
```

```
  mutate(grid = data_300_raw$grid) %>%
  pivot_longer(-grid, names_to = "observation")

print(ggplot(df_long_300_raw) +
        geom_line(mapping = aes(x = grid, y = value, group = observation)) +
        ggtitle("Plot of Group 6's DRC for Raw-data clustering")
)
#####################

#Procedura
#####################
#Inizio procedura per gruppo 6 tagliato a 220 ms
mod7_220 <- raw_ms_nbclust(data = data_220_raw, num_cluster_seq = num_cluster_seq)
mod7_220$mod_opt$ind_hc$nbclust ## hierarchical
max(mod7_220$mod_opt$ind_km$cluster) ## k-means
mod7_220$mod_opt$mod_opt$G ## model-based

cl_7_km_220 <- mod7_220$mod_opt$ind_km$cluster
cl_7_hc_220 <- mod7_220$mod_opt$ind_hc$cluster
cl_7_mb_220 <- mod7_220$mod_opt$mod_opt$classification

clustered_funs7_hc_220 <- get_clustered_funs_df(cl = cl_7_hc_220,
                                                data = data_220_raw,
                                                method = "raw-data\nhierarchical")
clustered_funs7_km_220 <- get_clustered_funs_df(cl = cl_7_km_220,
                                                data = data_220_raw,
                                                method = "raw-data\nk-means")

clustered_funs7_mb_220 <- get_clustered_funs_df(cl = cl_7_mb_220,
                                                data = data_220_raw,
                                                method = "raw-data\nmodel-based")
centroids7_hc_220 <- get_centroids_df(cl_7_hc_220, data_220_raw, "filtering raw-data\nhierarchical")
centroids7_km_220 <- get_centroids_df(cl_7_km_220, data_220_raw, "filtering raw-data\nk-means")
centroids7_mb_220 <- get_centroids_df(cl_7_mb_220, data_220_raw, "filtering raw-data\nmodel-based")

rawdata_hc_220 <- ggplot(clustered_funs7_hc_220) +
  geom_line(aes(time, Resistance, group = obs, col = cluster))  +
  ggtitle("Raw-data hierarchical clustering of Group 6 220's DRC")
rawdata_hc_220
rawdata_km_220 <- ggplot(clustered_funs7_km_220) +
  geom_line(aes(time, Resistance, group = obs, col = cluster))  +
  ggtitle("Raw-data k-means clustering of Group 6 220's DRC")
rawdata_km_220
rawdata_mb_220 <- ggplot(clustered_funs7_mb_220) +
  geom_line(aes(time, Resistance, group = obs, col = cluster))  +
  ggtitle("Raw-data model-based clustering of Group 6 220's DRC")
rawdata_mb_220
rawdata_hc_cen_220 <- ggplot(centroids7_hc_220) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of Filtering Raw-data hierarchical clustering", "\n","of Group 6 220's DRC"))
rawdata_hc_cen_220
rawdata_km_cen_220 <- ggplot(centroids7_km_220) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of Filtering Raw-data k-means clustering", "\n","of Group 6 220's DRC"))
rawdata_km_cen_220
rawdata_mb_cen_220 <- ggplot(centroids7_mb_220) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of Filtering Raw-data model-based clustering", "\n","of Group 6 220's DRC"))
rawdata_mb_cen_220

full_rawdata_220 <- ggarrange(rawdata_hc_220,rawdata_hc_cen_220,rawdata_km_220,rawdata_km_cen_220,
                              rawdata_mb_220,rawdata_mb_cen_220,ncol=1,nrow=2)
ggexport(full_rawdata_220, filename = "Raw-data profili-centroidi 220ms.png")

confr_rawdata_220 <- ggarrange(rawdata_hc_220,rawdata_km_220,rawdata_mb_220,rawdata_hc_cen_220,
                               rawdata_km_cen_220,rawdata_mb_cen_220,ncol=1,nrow=3)
ggexport(confr_filt_fpca_300, filename = "Raw-data technique's comparison 220ms.png")


#Inizio procedura per gruppo 6 tagliato a 300 ms
mod7_300 <- raw_ms_nbclust(data = data_300_raw, num_cluster_seq = num_cluster_seq)
mod7_300$mod_opt$ind_hc$nbclust ## hierarchical
max(mod7_300$mod_opt$ind_km$cluster) ## k-means
mod7_300$mod_opt$mod_opt$G ## model-based

cl_7_km_300 <- mod7_300$mod_opt$ind_km$cluster
cl_7_hc_300 <- mod7_300$mod_opt$ind_hc$cluster
cl_7_mb_300 <- mod7_300$mod_opt$mod_opt$classification

clustered_funs7_hc_300 <- get_clustered_funs_df(cl = cl_7_hc_300,
                                                data = data_300_raw,
                                                method = "raw-data\nhierarchical")
clustered_funs7_km_300 <- get_clustered_funs_df(cl = cl_7_km_300,
                                                data = data_300_raw,
                                                method = "raw-data\nk-means")

clustered_funs7_mb_300 <- get_clustered_funs_df(cl = cl_7_mb_300,
                                                data = data_300_raw,
                                                method = "raw-data\nmodel-based")
centroids7_hc_300 <- get_centroids_df(cl_7_hc_300, data_300_raw, "filtering raw-data\nhierarchical")
centroids7_km_300 <- get_centroids_df(cl_7_km_300, data_300_raw, "filtering raw-data\nk-means")
centroids7_mb_300 <- get_centroids_df(cl_7_mb_300, data_300_raw, "filtering raw-data\nmodel-based")

rawdata_hc_300 <- ggplot(clustered_funs7_hc_300) +
  geom_line(aes(time, Resistance, group = obs, col = cluster))  +
```

```
  ggtitle("Raw-data hierarchical clustering of Group 6 300's DRC")
rawdata_hc_300
rawdata_km_300 <- ggplot(clustered_funs7_km_300) +
  geom_line(aes(time, Resistance, group = obs, col = cluster))  +
  ggtitle("Raw-data k-means clustering of Group 6 300's DRC")
rawdata_km_300
rawdata_mb_300 <- ggplot(clustered_funs7_mb_300) +
  geom_line(aes(time, Resistance, group = obs, col = cluster))  +
  ggtitle("Raw-data model-based clustering of Group 6 300's DRC")
rawdata_mb_300
rawdata_hc_cen_300 <- ggplot(centroids7_hc_300) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of Filtering Raw-data hierarchical clustering", "\n","of Group 6 300's DRC"))
rawdata_hc_cen_300
rawdata_km_cen_300 <- ggplot(centroids7_km_300) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of Filtering Raw-data k-means clustering", "\n","of Group 6 300's DRC"))
rawdata_km_cen_300
rawdata_mb_cen_300 <- ggplot(centroids7_mb_300) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of Filtering Raw-data model-based clustering", "\n","of Group 6 300's DRC"))
rawdata_mb_cen_300

full_rawdata_300 <- ggarrange(rawdata_hc_300,rawdata_hc_cen_300,rawdata_km_300,rawdata_km_cen_300,
                              rawdata_mb_300,rawdata_mb_cen_300,ncol=1,nrow=2)
ggexport(full_rawdata_300, filename = "Raw-data profili-centroidi 300ms.png")

confr_rawdata_300 <- ggarrange(rawdata_hc_300,rawdata_km_300,rawdata_mb_300,rawdata_hc_cen_300,
                               rawdata_km_cen_300,rawdata_mb_cen_300,ncol=1,nrow=3)
ggexport(confr_filt_fpca_300, filename = "Raw-data technique's comparison 300ms.png")
######################



#Creazione tabelle di contingenza: per ogni spotname si avrà la ripartizione nei vari cluster
#Solo con distance based, ha mostrato i risultati migliori

#DISTANCE BASED 220
profili_cluster_dis_bas_220 <- clustered_funs4_220[order(clustered_funs4_220$obs),]
profili_cluster_dis_bas_220 <- profili_cluster_dis_bas_220[seq(
  from = 1, to = nrow(profili_cluster_dis_bas_220), by = 220),]

#cambio nome a colonna delle obs e di cluster
colnames(profili_cluster_dis_bas_220)[2] <- "dateTime"
colnames(profili_cluster_dis_bas_220)[4] <- "cluster dis-bas_220"

#creo copia scalari associazione 6
scalari_cluster_dis_bas_220 <- scalari_associazione
#unisco colonna relativa al cluster di appartenenza
scalari_cluster_dis_bas_220 <- merge(scalari_cluster_dis_bas_220,profili_cluster_dis_bas_220,
                                     by = ("dateTime"))


#DISTANCE BASED 300
profili_cluster_dis_bas_300 <- clustered_funs4_300[order(clustered_funs4_300$obs),]
profili_cluster_dis_bas_300 <- profili_cluster_dis_bas_300[seq(
  from = 1, to = nrow(profili_cluster_dis_bas_300), by = 300),]

colnames(profili_cluster_dis_bas_300)[2] <- "dateTime"
colnames(profili_cluster_dis_bas_300)[4] <- "cluster dis-bas_300"

scalari_cluster_dis_bas_300 <- scalari_associazione
scalari_cluster_dis_bas_300 <- merge(scalari_cluster_dis_bas_300,profili_cluster_dis_bas_300,
                                     by = ("dateTime"))

#seleziono solo le colonne di interesse (dateTime, spotName e la colonna dei cluster)
scalari_cluster_dis_bas_220_ridotta <- scalari_cluster_dis_bas_220[, c(1,2,130)]

scalari_cluster_dis_bas_300_ridotta <- scalari_cluster_dis_bas_300[, c(1,2,130)]

#Isolo i vari spotname per vedere in quali cluster ricadono
spotname_220 <- factor(scalari_cluster_dis_bas_220$spotName)
spotname_220 <- levels(spotname_220)
spotname_300 <- factor(scalari_cluster_dis_bas_300$spotName)
spotname_300 <- levels(spotname_300)


cluster_spotname_220 <- list()
cluster_spotname_220_tot <- list()

for (i in 1:length(spotname_220)) {

  cluster_spotname_220 <- scalari_cluster_dis_bas_220_ridotta[
    scalari_cluster_dis_bas_220_ridotta[,"spotName"] == spotname_220[i],]
  cluster_spotname_220_tot[[i]] <- cluster_spotname_220
  names(cluster_spotname_220_tot)[[i]] <- spotname_220[i]

}

cluster_spotname_300 <- list()
cluster_spotname_300_tot <- list()

for (i in 1:length(spotname_300)) {
```

```
  cluster_spotname_300 <- scalari_cluster_dis_bas300_ridotta[
    scalari_cluster_dis_bas300_ridotta[,"spotName"] == spotname_300[i],]
  cluster_spotname_300_tot[[i]] <- cluster_spotname_300
  names(cluster_spotname_300_tot)[[i]] <- spotname_300[i]

}
```

```
#Creazione tabella di contingenza clustering distance based-spotname 220
cluster1_dis_bas_220 <- vector()
cluster2_dis_bas_220 <- vector()
cluster3_dis_bas_220 <- vector()

for (i in 1:length(spotname_220)) {

  cluster1_dis_bas_220 [i] <- length(which(cluster_spotname_220_tot[[i]]["cluster dis-bas_220"] == 1))
  names(cluster1_dis_bas_220)[i] <- spotname_220[i]

  cluster2_dis_bas_220 [i] <- length(which(cluster_spotname_220_tot[[i]]["cluster dis-bas_220"] == 2))
  names(cluster2_dis_bas_220)[i] <- spotname_220[i]

  cluster3_dis_bas_220 [i] <- length(which(cluster_spotname_220_tot[[i]]["cluster dis-bas_220"] == 3))
  names(cluster3_dis_bas_220)[i] <- spotname_220[i]
}

tab_conting_dis_bas_220 <- matrix(nrow = (length(levels(
  cluster_spotname_220_tot[["53003_0_00"]][["cluster dis-bas_220"]]))+2),
                                  ncol = (length(spotname_220)+1))

tab_conting_dis_bas_220[1,] <- c("cluster",spotname_220)
tab_conting_dis_bas_220[2:5,1] <- c(1:3, "TOTALE")

row.names(tab_conting_dis_bas_220) <- tab_conting_dis_bas_220[,1]
tab_conting_dis_bas_220 <- tab_conting_dis_bas_220[,-1]
colnames(tab_conting_dis_bas_220) <- tab_conting_dis_bas_220[1,]
tab_conting_dis_bas_220 <- tab_conting_dis_bas_220[-1,]
tab_conting_dis_bas_220[1:4,] <- rbind(cluster1_dis_bas_220,cluster2_dis_bas_220,cluster3_dis_bas_220,
                                       cluster1_dis_bas_220+cluster2_dis_bas_220+cluster3_dis_bas_220)

tab_conting_dis_bas_220 <- t(tab_conting_dis_bas_220)
storage.mode(tab_conting_dis_bas_220) <- "numeric"

tab_conting_dis_bas_220_freq_intraspotname <- prop.table(tab_conting_dis_bas_220[,-4],margin = 1)

tab_conting_dis_bas_220 <- as.data.frame(tab_conting_dis_bas_220)
kable(tab_conting_dis_bas_220, format='latex', caption = "Contingency tables for 220ms' group",
      label = "Tabella contingenza 220.png")
tab_conting_dis_bas_220_freq_intraspotname <- as.data.frame(tab_conting_dis_bas_220_freq_intraspotname)
kable(tab_conting_dis_bas_220_freq_intraspotname, format='latex',
      caption = "Contingency frequency tables for 220ms' group",
      label = "Tabella contingenza intraspotname 220")

write.csv2(tab_conting_dis_bas_220, file="tabella contingenza distance based 220.csv",
           quote=F, na="", row.names=T, col.names=T)
write.csv2(tab_conting_dis_bas_220_freq_intraspotname,
           file="tabella contingenza distance based 220 frequenza intraspotname.csv",
           quote=F, na="", row.names=T, col.names=T)


#creazione tabella di contingenza per clustering distance based-spotname 300ms
cluster1_dis_bas_300 <- vector()
cluster2_dis_bas_300 <- vector()
cluster3_dis_bas_300 <- vector()
cluster4_dis_bas_300 <- vector()

for (i in 1:length(spotname_300)) {

  cluster1_dis_bas_300 [i] <- length(which(cluster_spotname_300_tot[[i]]["cluster dis-bas_300"] == 1))
  names(cluster1_dis_bas_300)[i] <- spotname_300[i]

  cluster2_dis_bas_300 [i] <- length(which(cluster_spotname_300_tot[[i]]["cluster dis-bas_300"] == 2))
  names(cluster2_dis_bas_300)[i] <- spotname_300[i]

  cluster3_dis_bas_300 [i] <- length(which(cluster_spotname_300_tot[[i]]["cluster dis-bas_300"] == 3))
  names(cluster3_dis_bas_300)[i] <- spotname_300[i]

  cluster4_dis_bas_300 [i] <- length(which(cluster_spotname_300_tot[[i]]["cluster dis-bas_300"] == 4))
  names(cluster4_dis_bas_300)[i] <- spotname_300[i]
}

tab_conting_dis_bas_300 <- matrix(nrow = (length(levels(
  cluster_spotname_300_tot[["53001_0_00"]][["cluster dis-bas_300"]]))+2),
                                  ncol = (length(spotname_300)+1))
tab_conting_dis_bas_300[1,] <- c("cluster",spotname_300)
tab_conting_dis_bas_300[2:6,1] <- c(1:4, "TOTALE")

row.names(tab_conting_dis_bas_300)<- tab_conting_dis_bas_300[,1]
tab_conting_dis_bas_300<- tab_conting_dis_bas_300[,-1]
colnames(tab_conting_dis_bas_300)<- tab_conting_dis_bas_300[1,]
tab_conting_dis_bas_300<- tab_conting_dis_bas_300[-1,]

tab_conting_dis_bas_300 [1:5,] <- rbind(cluster1_dis_bas_300,cluster2_dis_bas_300,
                                        cluster3_dis_bas_300,cluster4_dis_bas_300,
                                        (cluster1_dis_bas_300+cluster2_dis_bas_300+
                                           cluster3_dis_bas_300+cluster4_dis_bas_300))
tab_conting_dis_bas_300 <- t(tab_conting_dis_bas_300)
```

```r
storage.mode(tab_conting_dis_bas_300) <- "numeric"

tab_conting_dis_bas_300_freq_intraspotname <- prop.table(tab_conting_dis_bas_300[,-5],margin = 1)


tab_conting_dis_bas_300 <- as.data.frame(tab_conting_dis_bas_300)
kable(tab_conting_dis_bas_300, format='latex', caption = "Contingency tables for 300ms' group",
      label = "Tabella contingenza 300")
tab_conting_dis_bas_300_freq_intraspotname <- as.data.frame(tab_conting_dis_bas_300_freq_intraspotname)
kable(tab_conting_dis_bas_300_freq_intraspotname, format='latex',
      caption = "Contingency frequency tables for 300ms' group",
      label = "Tabella contingenza intraspotname 300")

write.csv2(tab_conting_dis_bas_300, file="tabella contingenza distance based 300.csv",
           quote=F, na="", row.names=T, col.names=T)
write.csv2(tab_conting_dis_bas_300_freq_intraspotname,
           file="tabella contingenza distance based 300 frequenza intraspotname.csv",
           quote=F, na="", row.names=T, col.names=T)


#Costituzione dei 3 gruppi
#1: 19-21-35-37-39-41-43-45-49-51-702
#2: 3-9-23-27-29-56-703
#3: 1-7-11-13-15-25-33
spotname <- c(spotname_300,spotname_220)
spotname <- factor(spotname)
spotname <- levels(spotname)

spotname_1 <- spotname[c(8,9,15:22,24)]
spotname_2 <- spotname[c(2,4,10,12,13,23,25)]
spotname_3 <- spotname[c(1,3,5:7,11,14)]

kable(spotname_1, format='latex', caption = "Spot names of Group 1",
      label = "Group 1")
kable(spotname_2, format='latex', caption = "Spot names of Group 2",
      label = "Group 2")
kable(spotname_3, format='latex', caption = "Spot names of Group 3",
      label = "Group 3")

write.csv2(spotname_1, file="gruppo 1 di spotname.csv", quote=F, na="", row.names=T, col.names=T)
write.csv2(spotname_2, file="gruppo 2 di spotname.csv", quote=F, na="", row.names=T, col.names=T)
write.csv2(spotname_3, file="gruppo 3 di spotname.csv", quote=F, na="", row.names=T, col.names=T)

#Inizio procedura per clustering
#gruppo 1
dateTime_scalari_1 <- vector()

for (i in 1:length(spotname_1)) {
  dateTime_scalari_1_appoggio <- scalari_associazione$dateTime[scalari_associazione$spotName == spotname_1[i]]
  dateTime_scalari_1 <- c(dateTime_scalari_1, dateTime_scalari_1_appoggio)
}
#gruppo 2
dateTime_scalari_2 <- vector()

for (i in 1:length(spotname_2)) {
  dateTime_scalari_2_appoggio <- scalari_associazione$dateTime[scalari_associazione$spotName == spotname_2[i]]
  dateTime_scalari_2 <- c(dateTime_scalari_2, dateTime_scalari_2_appoggio)
}
#gruppo 3
dateTime_scalari_3 <- vector()

for (i in 1:length(spotname_3)) {
  dateTime_scalari_3_appoggio <- scalari_associazione$dateTime[scalari_associazione$spotName == spotname_3[i]]
  dateTime_scalari_3 <- c(dateTime_scalari_3, dateTime_scalari_3_appoggio)
}


#Gruppo 1
#############
resistance_rawdata_1<- data_profili %>%
  filter(id %in% dateTime_scalari_1) %>%
  pivot_wider(names_from = time,
              values_from = ResistanceCurve,
              id_cols = id,
              values_fill = NA)
resistance_rawdata_1[is.na(resistance_rawdata_1)] <- 0

#Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
resistance_rawdata_1 <- t(resistance_rawdata_1)

colnames(resistance_rawdata_1)<- resistance_rawdata_1[1,]
resistance_rawdata_1<- resistance_rawdata_1[-1,]
storage.mode(resistance_rawdata_1) <- "numeric"

#tolgo outliers
resistance_rawdata_1 <- resistance_rawdata_1[,-c(159,316,327)]

X_1 <- resistance_rawdata_1[seq(from = 1, to = 240),]
storage.mode(X_1) <- "numeric"
n_obs_1 <- ncol(X_1)
n_point_1 <- nrow(X_1)
grid_1 <- seq(from = 0, to = 1, length.out = n_point_1)
data_1 <- list(X = X_1, grid = grid_1)

df_long_1 <- data_1$X %>%
```

```
  as.data.frame %>%
  mutate(grid = data_1$grid) %>%
  pivot_longer(-grid, names_to = "observation")

profili_1<- ggplot(df_long_1) +
  geom_line(mapping = aes(x = grid, y = value, group = observation)) +
  ggtitle("Plot of Group 1's DRC")
# profili_1
ggexport(profili_1, filename = "Profili gruppo 1.png")
#############

#Gruppo 2
#############
resistance_rawdata_2 <- data_profili %>%
  filter(id %in% dateTime_scalari_2) %>%
  pivot_wider(names_from = time,
              values_from = ResistanceCurve,
              id_cols = id,
              values_fill = NA)
resistance_rawdata_2[is.na(resistance_rawdata_2)] <- 0

#Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
resistance_rawdata_2 <- t(resistance_rawdata_2)

colnames(resistance_rawdata_2) <- resistance_rawdata_2[1,]
resistance_rawdata_2 <- resistance_rawdata_2[-1,]
storage.mode(resistance_rawdata_2) <- "numeric"

#tolgo outliers
resistance_rawdata_2 <- resistance_rawdata_2[,-c(172)]

X_2 <- resistance_rawdata_2[seq(from = 1, to = 220),]
storage.mode(X_2) <- "numeric"
n_obs_2 <- ncol(X_2)
n_point_2 <- nrow(X_2)
grid_2 <- seq(from = 0, to = 1, length.out = n_point_2)
data_2 <- list(X = X_2, grid = grid_2)

df_long_2 <- data_2$X %>%
  as.data.frame %>%
  mutate(grid = data_2$grid) %>%
  pivot_longer(-grid, names_to = "observation")

profili_2<- ggplot(df_long_2) +
  geom_line(mapping = aes(x = grid, y = value, group = observation)) +
  ggtitle("Plot of Group 2's DRC")
# profili_2
ggexport(profili_2, filename = "Profili gruppo 2.png")
#############

#Gruppo 3
#############
resistance_rawdata_3 <- data_profili %>%
  filter(id %in% dateTime_scalari_3) %>%
  pivot_wider(names_from = time,
              values_from = ResistanceCurve,
              id_cols = id,
              values_fill = NA)
resistance_rawdata_3[is.na(resistance_rawdata_3)] <- 0

#Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
resistance_rawdata_3 <- t(resistance_rawdata_3)

colnames(resistance_rawdata_3) <- resistance_rawdata_3[1,]
resistance_rawdata_3 <- resistance_rawdata_3[-1,]
storage.mode(resistance_rawdata_3) <- "numeric"

#tolgo outliers
resistance_rawdata_3 <- resistance_rawdata_3[,-c(58,192,212)]

X_3 <- resistance_rawdata_3[seq(from = 1, to = 300),]  #time x profili
storage.mode(X_3) <- "numeric"
n_obs_3 <- ncol(X_3)
n_point_3 <- nrow(X_3)
grid_3 <- seq(from = 0, to = 1, length.out = n_point_3)
data_3 <- list(X = X_3, grid = grid_3)

df_long_3 <- data_3$X %>%
  as.data.frame %>%
  mutate(grid = data_3$grid) %>%
  pivot_longer(-grid, names_to = "observation")

profili_3<- ggplot(df_long_3) +
  geom_line(mapping = aes(x = grid, y = value, group = observation)) +
  ggtitle("Plot of Group 3's DRC")
# profili_3
ggexport(profili_3, filename = "Profili gruppo 3.png")
#############

profili_1_2_3 <- ggarrange(profili_1,profili_2,profili_3, nrow = 1, ncol = 3)
# profili_1_2_3
ggexport(profili_1_2_3, filename = "Profili gruppo 1 2 3 a confronto.png")


#CLUSTERING DEI 3 GRUPPI
```

```r
#GRUPPO 1
#########################
#DISTANCE-BASED CLUSTERING
mod4_1 <- distance_ms(data = data_1, num_cluster_seq = num_cluster_seq, met = "other")
mod4_1$sil_opt
cl_4_1 <- mod4_1$mod_opt_sil$clus

clustered_funs4_1 <- get_clustered_funs_df(cl = cl_4_1,
                                           data = data_1,
                                           method = "distance-based")
centroids4_1 <- get_centroids_df(cl = cl_4_1,
                                 data = data_1,
                                 method = "distance-based")

dis_bas_1 <- ggplot(clustered_funs4_1) +
  geom_line(aes(time, Resistance, group = obs, col = cluster))  +
  ggtitle("Distance-based clustering of Group 1's DRC")
# dis_bas_1
dis_bas_cen_1 <- ggplot(centroids4_1) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of distance-based clustering", "\n", "of Group 1's DRC"))
# dis_bas_cen_1

full_distance_based_1 <- ggarrange(dis_bas_1,dis_bas_cen_1, ncol = 1, nrow = 2)
ggexport(full_distance_based_1, filename = "Distance based clustering 1.png")


#FILTERING METHODS

#B-spline basis
mod5_1 <- fil_bspline_ms_nclust(data = data_1, num_cluster_seq = num_cluster_seq, nbasis = 28)
mod5_1$mod_opt$ind_hc$nbclust ## hierarchical
max(mod5_1$mod_opt$ind_km$cluster) ## k-means
mod5_1$mod_opt$mod_opt$G ## model-based

cl_5_hc_1 <- mod5_1$mod_opt$ind_hc$cluster
cl_5_km_1 <- mod5_1$mod_opt$ind_km$cluster
cl_5_mb_1 <- mod5_1$mod_opt$mod_opt$classification

clustered_funs5_hc_1 <- get_clustered_funs_df(cl = cl_5_hc_1,
                                              data = data_1,
                                              method = "filtering B-spline\nhierarchical")
clustered_funs5_km_1 <- get_clustered_funs_df(cl = cl_5_km_1,
                                              data = data_1,
                                              method = "filtering B-spline\nk-means")
clustered_funs5_mb_1 <- get_clustered_funs_df(cl = cl_5_mb_1,
                                              data = data_1,
                                              method = "filtering B-spline\nmodel-based")
centroids5_hc_1 <- get_centroids_df(cl_5_hc_1, data_1, "filtering B-spline\nhierarchical")
centroids5_km_1 <- get_centroids_df(cl_5_km_1, data_1, "filtering B-spline\nk-means")
centroids5_mb_1 <- get_centroids_df(cl_5_mb_1, data_1, "filtering B-spline\nmodel-based")

filt_bspl_hc_1 <- ggplot(clustered_funs5_hc_1) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering B-spline hierarchical clustering of Group 1's DRC")
# filt_bspl_hc_1
filt_bspl_km_1 <- ggplot(clustered_funs5_km_1) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering B-spline k-means clustering of Group 1's DRC")
# filt_bspl_km_1
filt_bspl_mb_1 <- ggplot(clustered_funs5_mb_1) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering B-spline model-based clustering of Group 1's DRC")
# filt_bspl_mb_1
filt_bspl_hc_cen_1 <- ggplot(centroids5_hc_1) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering B-spline hierarchical clustering", "\n", "of Group 1's DRC"))
# filt_bspl_hc_cen_1
filt_bspl_km_cen_1 <- ggplot(centroids5_km_1) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering B-spline k-means clustering", "\n", "of Group 1's DRC"))
# filt_bspl_km_cen_1
filt_bspl_mb_cen_1 <- ggplot(centroids5_mb_1) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering B-spline modelbased clustering", "\n", "of Group 1's DRC"))
# filt_bspl_mb_cen_1

full_filt_bspl_1 <- ggarrange(filt_bspl_hc_1,filt_bspl_hc_cen_1,filt_bspl_km_1,filt_bspl_km_cen_1,
                              filt_bspl_mb_1,filt_bspl_mb_cen_1,ncol=1,nrow=2)
# full_filt_bspl_1
ggexport(full_filt_bspl_1, filename = "Filtering B-spline profili-centroidi group 1.png")
confr_filt_bspl_1 <- ggarrange(filt_bspl_hc_1,filt_bspl_km_1,filt_bspl_mb_1,filt_bspl_hc_cen_1,
                               filt_bspl_km_cen_1,filt_bspl_mb_cen_1,ncol=1,nrow=3)
# confr_filt_bspl_1
ggexport(confr_filt_bspl_1, filename = "Filtering B-spline technique's comparison group 1.png")


#FPCA basis
mod6_1 <- fil_fpca_ss_nbclust(data = data_1, num_cluster_seq = num_cluster_seq, per_comp = 0.8)
mod6_1$mod_opt$ind_hc$nbclust ## hierarchical
max(mod6_1$mod_opt$ind_km$cluster) ## k-means
mod6_1$mod_opt$mod_opt$G ## model-based

cl_6_km_1 <- mod6_1$mod_opt$ind_km$cluster
```

```r
cl_6_hc_1 <- mod6_1$mod_opt$ind_hc$cluster
cl_6_mb_1 <- mod6_1$mod_opt$mod_opt$classification

clustered_funs6_hc_1 <- get_clustered_funs_df(cl = cl_6_hc_1,
                                              data = data_1,
                                              method = "filtering FPCA\nhierarchical")
clustered_funs6_km_1 <- get_clustered_funs_df(cl = cl_6_km_1,
                                              data = data_1,
                                              method = "filtering FPCA\nk-means")
clustered_funs6_mb_1 <- get_clustered_funs_df(cl = cl_6_mb_1,
                                              data = data_1,
                                              method = "filtering FPCA\nmodel-based")
centroids6_hc_1 <- get_centroids_df(cl_6_hc_1, data_1, "filtering FPCA\nhierarchical")
centroids6_km_1 <- get_centroids_df(cl_6_km_1, data_1, "filtering FPCA\nk-means")
centroids6_mb_1 <- get_centroids_df(cl_6_mb_1, data_1, "filtering FPCA\nmodel-based")

filt_fpca_hc_1 <- ggplot(clustered_funs6_hc_1) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering FPCA hierarchical clustering of Group 1's DRC")
# filt_fpca_hc_1
filt_fpca_km_1 <- ggplot(clustered_funs6_km_1) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering FPCA k-means clustering of Group 1's DRC")
# filt_fpca_km_1
filt_fpca_mb_1 <- ggplot(clustered_funs6_mb_1) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering FPCA model-based clustering of Group 1's DRC")
# filt_fpca_mb_1
filt_fpca_hc_cen_1 <- ggplot(centroids6_hc_1) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering FPCA hierarchical clustering", "\n", "of Group 1's DRC"))
# filt_fpca_hc_cen_1
filt_fpca_km_cen_1 <- ggplot(centroids6_km_1) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering FPCA k-means clustering", "\n", "of Group 1's DRC"))
# filt_fpca_km_cen_1
filt_fpca_mb_cen_1 <- ggplot(centroids6_mb_1) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering FPCA model-based clustering", "\n", "of Group 1's DRC"))
# filt_fpca_mb_cen_1

full_filt_fpca_1 <- ggarrange(filt_fpca_hc_1,filt_fpca_hc_cen_1,filt_fpca_km_1,filt_fpca_km_cen_1,
                              filt_fpca_mb_1,filt_fpca_mb_cen_1,ncol=1,nrow=2)
# full_filt_fpca_1
ggexport(full_filt_fpca_1, filename = "Filtering fPCA profili-centroidi group 1.png")
confr_filt_fpca_1 <- ggarrange(filt_fpca_hc_1,filt_fpca_km_1,filt_fpca_mb_1,filt_fpca_hc_cen_1,
                               filt_fpca_km_cen_1,filt_fpca_mb_cen_1,ncol=1,nrow=3)
# confr_filt_fpca_1
ggexport(confr_filt_fpca_1, filename = "Filtering fPCA technique's comparison group 1.png")
#########################


#GRUPPO 2
#########################
#DISTANCE-BASED CLUSTERING
mod4_2 <- distance_ms(data = data_2, num_cluster_seq = num_cluster_seq, met = "other")
mod4_2$sil_opt
cl_4_2 <- mod4_2$mod_opt_sil$clus

clustered_funs4_2 <- get_clustered_funs_df(cl = cl_4_2,
                                           data = data_2,
                                           method = "distance-based")
centroids4_2 <- get_centroids_df(cl = cl_4_2,
                                 data = data_2,
                                 method = "distance-based")

dis_bas_2 <- ggplot(clustered_funs4_2) +
  geom_line(aes(time, Resistance, group = obs, col = cluster))  +
  ggtitle("Distance-based clustering of Group 2's DRC")
# dis_bas_2
dis_bas_cen_2 <- ggplot(centroids4_2) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of distance-based clustering", "\n", "of Group 2's DRC"))
# dis_bas_cen_2

full_distance_based_2 <- ggarrange(dis_bas_2,dis_bas_cen_2, ncol = 1, nrow = 2)
ggexport(full_distance_based_2, filename = "Distance based clustering 2.png")


#FILTERING METHODS

#B-spline basis
mod5_2 <- fil_bspline_ms_nclust(data = data_2, num_cluster_seq = num_cluster_seq, nbasis = 28)
mod5_2$mod_opt$ind_hc$nbclust ## hierarchical
max(mod5_2$mod_opt$ind_km$cluster) ## k-means
mod5_2$mod_opt$mod_opt$G ## model-based

cl_5_hc_2 <- mod5_2$mod_opt$ind_hc$cluster
cl_5_km_2 <- mod5_2$mod_opt$ind_km$cluster
cl_5_mb_2 <- mod5_2$mod_opt$mod_opt$classification

clustered_funs5_hc_2 <- get_clustered_funs_df(cl = cl_5_hc_2,
                                              data = data_2,
                                              method = "filtering B-spline\nhierarchical")
clustered_funs5_km_2 <- get_clustered_funs_df(cl = cl_5_km_2,
```

```
                                          data = data_2,
                                          method = "filtering B-spline\nk-means")
clustered_funs5_mb_2 <- get_clustered_funs_df(cl = cl_5_mb_2,
                                          data = data_2,
                                          method = "filtering B-spline\nmodel-based")
centroids5_hc_2 <- get_centroids_df(cl_5_hc_2, data_2, "filtering B-spline\nhierarchical")
centroids5_km_2 <- get_centroids_df(cl_5_km_2, data_2, "filtering B-spline\nk-means")
centroids5_mb_2 <- get_centroids_df(cl_5_mb_2, data_2, "filtering B-spline\nmodel-based")

filt_bspl_hc_2 <- ggplot(clustered_funs5_hc_2) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering B-spline hierarchical clustering of Group 2's DRC")
# filt_bspl_hc_2
filt_bspl_km_2 <- ggplot(clustered_funs5_km_2) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering B-spline k-means clustering of Group 2's DRC")
# filt_bspl_km_2
filt_bspl_mb_2 <- ggplot(clustered_funs5_mb_2) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering B-spline model-based clustering of Group 2's DRC")
# filt_bspl_mb_2
filt_bspl_hc_cen_2 <- ggplot(centroids5_hc_2) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering B-spline hierarchical clustering", "\n", "of Group 2's DRC"))
# filt_bspl_hc_cen_2
filt_bspl_km_cen_2 <- ggplot(centroids5_km_2) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering B-spline k-means clustering", "\n", "of Group 2's DRC"))
# filt_bspl_km_cen_2
filt_bspl_mb_cen_2 <- ggplot(centroids5_mb_2) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering B-spline modelbased clustering", "\n", "of Group 2's DRC"))
# filt_bspl_mb_cen_2

full_filt_bspl_2 <- ggarrange(filt_bspl_hc_2,filt_bspl_hc_cen_2,filt_bspl_km_2,filt_bspl_km_cen_2,
                              filt_bspl_mb_2,filt_bspl_mb_cen_2,ncol=1,nrow=2)
# full_filt_bspl_2
ggexport(full_filt_bspl_2, filename = "Filtering B-spline profili-centroidi group 2.png")
confr_filt_bspl_2 <- ggarrange(filt_bspl_hc_2,filt_bspl_km_2,filt_bspl_mb_2,filt_bspl_hc_cen_2,
                               filt_bspl_km_cen_2,filt_bspl_mb_cen_2,ncol=1,nrow=3)
# confr_filt_bspl_2
ggexport(confr_filt_bspl_2, filename = "Filtering B-spline technique's comparison group 2.png")


#FPCA basis
mod6_2 <- fil_fpca_ss_nbclust(data = data_2, num_cluster_seq = num_cluster_seq, per_comp = 0.8)
mod6_2$mod_opt$ind_hc$nbclust ## hierarchical
max(mod6_2$mod_opt$ind_km$cluster) ## k-means
mod6_2$mod_opt$mod_opt$G ## model-based

cl_6_km_2 <- mod6_2$mod_opt$ind_km$cluster
cl_6_hc_2 <- mod6_2$mod_opt$ind_hc$cluster
cl_6_mb_2 <- mod6_2$mod_opt$mod_opt$classification

clustered_funs6_hc_2 <- get_clustered_funs_df(cl = cl_6_hc_2,
                                          data = data_2,
                                          method = "filtering FPCA\nhierarchical")
clustered_funs6_km_2 <- get_clustered_funs_df(cl = cl_6_km_2,
                                          data = data_2,
                                          method = "filtering FPCA\nk-means")
clustered_funs6_mb_2 <- get_clustered_funs_df(cl = cl_6_mb_2,
                                          data = data_2,
                                          method = "filtering FPCA\nmodel-based")
centroids6_hc_2 <- get_centroids_df(cl_6_hc_2, data_2, "filtering FPCA\nhierarchical")
centroids6_km_2 <- get_centroids_df(cl_6_km_2, data_2, "filtering FPCA\nk-means")
centroids6_mb_2 <- get_centroids_df(cl_6_mb_2, data_2, "filtering FPCA\nmodel-based")

filt_fpca_hc_2 <- ggplot(clustered_funs6_hc_2) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering FPCA hierarchical clustering of Group 2's DRC")
# filt_fpca_hc_2
filt_fpca_km_2 <- ggplot(clustered_funs6_km_2) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering FPCA k-means clustering of Group 2's DRC")
# filt_fpca_km_2
filt_fpca_mb_2 <- ggplot(clustered_funs6_mb_2) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering FPCA model-based clustering of Group 2's DRC")
# filt_fpca_mb_2
filt_fpca_hc_cen_2 <- ggplot(centroids6_hc_2) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering FPCA hierarchical clustering", "\n", "of Group 2's DRC"))
# filt_fpca_hc_cen_2
filt_fpca_km_cen_2 <- ggplot(centroids6_km_2) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering FPCA k-means clustering", "\n", "of Group 2's DRC"))
# filt_fpca_km_cen_2
filt_fpca_mb_cen_2 <- ggplot(centroids6_mb_2) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering FPCA model-based clustering", "\n", "of Group 2's DRC"))
# filt_fpca_mb_cen_2

full_filt_fpca_2 <- ggarrange(filt_fpca_hc_2,filt_fpca_hc_cen_2,filt_fpca_km_2,filt_fpca_km_cen_2,
                              filt_fpca_mb_2,filt_fpca_mb_cen_2,ncol=1,nrow=2)
# full_filt_fpca_2
```

```r
ggexport(full_filt_fpca_2, filename = "Filtering fPCA profili-centroidi group 2.png")
confr_filt_fpca_2 <- ggarrange(filt_fpca_hc_2,filt_fpca_km_2,filt_fpca_mb_2,filt_fpca_hc_cen_2,
                               filt_fpca_km_cen_2,filt_fpca_mb_cen_2,ncol=1,nrow=3)
# confr_filt_fpca_2
ggexport(confr_filt_fpca_2, filename = "Filtering fPCA technique's comparison group 2.png")
#########################

#GRUPPO 3
#########################
#DISTANCE-BASED CLUSTERING
mod4_3 <- distance_ms(data = data_3, num_cluster_seq = num_cluster_seq, met = "other")
mod4_3$sil_opt
cl_4_3 <- mod4_3$mod_opt_sil$clus

clustered_funs4_3 <- get_clustered_funs_df(cl = cl_4_3,
                                           data = data_3,
                                           method = "distance-based")
centroids4_3 <- get_centroids_df(cl = cl_4_3,
                                 data = data_3,
                                 method = "distance-based")

dis_bas_3 <- ggplot(clustered_funs4_3) +
  geom_line(aes(time, Resistance, group = obs, col = cluster))  +
  ggtitle("Distance-based clustering of Group 3's DRC")
# dis_bas_3
#plotto i centroidi dei cluster
dis_bas_cen_3 <- ggplot(centroids4_3) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of distance-based clustering", "\n", "of Group 3's DRC"))
# dis_bas_cen_3

full_distance_based_3 <- ggarrange(dis_bas_3,dis_bas_cen_3, ncol = 1, nrow = 2)
ggexport(full_distance_based_3, filename = "Distance based clustering 3.png")


#FILTERING METHODS

#B-spline basis
mod5_3 <- fil_bspline_ms_nclust(data = data_3, num_cluster_seq = num_cluster_seq, nbasis = 28)
mod5_3$mod_opt$ind_hc$nbclust ## hierarchical
max(mod5_3$mod_opt$ind_km$cluster) ## k-means
mod5_3$mod_opt$mod_opt$G ## model-based

cl_5_hc_3 <- mod5_3$mod_opt$ind_hc$cluster
cl_5_km_3 <- mod5_3$mod_opt$ind_km$cluster
cl_5_mb_3 <- mod5_3$mod_opt$mod_opt$classification

clustered_funs5_hc_3 <- get_clustered_funs_df(cl = cl_5_hc_3,
                                              data = data_3,
                                              method = "filtering B-spline\nhierarchical")
clustered_funs5_km_3 <- get_clustered_funs_df(cl = cl_5_km_3,
                                              data = data_3,
                                              method = "filtering B-spline\nk-means")
clustered_funs5_mb_3 <- get_clustered_funs_df(cl = cl_5_mb_3,
                                              data = data_3,
                                              method = "filtering B-spline\nmodel-based")
centroids5_hc_3 <- get_centroids_df(cl_5_hc_3, data_3, "filtering B-spline\nhierarchical")
centroids5_km_3 <- get_centroids_df(cl_5_km_3, data_3, "filtering B-spline\nk-means")
centroids5_mb_3 <- get_centroids_df(cl_5_mb_3, data_3, "filtering B-spline\nmodel-based")

filt_bspl_hc_3 <- ggplot(clustered_funs5_hc_3) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering B-spline hierarchical clustering of Group 3's DRC")
# filt_bspl_hc_3
filt_bspl_km_3 <- ggplot(clustered_funs5_km_3) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering B-spline k-means clustering of Group 3's DRC")
# filt_bspl_km_3
filt_bspl_mb_3 <- ggplot(clustered_funs5_mb_3) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering B-spline model-based clustering of Group 3's DRC")
# filt_bspl_mb_3

filt_bspl_hc_cen_3 <- ggplot(centroids5_hc_3) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering B-spline hierarchical clustering", "\n", "of Group 3's DRC"))
# filt_bspl_hc_cen_3
filt_bspl_km_cen_3 <- ggplot(centroids5_km_3) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering B-spline k-means clustering", "\n", "of Group 3's DRC"))
# filt_bspl_km_cen_3
filt_bspl_mb_cen_3 <- ggplot(centroids5_mb_3) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering B-spline modelbased clustering", "\n", "of Group 3's DRC"))
# filt_bspl_mb_cen_3

full_filt_bspl_3 <- ggarrange(filt_bspl_hc_3,filt_bspl_hc_cen_3,filt_bspl_km_3,filt_bspl_km_cen_3,
                              filt_bspl_mb_3,filt_bspl_mb_cen_3,ncol=1,nrow=2)
# full_filt_bspl_3
ggexport(full_filt_bspl_3, filename = "Filtering B-spline profili-centroidi group 3.png")
confr_filt_bspl_3 <- ggarrange(filt_bspl_hc_3,filt_bspl_km_3,filt_bspl_mb_3,filt_bspl_hc_cen_3,
                               filt_bspl_km_cen_3,filt_bspl_mb_cen_3,ncol=1,nrow=3)
# confr_filt_bspl_3
ggexport(confr_filt_bspl_3, filename = "Filtering B-spline technique's comparison group 3.png")
```

```r
#FPCA basis
mod6_3 <- fil_fpca_ss_nbclust(data = data_3, num_cluster_seq = num_cluster_seq, per_comp = 0.8)
mod6_3$mod_opt$ind_hc$nbclust ## hierarchical
max(mod6_3$mod_opt$ind_km$cluster) ## k-means
mod6_3$mod_opt$mod_opt$G ## model-based

cl_6_km_3 <- mod6_3$mod_opt$ind_km$cluster
cl_6_hc_3 <- mod6_3$mod_opt$ind_hc$cluster
cl_6_mb_3 <- mod6_3$mod_opt$mod_opt$classification

clustered_funs6_hc_3 <- get_clustered_funs_df(cl = cl_6_hc_3,
                                              data = data_3,
                                              method = "filtering FPCA\nhierarchical")
clustered_funs6_km_3 <- get_clustered_funs_df(cl = cl_6_km_3,
                                              data = data_3,
                                              method = "filtering FPCA\nk-means")
clustered_funs6_mb_3 <- get_clustered_funs_df(cl = cl_6_mb_3,
                                              data = data_3,
                                              method = "filtering FPCA\nmodel-based")
centroids6_hc_3 <- get_centroids_df(cl_6_hc_3, data_3, "filtering FPCA\nhierarchical")
centroids6_km_3 <- get_centroids_df(cl_6_km_3, data_3, "filtering FPCA\nk-means")
centroids6_mb_3 <- get_centroids_df(cl_6_mb_3, data_3, "filtering FPCA\nmodel-based")

filt_fpca_hc_3 <- ggplot(clustered_funs6_hc_3) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering FPCA hierarchical clustering of Group 3's DRC")
# filt_fpca_hc_3
filt_fpca_km_3 <- ggplot(clustered_funs6_km_3) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering FPCA k-means clustering of Group 3's DRC")
# filt_fpca_km_3
filt_fpca_mb_3 <- ggplot(clustered_funs6_mb_3) +
  geom_line(aes(time, Resistance, group = obs, col = cluster)) +
  ggtitle("Filtering FPCA model-based clustering of Group 3's DRC")
# filt_fpca_mb_3
filt_fpca_hc_cen_3 <- ggplot(centroids6_hc_3) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering FPCA hierarchical clustering", "\n", "of Group 3's DRC"))
# filt_fpca_hc_cen_3
filt_fpca_km_cen_3 <- ggplot(centroids6_km_3) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering FPCA k-means clustering", "\n", "of Group 3's DRC"))
# filt_fpca_km_cen_3
filt_fpca_mb_cen_3 <- ggplot(centroids6_mb_3) +
  geom_line(aes(time, Resistance, col = cluster)) +
  ggtitle(paste("Centroids of filtering FPCA model-based clustering", "\n", "of Group 3's DRC"))
# filt_fpca_mb_cen_3

full_filt_fpca_3 <- ggarrange(filt_fpca_hc_3,filt_fpca_hc_cen_3,filt_fpca_km_3,filt_fpca_km_cen_3,
                              filt_fpca_mb_3,filt_fpca_mb_cen_3,ncol=1,nrow=2)
# full_filt_fpca_3
ggexport(full_filt_fpca_3, filename = "Filtering fPCA profili-centroidi group 3.png")
confr_filt_fpca_3 <- ggarrange(filt_fpca_hc_3,filt_fpca_km_3,filt_fpca_mb_3,filt_fpca_hc_cen_3,
                               filt_fpca_km_cen_3,filt_fpca_mb_cen_3,ncol=1,nrow=3)
# confr_filt_fpca_3
ggexport(confr_filt_fpca_3, filename = "Filtering fPCA technique's comparison group 3.png")
##########################




#DATA SET COMPLETO, SECONDA ANALISI

#Carico dataset dei nuovi dati
######################
data_profili_new <- import("D:/Gianluca/Desktop/TESI/DATI/SCC100R01_Profili_Weld_Group6_2021.csv")
scalari_associazione_new <- import("D:/Gianluca/Desktop/TESI/DATI/SCC100R01_Scalari_Weld_Group6_2021.csv")
gruppi_spotname <- import("D:/Gianluca/Desktop/TESI/DATI/MaseratiSCC100R01SpotNameGruppo.csv")

spotname_new <- factor(scalari_associazione_new$spotName)
spotname_new <- levels(spotname_new)

# dateTime_new_tot <- list()
resistance_rawdata_spotname_new_tot <- list()

for (i in 1:length(spotname_new)) {

  dateTime_new <- scalari_associazione_new$dateTime[scalari_associazione_new$spotName == spotname_new[i]]
  resistance_rawdata_spotname_new <- data_profili_new %>%
    filter(id %in% dateTime_new) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_new<- t(resistance_rawdata_spotname_new)

  colnames(resistance_rawdata_spotname_new)<- resistance_rawdata_spotname_new[1,]
  resistance_rawdata_spotname_new<- resistance_rawdata_spotname_new[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_new[is.na(resistance_rawdata_spotname_new)] <- 0
```

```
  storage.mode(resistance_rawdata_spotname_new) <- "numeric"
  # dateTime_new_tot[[i]] <- dateTime_new
  resistance_rawdata_spotname_new_tot[[i]] <- resistance_rawdata_spotname_new
  names(resistance_rawdata_spotname_new_tot)[[i]] <- spotname_new[i]
}
######################

#Levo outlyer e taglio ai giusti ms (240,280,300,320,340)
###################
resistance_rawdata_spotname_new_tot[[1]]<-
  resistance_rawdata_spotname_new_tot[[1]][1:300,-c(191,391,448,621,784,992,1032,1087,1117,1132,1135,1250,1274,1406,1467,1517,1561,1586,1596,1
resistance_rawdata_spotname_new_tot[[2]]<-
  resistance_rawdata_spotname_new_tot[[2]][1:240,]
resistance_rawdata_spotname_new_tot[[3]]<-
  resistance_rawdata_spotname_new_tot[[3]][1:320,-c(387,520,530,587,897,898,1293)]
resistance_rawdata_spotname_new_tot[[4]]<-
  resistance_rawdata_spotname_new_tot[[4]][1:280,]
resistance_rawdata_spotname_new_tot[[5]]<-
  resistance_rawdata_spotname_new_tot[[5]][1:300,-412]
resistance_rawdata_spotname_new_tot[[6]]<-
  resistance_rawdata_spotname_new_tot[[6]][1:320,-c(1548,1578,1706)]
resistance_rawdata_spotname_new_tot[[7]]<-
  resistance_rawdata_spotname_new_tot[[7]][1:320,]
resistance_rawdata_spotname_new_tot[[8]]<-
  resistance_rawdata_spotname_new_tot[[8]][1:300,]
resistance_rawdata_spotname_new_tot[[9]]<-
  resistance_rawdata_spotname_new_tot[[9]][1:240,]
resistance_rawdata_spotname_new_tot[[10]]<-
  resistance_rawdata_spotname_new_tot[[10]][1:220,]
resistance_rawdata_spotname_new_tot[[11]]<-
  resistance_rawdata_spotname_new_tot[[11]][1:320,-c(683,1051,1201,1212,1218)]
resistance_rawdata_spotname_new_tot[[12]]<-
  resistance_rawdata_spotname_new_tot[[12]][1:320,-c(363,700,917,1026,1036)]
resistance_rawdata_spotname_new_tot[[13]]<-
  resistance_rawdata_spotname_new_tot[[13]][1:240,-c(788,929,932,1115,1178,1188)]
resistance_rawdata_spotname_new_tot[[14]]<-
  resistance_rawdata_spotname_new_tot[[14]][1:340,-c(528,842)]
resistance_rawdata_spotname_new_tot[[15]]<-
  resistance_rawdata_spotname_new_tot[[15]][1:240,-c(386,768)]
resistance_rawdata_spotname_new_tot[[16]]<-
  resistance_rawdata_spotname_new_tot[[16]][1:240,-c(165,877,1069,1223)]
resistance_rawdata_spotname_new_tot[[17]]<-
  resistance_rawdata_spotname_new_tot[[17]][1:240,-c(1106,1161,1170,1239,1461)]
resistance_rawdata_spotname_new_tot[[18]]<-
  resistance_rawdata_spotname_new_tot[[18]][1:240,]
resistance_rawdata_spotname_new_tot[[19]]<-
  resistance_rawdata_spotname_new_tot[[19]][1:240,-c(269,567,620)]
resistance_rawdata_spotname_new_tot[[20]]<-
  resistance_rawdata_spotname_new_tot[[20]][1:240,-719]
resistance_rawdata_spotname_new_tot[[21]]<-
  resistance_rawdata_spotname_new_tot[[21]][1:240,-c(577,627)]
resistance_rawdata_spotname_new_tot[[22]]<-
  resistance_rawdata_spotname_new_tot[[22]][1:240,]
resistance_rawdata_spotname_new_tot[[23]]<-
  resistance_rawdata_spotname_new_tot[[23]][1:240,-633]
resistance_rawdata_spotname_new_tot[[24]]<-
  resistance_rawdata_spotname_new_tot[[24]][1:320,]
resistance_rawdata_spotname_new_tot[[25]]<-
  resistance_rawdata_spotname_new_tot[[25]][1:320,]
###################

#Costruisco plot dei 25 spotname
######################
data_spotname_new_tot <- list()
# df_long_spotname_new_tot <- list()
# profili_spotname_new_tot <- list()

for (i in 1:length(spotname_new)) {
  X_spotname_new <- resistance_rawdata_spotname_new_tot[[i]]
  storage.mode(X_spotname_new) <- "numeric"
  n_obs_spotname_new <- ncol(X_spotname_new)
  n_point_spotname_new <- nrow(X_spotname_new)
  grid_spotname_new <- seq(from = 0, to = 1, length.out = n_point_spotname_new)
  data_spotname_new <- list(X = X_spotname_new, grid = grid_spotname_new)

  data_spotname_new_tot[[i]] <- data_spotname_new

  df_long_spotname_new <- data_spotname_new$X %>%
    as.data.frame %>%
    mutate(grid = data_spotname_new$grid) %>%
    pivot_longer(-grid, names_to = "observation")

  # df_long_spotname_new_tot[[i]] <- df_long_spotname_new

  profili_spotname_new <- ggplot(df_long_spotname_new) +
    geom_line(mapping = aes(x = grid, y = value, group = observation)) +
    ggtitle(paste("Plot of Spotname",spotname_new[i],"'s DRC","\n","cut at", n_point_spotname_new, "ms"))
  # print(profili_spotname_new)
  # ggexport(profili_spotname_new, filename = paste("ggPlot spotname",spotname_new[i], "'s DRC.png"))

  # profili_spotname_new_tot[[i]] <- profili_spotname_new
}
######################
```

```r
#CLUSTERING
num_cluster_seq_spotname <- 2:4

#DISTANCE-BASED CLUSTERING - Prima clusterizzazione
#creo 2 cluster
#####################
clustered_funs4_spotname_new_tot <- list()

for (i in 1:25) {
  print(i)
  mod4_spotname_new <-distance_ms(data = data_spotname_new_tot[[i]], num_cluster_seq = 2, met = "other")
  mod4_spotname_new$sil_opt
  cl_4_spotname_new <- mod4_spotname_new$mod_opt_sil$clus

  clustered_funs4_spotname_new <- get_clustered_funs_df(cl = cl_4_spotname_new,
                                                        data = data_spotname_new_tot[[i]],
                                                        method = "distance-based")
  centroids4_spotname_new <- get_centroids_df(cl = cl_4_spotname_new,
                                              data = data_spotname_new_tot[[i]],
                                              method = "distance-based")
  clustered_funs4_spotname_new_tot[[i]] <- clustered_funs4_spotname_new

  dis_bas_spotname_new <- ggplot(clustered_funs4_spotname_new) +
    geom_line(aes(time, Resistance, group = obs, col = cluster))  +
    ggtitle(paste("Distance-based clustering of spotname ",spotname_new[i],"'s DRC"))
  # print(dis_bas_spotname)
  dis_bas_cen_spotname_new <- ggplot(centroids4_spotname_new) +
    geom_line(aes(time, Resistance, col = cluster)) +
    ggtitle(paste("Centroids of Distance-based clustering", "\n","of spotname ",spotname_new[i],"'s DRC"))
  # print(dis_bas_cen_spotname_new)

  full_distance_based_spotname_new <- ggarrange(dis_bas_spotname_new,dis_bas_cen_spotname_new,
                                                ncol = 1, nrow = 2)
  # ggexport(full_distance_based_spotname_new,
  #          filename = paste("Distance based clustering of Spotname",spotname_new[i],"DRC.png"))
}
#####################

#suddivido i cluster di distance based in due oggetti diversi
#################
profili_cluster_1_dis_bas_tot <- list()
profili_cluster_2_dis_bas_tot <- list()
scalari_cluster_1_dis_bas_tot <- list()
scalari_cluster_2_dis_bas_tot <- list()

for (i in 1:25) {

  profili_cluster_dis_bas <- clustered_funs4_spotname_new_tot[[i]][seq(from = 1, to = ncol(resistance_rawdata_spotname_new_tot[[i]])),]
  profili_cluster_1_dis_bas <- profili_cluster_dis_bas[profili_cluster_dis_bas$cluster == 1,]
  profili_cluster_2_dis_bas <- profili_cluster_dis_bas[profili_cluster_dis_bas$cluster == 2,]

  #cambio nome a colonna delle obs e di cluster
  colnames(profili_cluster_1_dis_bas)[2] <- "dateTime"
  colnames(profili_cluster_1_dis_bas)[4] <- "cluster dis-bas"

  colnames(profili_cluster_2_dis_bas)[2] <- "dateTime"
  colnames(profili_cluster_2_dis_bas)[4] <- "cluster dis-bas"

  #unisco colonna relativa al cluster di appartenenza
  scalari_cluster_1_dis_bas <- merge(scalari_associazione_new,profili_cluster_1_dis_bas,by = ("dateTime"))
  scalari_cluster_2_dis_bas <- merge(scalari_associazione_new,profili_cluster_2_dis_bas,by = ("dateTime"))

  profili_cluster_1_dis_bas_tot[[i]] <- profili_cluster_1_dis_bas
  profili_cluster_2_dis_bas_tot[[i]] <- profili_cluster_2_dis_bas

  scalari_cluster_1_dis_bas_tot[[i]] <- scalari_cluster_1_dis_bas
  scalari_cluster_2_dis_bas_tot[[i]] <- scalari_cluster_2_dis_bas
}
#################

#creo matrici resistance rawdata
#################
resistance_rawdata_spotname_cluster_1_tot <- list()
resistance_rawdata_spotname_cluster_2_tot <- list()

for (i in 1:25) {

  resistance_rawdata_spotname_cluster_1 <- data_profili_new %>%
    filter(id %in% scalari_cluster_1_dis_bas_tot[[i]]$dateTime ) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_cluster_1<- t(resistance_rawdata_spotname_cluster_1)

  colnames(resistance_rawdata_spotname_cluster_1)<- resistance_rawdata_spotname_cluster_1[1,]
  resistance_rawdata_spotname_cluster_1<- resistance_rawdata_spotname_cluster_1[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_cluster_1[is.na(resistance_rawdata_spotname_cluster_1)] <- 0

  storage.mode(resistance_rawdata_spotname_cluster_1) <- "numeric"       #la matrice veniva salvata come character, cosi diventa numeric
  resistance_rawdata_spotname_cluster_1_tot[[i]] <- resistance_rawdata_spotname_cluster_1
```

```
  names(resistance_rawdata_spotname_cluster_1_tot)[[i]] <- spotname_new[i]


  resistance_rawdata_spotname_cluster_2 <- data_profili_new %>%
    filter(id %in% scalari_cluster_2_dis_bas_tot[[i]]$dateTime) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_cluster_2<- t(resistance_rawdata_spotname_cluster_2)

  colnames(resistance_rawdata_spotname_cluster_2)<- resistance_rawdata_spotname_cluster_2[1,]
  resistance_rawdata_spotname_cluster_2<- resistance_rawdata_spotname_cluster_2[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_cluster_2[is.na(resistance_rawdata_spotname_cluster_2)] <- 0

  storage.mode(resistance_rawdata_spotname_cluster_2) <- "numeric"       #la matrice veniva salvata come character, cosi diventa numeric
  resistance_rawdata_spotname_cluster_2_tot[[i]] <- resistance_rawdata_spotname_cluster_2
  names(resistance_rawdata_spotname_cluster_2_tot)[[i]] <- spotname_new[i]
}
##################
#taglio ai giusti ms (240,280,300,320,340)
####################
resistance_rawdata_spotname_cluster_1_tot[[1]]<-
  resistance_rawdata_spotname_cluster_1_tot[[1]][1:300,]
resistance_rawdata_spotname_cluster_1_tot[[2]]<-
  resistance_rawdata_spotname_cluster_1_tot[[2]][1:240,]
resistance_rawdata_spotname_cluster_1_tot[[3]]<-
  resistance_rawdata_spotname_cluster_1_tot[[3]][1:320,]
resistance_rawdata_spotname_cluster_1_tot[[4]]<-
  resistance_rawdata_spotname_cluster_1_tot[[4]][1:280,]
resistance_rawdata_spotname_cluster_1_tot[[5]]<-
  resistance_rawdata_spotname_cluster_1_tot[[5]][1:300,]
resistance_rawdata_spotname_cluster_1_tot[[6]]<-
  resistance_rawdata_spotname_cluster_1_tot[[6]][1:320,]
resistance_rawdata_spotname_cluster_1_tot[[7]]<-
  resistance_rawdata_spotname_cluster_1_tot[[7]][1:320,]
resistance_rawdata_spotname_cluster_1_tot[[8]]<-
  resistance_rawdata_spotname_cluster_1_tot[[8]][1:300,]
resistance_rawdata_spotname_cluster_1_tot[[9]]<-
  resistance_rawdata_spotname_cluster_1_tot[[9]][1:240,]
resistance_rawdata_spotname_cluster_1_tot[[10]]<-
  resistance_rawdata_spotname_cluster_1_tot[[10]][1:220,]
resistance_rawdata_spotname_cluster_1_tot[[11]]<-
  resistance_rawdata_spotname_cluster_1_tot[[11]][1:320,]
resistance_rawdata_spotname_cluster_1_tot[[12]]<-
  resistance_rawdata_spotname_cluster_1_tot[[12]][1:320,]
resistance_rawdata_spotname_cluster_1_tot[[13]]<-
  resistance_rawdata_spotname_cluster_1_tot[[13]][1:240,]
resistance_rawdata_spotname_cluster_1_tot[[14]]<-
  resistance_rawdata_spotname_cluster_1_tot[[14]][1:340,]
resistance_rawdata_spotname_cluster_1_tot[[15]]<-
  resistance_rawdata_spotname_cluster_1_tot[[15]][1:240,]
resistance_rawdata_spotname_cluster_1_tot[[16]]<-
  resistance_rawdata_spotname_cluster_1_tot[[16]][1:240,]
resistance_rawdata_spotname_cluster_1_tot[[17]]<-
  resistance_rawdata_spotname_cluster_1_tot[[17]][1:240,]
resistance_rawdata_spotname_cluster_1_tot[[18]]<-
  resistance_rawdata_spotname_cluster_1_tot[[18]][1:240,]
resistance_rawdata_spotname_cluster_1_tot[[19]]<-
  resistance_rawdata_spotname_cluster_1_tot[[19]][1:240,]
resistance_rawdata_spotname_cluster_1_tot[[20]]<-
  resistance_rawdata_spotname_cluster_1_tot[[20]][1:240,]
resistance_rawdata_spotname_cluster_1_tot[[21]]<-
  resistance_rawdata_spotname_cluster_1_tot[[21]][1:240,]
resistance_rawdata_spotname_cluster_1_tot[[22]]<-
  resistance_rawdata_spotname_cluster_1_tot[[22]][1:240,]
resistance_rawdata_spotname_cluster_1_tot[[23]]<-
  resistance_rawdata_spotname_cluster_1_tot[[23]][1:240,]
resistance_rawdata_spotname_cluster_1_tot[[24]]<-
  resistance_rawdata_spotname_cluster_1_tot[[24]][1:320,]
resistance_rawdata_spotname_cluster_1_tot[[25]]<-
  resistance_rawdata_spotname_cluster_1_tot[[25]][1:320,]

resistance_rawdata_spotname_cluster_2_tot[[1
                                          ]]<-
  resistance_rawdata_spotname_cluster_2_tot[[1]][1:300,]
resistance_rawdata_spotname_cluster_2_tot[[2]]<-
  resistance_rawdata_spotname_cluster_2_tot[[2]][1:240,]
resistance_rawdata_spotname_cluster_2_tot[[3]]<-
  resistance_rawdata_spotname_cluster_2_tot[[3]][1:320,]
resistance_rawdata_spotname_cluster_2_tot[[4]]<-
  resistance_rawdata_spotname_cluster_2_tot[[4]][1:280,]
resistance_rawdata_spotname_cluster_2_tot[[5]]<-
  resistance_rawdata_spotname_cluster_2_tot[[5]][1:300,]
resistance_rawdata_spotname_cluster_2_tot[[6]]<-
  resistance_rawdata_spotname_cluster_2_tot[[6]][1:320,]
resistance_rawdata_spotname_cluster_2_tot[[7]]<-
  resistance_rawdata_spotname_cluster_2_tot[[7]][1:320,]
resistance_rawdata_spotname_cluster_2_tot[[8]]<-
  resistance_rawdata_spotname_cluster_2_tot[[8]][1:300,]
```

```
resistance_rawdata_spotname_cluster_2_tot[[9]]<-
  resistance_rawdata_spotname_cluster_2_tot[[9]][1:240,]
resistance_rawdata_spotname_cluster_2_tot[[10]]<-
  resistance_rawdata_spotname_cluster_2_tot[[10]][1:220,]
resistance_rawdata_spotname_cluster_2_tot[[11]]<-
  resistance_rawdata_spotname_cluster_2_tot[[11]][1:320,]
resistance_rawdata_spotname_cluster_2_tot[[12]]<-
  resistance_rawdata_spotname_cluster_2_tot[[12]][1:320,]
resistance_rawdata_spotname_cluster_2_tot[[13]]<-
  resistance_rawdata_spotname_cluster_2_tot[[13]][1:240,]
resistance_rawdata_spotname_cluster_2_tot[[14]]<-
  resistance_rawdata_spotname_cluster_2_tot[[14]][1:340,]
resistance_rawdata_spotname_cluster_2_tot[[15]]<-
  resistance_rawdata_spotname_cluster_2_tot[[15]][1:240,]
resistance_rawdata_spotname_cluster_2_tot[[16]]<-
  resistance_rawdata_spotname_cluster_2_tot[[16]][1:240,]
resistance_rawdata_spotname_cluster_2_tot[[17]]<-
  resistance_rawdata_spotname_cluster_2_tot[[17]][1:240,]
resistance_rawdata_spotname_cluster_2_tot[[18]]<-
  resistance_rawdata_spotname_cluster_2_tot[[18]][1:240,]
resistance_rawdata_spotname_cluster_2_tot[[19]]<-
  resistance_rawdata_spotname_cluster_2_tot[[19]][1:240,]
resistance_rawdata_spotname_cluster_2_tot[[20]]<-
  resistance_rawdata_spotname_cluster_2_tot[[20]][1:240,]
resistance_rawdata_spotname_cluster_2_tot[[21]]<-
  resistance_rawdata_spotname_cluster_2_tot[[21]][1:240,]
resistance_rawdata_spotname_cluster_2_tot[[22]]<-
  resistance_rawdata_spotname_cluster_2_tot[[22]][1:240,]
resistance_rawdata_spotname_cluster_2_tot[[23]]<-
  resistance_rawdata_spotname_cluster_2_tot[[23]][1:240,]
resistance_rawdata_spotname_cluster_2_tot[[24]]<-
  resistance_rawdata_spotname_cluster_2_tot[[24]][1:320,]
resistance_rawdata_spotname_cluster_2_tot[[25]]<-
  resistance_rawdata_spotname_cluster_2_tot[[25]][1:320,]
####################

#plotto i profili degli spotname relativi ai 2 cluster (passaggio facoltativo)
##################
data_spotname_cluster_1_tot <- list()
data_spotname_cluster_2_tot <- list()

for (i in 1:25) {
  X_spotname_cluster_1 <- resistance_rawdata_spotname_cluster_1_tot[[i]]
  storage.mode(X_spotname_cluster_1) <- "numeric"
  n_obs_spotname_cluster_1 <- ncol(X_spotname_cluster_1)
  n_point_spotname_cluster_1 <- nrow(X_spotname_cluster_1)
  grid_spotname_cluster_1 <- seq(from = 0, to = 1, length.out = n_point_spotname_cluster_1)
  data_spotname_cluster_1 <- list(X = X_spotname_cluster_1, grid = grid_spotname_cluster_1)

  data_spotname_cluster_1_tot[[i]] <- data_spotname_cluster_1

  df_long_spotname_cluster_1 <- data_spotname_cluster_1$X %>%
    as.data.frame %>%
    mutate(grid = data_spotname_cluster_1$grid) %>%
    pivot_longer(-grid, names_to = "observation")

  profili_spotname_cluster_1 <- ggplot(df_long_spotname_cluster_1) +
    geom_line(mapping = aes(x = grid, y = value, group = observation)) +
    ggtitle(paste("Plot of Spotname",spotname_new[i],"'s DRC","\n","of cluster 1 cut at",
                  n_point_spotname_cluster_1, "ms"))
  # print(profili_spotname_cluster_1)
  # ggexport(profili_spotname_cluster_1, filename = paste("ggPlot spotname",spotname_new[i],
  #                                                       "'s DRC of cluster 1.png"))


  X_spotname_cluster_2 <- resistance_rawdata_spotname_cluster_2_tot[[i]]
  storage.mode(X_spotname_cluster_2) <- "numeric"
  n_obs_spotname_cluster_2 <- ncol(X_spotname_cluster_2)
  n_point_spotname_cluster_2 <- nrow(X_spotname_cluster_2)
  grid_spotname_cluster_2 <- seq(from = 0, to = 1, length.out = n_point_spotname_cluster_2)
  data_spotname_cluster_2 <- list(X = X_spotname_cluster_2, grid = grid_spotname_cluster_2)

  data_spotname_cluster_2_tot[[i]] <- data_spotname_cluster_2

  df_long_spotname_cluster_2 <- data_spotname_cluster_2$X %>%
    as.data.frame %>%
    mutate(grid = data_spotname_cluster_2$grid) %>%
    pivot_longer(-grid, names_to = "observation")

  profili_spotname_cluster_2 <- ggplot(df_long_spotname_cluster_2) +
    geom_line(mapping = aes(x = grid, y = value, group = observation)) +
    ggtitle(paste("Plot of Spotname",spotname_new[i],"'s DRC","\n","of cluster 2 cut at",
                  n_point_spotname_cluster_2, "ms"))
  # print(profili_spotname_cluster_2)
  # ggexport(profili_spotname_cluster_2, filename = paste("ggPlot spotname",spotname_new[i],
  #                                                       "'s DRC of cluster 2.png"))
}
##################

#Creazione tabelle di contingenza
##################

tab_conting_dis_bas <- matrix(nrow = length(spotname_new), ncol = 2+1)
row.names(tab_conting_dis_bas) <- spotname_new
colnames(tab_conting_dis_bas) <- c(1,2,"Totale")
```

```
for (i in 1:25) {
  tab_conting_dis_bas[i,] <- c(ncol(resistance_rawdata_spotname_cluster_1_tot[[i]]),
                               ncol(resistance_rawdata_spotname_cluster_2_tot[[i]]),
                               (ncol(resistance_rawdata_spotname_cluster_1_tot[[i]])+
                                  ncol(resistance_rawdata_spotname_cluster_2_tot[[i]])))
}

#creo tabelle con frequenze
tab_conting_dis_bas_intraspotname <- prop.table(tab_conting_dis_bas[,-3],margin = 1)

tab_conting_dis_bas <- as.data.frame(tab_conting_dis_bas)
kable(tab_conting_dis_bas, format='latex', caption = "Contingency tables for distance based clustering",
      label = "Contingency tables for distance based clustering")
tab_conting_dis_bas_intraspotname <- as.data.frame(tab_conting_dis_bas_intraspotname)
kable(tab_conting_dis_bas_intraspotname, format='latex',
      caption = "Contingency frequency tables for distance based clustering",
      label = "Contingency frequency tables for distance based clustering")

write.csv2(tab_conting_dis_bas, file="tabella contingenza distance based.csv",
           quote=F, na="", row.names=T, col.names=T)
write.csv2(tab_conting_dis_bas_intraspotname,
           file="tabella contingenza distance based frequenza intraspotname.csv",
           quote=F, na="", row.names=T, col.names=T)
##################


#DISTANCE BASED CLUSTERING - SECONDA CLUSTERIZZAZIONE

#DISTANCE-BASED CLUSTERING
######################
clustered_funs4_spotname_cluster_1_tot <- list()
clustered_funs4_spotname_cluster_2_tot <- list()

#cluster 1
for (i in c(1:10,12,13,15:17,19:22,24,25)) {
  i
  mod4_spotname_cluster_1 <-distance_ms(data = data_spotname_cluster_1_tot[[i]], num_cluster_seq = 2, met = "other")
  mod4_spotname_cluster_1$sil_opt
  cl_4_spotname_cluster_1 <- mod4_spotname_cluster_1$mod_opt_sil$clus

  clustered_funs4_spotname_cluster_1 <- get_clustered_funs_df(cl = cl_4_spotname_cluster_1,
                                                              data = data_spotname_cluster_1_tot[[i]],
                                                              method = "distance-based")
  centroids4_spotname_cluster_1 <- get_centroids_df(cl = cl_4_spotname_cluster_1,
                                                    data = data_spotname_cluster_1_tot[[i]],
                                                    method = "distance-based")
  clustered_funs4_spotname_cluster_1_tot[[i]] <- clustered_funs4_spotname_cluster_1

  dis_bas_spotname_cluster_1 <- ggplot(clustered_funs4_spotname_cluster_1) +
    geom_line(aes(time, Resistance, group = obs, col = cluster))  +
    ggtitle(paste("Distance-based clustering of spotname ",spotname_new[i],"\n","of cluster 1's DRC"))
  # print(dis_bas_spotname_cluster_1)
  dis_bas_cen_spotname_cluster_1 <- ggplot(centroids4_spotname_cluster_1) +
    geom_line(aes(time, Resistance, col = cluster)) +
    ggtitle(paste("Centroids of Distance-based clustering", "\n","of spotname ",spotname_new[i],"\n","of cluster 1's DRC"))
  # print(dis_bas_cen_spotname_cluster_1)

  full_distance_based_spotname_cluster_1 <- ggarrange(dis_bas_spotname_cluster_1,dis_bas_cen_spotname_cluster_1, ncol = 1, nrow = 2)
  ggexport(full_distance_based_spotname_cluster_1, filename = paste("Distance based clustering of Spotname",spotname_new[i],"of cluster 1 DRC.p
}

#cluster 2
for (i in c(1:9,11,12,14:19,21:23)) {
  i
  mod4_spotname_cluster_2 <-distance_ms(data = data_spotname_cluster_2_tot[[i]], num_cluster_seq = 2, met = "other")
  mod4_spotname_cluster_2$sil_opt
  cl_4_spotname_cluster_2 <- mod4_spotname_cluster_2$mod_opt_sil$clus

  clustered_funs4_spotname_cluster_2 <- get_clustered_funs_df(cl = cl_4_spotname_cluster_2,
                                                              data = data_spotname_cluster_2_tot[[i]],
                                                              method = "distance-based")
  centroids4_spotname_cluster_2 <- get_centroids_df(cl = cl_4_spotname_cluster_2,
                                                    data = data_spotname_cluster_2_tot[[i]],
                                                    method = "distance-based")
  clustered_funs4_spotname_cluster_2_tot[[i]] <- clustered_funs4_spotname_cluster_2

  dis_bas_spotname_cluster_2 <- ggplot(clustered_funs4_spotname_cluster_2) +
    geom_line(aes(time, Resistance, group = obs, col = cluster))  +
    ggtitle(paste("Distance-based clustering of spotname ",spotname_new[i],"\n","of cluster 2's DRC"))
  # print(dis_bas_spotname_cluster_2)
  dis_bas_cen_spotname_cluster_2 <- ggplot(centroids4_spotname_cluster_2) +
    geom_line(aes(time, Resistance, col = cluster)) +
    ggtitle(paste("Centroids of Distance-based clustering", "\n","of spotname ",spotname_new[i],"\n","of cluster 2's DRC"))
  # print(dis_bas_cen_spotname_cluster_2)

  full_distance_based_spotname_cluster_2 <- ggarrange(dis_bas_spotname_cluster_2,dis_bas_cen_spotname_cluster_2, ncol = 1, nrow = 2)
  ggexport(full_distance_based_spotname_cluster_2, filename = paste("Distance based clustering of Spotname",spotname_new[i],"of cluster 2 DRC.p
}
######################

#B-spline basis
####################
#cluster 1
clustered_funs5_hc_spotname_cluster_1_tot <- list()
```

```
clustered_funs5_km_spotname_cluster_1_tot <- list()
clustered_funs5_mb_spotname_cluster_1_tot <- list()

for (i in 1:25) {
  print(i)
  mod5_spotname_cluster_1 <- fil_bspline_ms_nclust(data = data_spotname_cluster_1_tot[[i]],
                                                   num_cluster_seq = 2:4, nbasis = 28)
  mod5_spotname_cluster_1$mod_opt$ind_hc$nbclust ## hierarchical
  max(mod5_spotname_cluster_1$mod_opt$ind_km$cluster) ## k-means
  mod5_spotname_cluster_1$mod_opt$mod_opt$G ## model-based

  cl_5_hc_spotname_cluster_1 <- mod5_spotname_cluster_1$mod_opt$ind_hc$cluster
  cl_5_km_spotname_cluster_1 <- mod5_spotname_cluster_1$mod_opt$ind_km$cluster
  cl_5_mb_spotname_cluster_1 <- mod5_spotname_cluster_1$mod_opt$mod_opt$classification

  clustered_funs5_hc_spotname_cluster_1 <- get_clustered_funs_df(cl = cl_5_hc_spotname_cluster_1,
                                                                 data = data_spotname_cluster_1_tot[[i]],
                                                                 method = "filtering B-spline\nhierarchical")
  clustered_funs5_km_spotname_cluster_1 <- get_clustered_funs_df(cl = cl_5_km_spotname_cluster_1,
                                                                 data = data_spotname_cluster_1_tot[[i]],
                                                                 method = "filtering B-spline\nk-means")
  clustered_funs5_mb_spotname_cluster_1 <- get_clustered_funs_df(cl = cl_5_mb_spotname_cluster_1,
                                                                 data = data_spotname_cluster_1_tot[[i]],
                                                                 method = "filtering B-spline\nmodel-based")
  centroids5_hc_spotname_cluster_1 <- get_centroids_df(cl_5_hc_spotname_cluster_1,
                                                       data_spotname_cluster_1_tot[[i]],
                                                       "filtering B-spline\nhierarchical")
  centroids5_km_spotname_cluster_1 <- get_centroids_df(cl_5_km_spotname_cluster_1,
                                                       data_spotname_cluster_1_tot[[i]],
                                                       "filtering B-spline\nk-means")
  centroids5_mb_spotname_cluster_1 <- get_centroids_df(cl_5_mb_spotname_cluster_1,
                                                       data_spotname_cluster_1_tot[[i]],
                                                       "filtering B-spline\nmodel-based")

  clustered_funs5_hc_spotname_cluster_1_tot[[i]] <- clustered_funs5_hc_spotname_cluster_1
  clustered_funs5_km_spotname_cluster_1_tot[[i]] <- clustered_funs5_km_spotname_cluster_1
  clustered_funs5_mb_spotname_cluster_1_tot[[i]] <- clustered_funs5_mb_spotname_cluster_1

  filt_bspl_hc_spotname_cluster_1 <- ggplot(clustered_funs5_hc_spotname_cluster_1) +
    geom_line(aes(time, Resistance, group = obs, col = cluster)) +
    ggtitle(paste("Filtering B-spline hierarchical clustering", "\n",
                  "of spotname",spotname_new[i],"of cluster 1's DRC"))
  # filt_bspl_hc_spotname_cluster_1
  filt_bspl_km_spotname_cluster_1 <- ggplot(clustered_funs5_km_spotname_cluster_1) +
    geom_line(aes(time, Resistance, group = obs, col = cluster)) +
    ggtitle(paste("Filtering B-spline k-means clustering", "\n",
                  "of spotname",spotname_new[i],"of cluster 1's DRC"))
  # filt_bspl_km_spotname_cluster_1
  filt_bspl_mb_spotname_cluster_1 <- ggplot(clustered_funs5_mb_spotname_cluster_1) +
    geom_line(aes(time, Resistance, group = obs, col = cluster)) +
    ggtitle(paste("Filtering B-spline model-based clustering", "\n",
                  "of spotname",spotname_new[i],"of cluster 1's DRC"))
  # filt_bspl_mb_spotname_cluster_1
  filt_bspl_hc_cen_spotname_cluster_1 <- ggplot(centroids5_hc_spotname_cluster_1) +
    geom_line(aes(time, Resistance, col = cluster)) +
    ggtitle(paste("Centroids of filtering B-spline hierarchical clustering", "\n",
                  "of spotname",spotname_new[i],"of cluster 1's DRC"))
  # filt_bspl_hc_cen_spotname_cluster_1
  filt_bspl_km_cen_spotname_cluster_1 <- ggplot(centroids5_km_spotname_cluster_1) +
    geom_line(aes(time, Resistance, col = cluster)) +
    ggtitle(paste("Centroids of filtering B-spline k-means clustering", "\n",
                  "of spotname",spotname_new[i],"of cluster 1's DRC"))
  # filt_bspl_km_cen_spotname_cluster_1
  filt_bspl_mb_cen_spotname_cluster_1 <- ggplot(centroids5_mb_spotname_cluster_1) +
    geom_line(aes(time, Resistance, col = cluster)) +
    ggtitle(paste("Centroids of filtering B-spline model-based clustering", "\n",
                  "of spotname",spotname_new[i],"of cluster 1's DRC"))
  # filt_bspl_mb_cen_spotname_cluster_1

  full_filt_bspl_new <- ggarrange(filt_bspl_hc_spotname_cluster_1,filt_bspl_hc_cen_spotname_cluster_1,
                                  filt_bspl_km_spotname_cluster_1,filt_bspl_km_cen_spotname_cluster_1,
                                  filt_bspl_mb_spotname_cluster_1,filt_bspl_mb_cen_spotname_cluster_1,
                                  ncol=1,nrow=2)
  ggexport(full_filt_bspl_new, filename = paste("Filtering B-spline profili-centroidi of spotname",
                                                spotname_new[i],"of cluster 1's DRC.png"))
  confr_filt_bspl_new <- ggarrange(filt_bspl_hc_spotname_cluster_1,filt_bspl_km_spotname_cluster_1,
                                   filt_bspl_mb_spotname_cluster_1,
                                   filt_bspl_hc_cen_spotname_cluster_1,filt_bspl_km_cen_spotname_cluster_1,
                                   filt_bspl_mb_cen_spotname_cluster_1,ncol=1,nrow=3)
  ggexport(confr_filt_bspl_new, filename = paste("Filtering B-spline technique's comparison of spotname",
                                                 spotname_new[i],"of cluster 1's DRC.png"))

}

#cluster 2
clustered_funs5_hc_spotname_cluster_2_tot <- list()
clustered_funs5_km_spotname_cluster_2_tot <- list()
clustered_funs5_mb_spotname_cluster_2_tot <- list()

for (i in 1:25) {
  print(i)
  mod5_spotname_cluster_2 <- fil_bspline_ms_nclust(data = data_spotname_cluster_2_tot[[i]],
                                                   num_cluster_seq = 2:4, nbasis = 28)
  mod5_spotname_cluster_2$mod_opt$ind_hc$nbclust ## hierarchical
  max(mod5_spotname_cluster_2$mod_opt$ind_km$cluster) ## k-means
```

```
  mod5_spotname_cluster_2$mod_opt$mod_opt$G ## model-based

  cl_5_hc_spotname_cluster_2 <- mod5_spotname_cluster_2$mod_opt$ind_hc$cluster
  cl_5_km_spotname_cluster_2 <- mod5_spotname_cluster_2$mod_opt$ind_km$cluster
  cl_5_mb_spotname_cluster_2 <- mod5_spotname_cluster_2$mod_opt$mod_opt$classification

  clustered_funs5_hc_spotname_cluster_2 <- get_clustered_funs_df(cl = cl_5_hc_spotname_cluster_2,
                                                                 data = data_spotname_cluster_2_tot[[i]],
                                                                 method = "filtering B-spline\nhierarchical")
  clustered_funs5_km_spotname_cluster_2 <- get_clustered_funs_df(cl = cl_5_km_spotname_cluster_2,
                                                                 data = data_spotname_cluster_2_tot[[i]],
                                                                 method = "filtering B-spline\nk-means")
  clustered_funs5_mb_spotname_cluster_2 <- get_clustered_funs_df(cl = cl_5_mb_spotname_cluster_2,
                                                                 data = data_spotname_cluster_2_tot[[i]],
                                                                 method = "filtering B-spline\nmodel-based")
  centroids5_hc_spotname_cluster_2 <- get_centroids_df(cl_5_hc_spotname_cluster_2,
                                                       data_spotname_cluster_2_tot[[i]],
                                                       "filtering B-spline\nhierarchical")
  centroids5_km_spotname_cluster_2 <- get_centroids_df(cl_5_km_spotname_cluster_2,
                                                       data_spotname_cluster_2_tot[[i]],
                                                       "filtering B-spline\nk-means")
  centroids5_mb_spotname_cluster_2 <- get_centroids_df(cl_5_mb_spotname_cluster_2,
                                                       data_spotname_cluster_2_tot[[i]],
                                                       "filtering B-spline\nmodel-based")

  clustered_funs5_hc_spotname_cluster_2_tot[[i]] <- clustered_funs5_hc_spotname_cluster_2
  clustered_funs5_km_spotname_cluster_2_tot[[i]] <- clustered_funs5_km_spotname_cluster_2
  clustered_funs5_mb_spotname_cluster_2_tot[[i]] <- clustered_funs5_mb_spotname_cluster_2

  filt_bspl_hc_spotname_cluster_2 <- ggplot(clustered_funs5_hc_spotname_cluster_2) +
    geom_line(aes(time, Resistance, group = obs, col = cluster)) +
    ggtitle(paste("Filtering B-spline hierarchical clustering", "\n", "of spotname",spotname_new[i],
                  "of cluster 2's DRC"))
  # filt_bspl_hc_spotname_cluster_2
  filt_bspl_km_spotname_cluster_2 <- ggplot(clustered_funs5_km_spotname_cluster_2) +
    geom_line(aes(time, Resistance, group = obs, col = cluster)) +
    ggtitle(paste("Filtering B-spline k-means clustering", "\n", "of spotname",spotname_new[i],
                  "of cluster 2's DRC"))
  # filt_bspl_km_spotname_cluster_2
  filt_bspl_mb_spotname_cluster_2 <- ggplot(clustered_funs5_mb_spotname_cluster_2) +
    geom_line(aes(time, Resistance, group = obs, col = cluster)) +
    ggtitle(paste("Filtering B-spline model-based clustering", "\n", "of spotname",spotname_new[i],
                  "of cluster 2's DRC"))
  # filt_bspl_mb_spotname_cluster_2
  filt_bspl_hc_cen_spotname_cluster_2 <- ggplot(centroids5_hc_spotname_cluster_2) +
    geom_line(aes(time, Resistance, col = cluster)) +
    ggtitle(paste("Centroids of filtering B-spline hierarchical clustering", "\n",
                  "of spotname",spotname_new[i],"of cluster 2's DRC"))
  # filt_bspl_hc_cen_spotname_cluster_2
  filt_bspl_km_cen_spotname_cluster_2 <- ggplot(centroids5_km_spotname_cluster_2) +
    geom_line(aes(time, Resistance, col = cluster)) +
    ggtitle(paste("Centroids of filtering B-spline k-means clustering", "\n",
                  "of spotname",spotname_new[i],"of cluster 2's DRC"))
  # filt_bspl_km_cen_spotname_cluster_2
  filt_bspl_mb_cen_spotname_cluster_2 <- ggplot(centroids5_mb_spotname_cluster_2) +
    geom_line(aes(time, Resistance, col = cluster)) +
    ggtitle(paste("Centroids of filtering B-spline model-based clustering", "\n",
                  "of spotname",spotname_new[i],"of cluster 2's DRC"))
  # filt_bspl_mb_cen_spotname_cluster_2

  full_filt_bspl_new <- ggarrange(filt_bspl_hc_spotname_cluster_2,filt_bspl_hc_cen_spotname_cluster_2,
                                  filt_bspl_km_spotname_cluster_2,filt_bspl_km_cen_spotname_cluster_2,
                                  filt_bspl_mb_spotname_cluster_2,filt_bspl_mb_cen_spotname_cluster_2,
                                  ncol=1,nrow=2)
  ggexport(full_filt_bspl_new, filename = paste("Filtering B-spline profili-centroidi of spotname",
                                                spotname_new[i],"of cluster 2's DRC.png"))
  confr_filt_bspl_new <- ggarrange(filt_bspl_hc_spotname_cluster_2,filt_bspl_km_spotname_cluster_2,
                                   filt_bspl_mb_spotname_cluster_2,
                                   filt_bspl_hc_cen_spotname_cluster_2,filt_bspl_km_cen_spotname_cluster_2,
                                   filt_bspl_mb_cen_spotname_cluster_2,ncol=1,nrow=3)
  ggexport(confr_filt_bspl_new, filename = paste("Filtering B-spline technique's comparison of spotname",
                                                 spotname_new[i],"of cluster 2's DRC.png"))
}
######################

#FPCA basis
######################
#cluster 1
clustered_funs6_hc_spotname_cluster_1_tot <- list()
clustered_funs6_km_spotname_cluster_1_tot <- list()
clustered_funs6_mb_spotname_cluster_1_tot <- list()

for (i in 1:25) {
  print(i)
  mod6_spotname_cluster_1 <- fil_fpca_ss_nbclust(data = data_spotname_cluster_1_tot[[i]], num_cluster_seq = 2:4, per_comp = 0.8)
  mod6_spotname_cluster_1$mod_opt$ind_hc$nbclust ## hierarchical
  max(mod6_spotname_cluster_1$mod_opt$ind_km$cluster) ## k-means
  mod6_spotname_cluster_1$mod_opt$mod_opt$G ## model-based

  cl_6_hc_spotname_cluster_1 <- mod6_spotname_cluster_1$mod_opt$ind_hc$cluster
  cl_6_km_spotname_cluster_1 <- mod6_spotname_cluster_1$mod_opt$ind_km$cluster
  cl_6_mb_spotname_cluster_1 <- mod6_spotname_cluster_1$mod_opt$mod_opt$classification

  clustered_funs6_hc_spotname_cluster_1 <- get_clustered_funs_df(cl = cl_6_hc_spotname_cluster_1,
                                                                 data = data_spotname_cluster_1_tot[[i]],
```

```
                                                          method = "filtering FPCA\nhierarchical")
  clustered_funs6_km_spotname_cluster_1 <- get_clustered_funs_df(cl = cl_6_km_spotname_cluster_1,
                                                          data = data_spotname_cluster_1_tot[[i]],
                                                          method = "filtering FPCA\nk-means")
  clustered_funs6_mb_spotname_cluster_1 <- get_clustered_funs_df(cl = cl_6_mb_spotname_cluster_1,
                                                          data = data_spotname_cluster_1_tot[[i]],
                                                          method = "filtering FPCA\nmodel-based")
  centroids6_hc_spotname_cluster_1 <- get_centroids_df(cl_6_hc_spotname_cluster_1, data_spotname_cluster_1_tot[[i]], "filtering FPCA\nhierarch
  centroids6_km_spotname_cluster_1 <- get_centroids_df(cl_6_km_spotname_cluster_1, data_spotname_cluster_1_tot[[i]], "filtering FPCA\nk-means"
  centroids6_mb_spotname_cluster_1 <- get_centroids_df(cl_6_mb_spotname_cluster_1, data_spotname_cluster_1_tot[[i]], "filtering FPCA\nmodel-bas

  clustered_funs6_hc_spotname_cluster_1_tot[[i]] <- clustered_funs6_hc_spotname_cluster_1
  clustered_funs6_km_spotname_cluster_1_tot[[i]] <- clustered_funs6_km_spotname_cluster_1
  clustered_funs6_mb_spotname_cluster_1_tot[[i]] <- clustered_funs6_mb_spotname_cluster_1

  filt_fpca_hc_spotname_cluster_1 <- ggplot(clustered_funs6_hc_spotname_cluster_1) +
    geom_line(aes(time, Resistance, group = obs, col = cluster)) +
    ggtitle(paste("Filtering FPCA hierarchical clustering", "\n", "of spotname",spotname_new[i],"of cluster 1's DRC"))
  # filt_fpca_hc_spotname_cluster_1
  filt_fpca_km_spotname_cluster_1 <- ggplot(clustered_funs6_km_spotname_cluster_1) +
    geom_line(aes(time, Resistance, group = obs, col = cluster)) +
    ggtitle(paste("Filtering FPCA k-means clustering", "\n", "of spotname",spotname_new[i],"of cluster 1's DRC"))
  # filt_fpca_km_spotname_cluster_1
  filt_fpca_mb_spotname_cluster_1 <- ggplot(clustered_funs6_mb_spotname_cluster_1) +
    geom_line(aes(time, Resistance, group = obs, col = cluster)) +
    ggtitle(paste("Filtering FPCA model-based clustering", "\n", "of spotname",spotname_new[i],"of cluster 1's DRC"))
  # filt_fpca_mb_spotname_cluster_1
  filt_fpca_hc_cen_spotname_cluster_1 <- ggplot(centroids6_hc_spotname_cluster_1) +
    geom_line(aes(time, Resistance, col = cluster)) +
    ggtitle(paste("Centroids of filtering FPCA hierarchical clustering", "\n", "of spotname",spotname_new[i],"of cluster 1's DRC"))
  # filt_fpca_hc_cen_spotname_cluster_1
  filt_fpca_km_cen_spotname_cluster_1 <- ggplot(centroids6_km_spotname_cluster_1) +
    geom_line(aes(time, Resistance, col = cluster)) +
    ggtitle(paste("Centroids of filtering FPCA k-means clustering", "\n", "of spotname",spotname_new[i],"of cluster 1's DRC"))
  # filt_fpca_km_cen_spotname_cluster_1
  filt_fpca_mb_cen_spotname_cluster_1 <- ggplot(centroids6_mb_spotname_cluster_1) +
    geom_line(aes(time, Resistance, col = cluster)) +
    ggtitle(paste("Centroids of filtering FPCA model-based clustering", "\n", "of spotname",spotname_new[i],"of cluster 1's DRC"))
  # filt_fpca_mb_cen_spotname_cluster_1

  full_filt_fpca_new <- ggarrange(filt_fpca_hc_spotname_cluster_1,filt_fpca_hc_cen_spotname_cluster_1,filt_fpca_km_spotname_cluster_1,filt_fpca
                                  filt_fpca_mb_spotname_cluster_1,filt_fpca_mb_cen_spotname_cluster_1,ncol=1,nrow=2)
  ggexport(full_filt_fpca_new, filename = paste("Filtering FPCA profili-centroidi of spotname",spotname_new[i],"of cluster 1's DRC.png"))
  confr_filt_fpca_new <- ggarrange(filt_fpca_hc_spotname_cluster_1,filt_fpca_km_spotname_cluster_1,filt_fpca_mb_spotname_cluster_1,
                                   filt_fpca_hc_cen_spotname_cluster_1,filt_fpca_km_cen_spotname_cluster_1,filt_fpca_mb_cen_spotname_cluster_1,
  ggexport(confr_filt_fpca_new, filename = paste("Filtering FPCA technique's comparison of spotname",spotname_new[i],"of cluster 1's DRC.png"))
}

#cluster 2
clustered_funs6_hc_spotname_cluster_2_tot <- list()
clustered_funs6_km_spotname_cluster_2_tot <- list()
clustered_funs6_mb_spotname_cluster_2_tot <- list()
for (i in 1:25) {
  print(i)
  mod6_spotname_cluster_2 <- fil_fpca_ss_nbclust(data = data_spotname_cluster_2_tot[[i]],
                                                 num_cluster_seq = 2:4, per_comp = 0.8)
  mod6_spotname_cluster_2$mod_opt$ind_hc$nbclust ## hierarchical
  max(mod6_spotname_cluster_2$mod_opt$ind_km$cluster) ## k-means
  mod6_spotname_cluster_2$mod_opt$mod_opt$G ## model-based

  cl_6_hc_spotname_cluster_2 <- mod6_spotname_cluster_2$mod_opt$ind_hc$cluster
  cl_6_km_spotname_cluster_2 <- mod6_spotname_cluster_2$mod_opt$ind_km$cluster
  cl_6_mb_spotname_cluster_2 <- mod6_spotname_cluster_2$mod_opt$mod_opt$classification

  clustered_funs6_hc_spotname_cluster_2 <- get_clustered_funs_df(cl = cl_6_hc_spotname_cluster_2,
                                                          data = data_spotname_cluster_2_tot[[i]],
                                                          method = "filtering FPCA\nhierarchical")
  clustered_funs6_km_spotname_cluster_2 <- get_clustered_funs_df(cl = cl_6_km_spotname_cluster_2,
                                                          data = data_spotname_cluster_2_tot[[i]],
                                                          method = "filtering FPCA\nk-means")
  clustered_funs6_mb_spotname_cluster_2 <- get_clustered_funs_df(cl = cl_6_mb_spotname_cluster_2,
                                                          data = data_spotname_cluster_2_tot[[i]],
                                                          method = "filtering FPCA\nmodel-based")
  centroids6_hc_spotname_cluster_2 <- get_centroids_df(cl_6_hc_spotname_cluster_2,
                                                       data_spotname_cluster_2_tot[[i]],
                                                       "filtering FPCA\nhierarchical")
  centroids6_km_spotname_cluster_2 <- get_centroids_df(cl_6_km_spotname_cluster_2,
                                                       data_spotname_cluster_2_tot[[i]],
                                                       "filtering FPCA\nk-means")
  centroids6_mb_spotname_cluster_2 <- get_centroids_df(cl_6_mb_spotname_cluster_2,
                                                       data_spotname_cluster_2_tot[[i]],
                                                       "filtering FPCA\nmodel-based")

  clustered_funs6_hc_spotname_cluster_2_tot[[i]] <- clustered_funs6_hc_spotname_cluster_2
  clustered_funs6_km_spotname_cluster_2_tot[[i]] <- clustered_funs6_km_spotname_cluster_2
  clustered_funs6_mb_spotname_cluster_2_tot[[i]] <- clustered_funs6_mb_spotname_cluster_2

  filt_fpca_hc_spotname_cluster_2 <- ggplot(clustered_funs6_hc_spotname_cluster_2) +
    geom_line(aes(time, Resistance, group = obs, col = cluster)) +
    ggtitle(paste("Filtering FPCA hierarchical clustering", "\n", "of spotname",spotname_new[i],
                  "of cluster 2's DRC"))
  # filt_fpca_hc_spotname_cluster_2
  filt_fpca_km_spotname_cluster_2 <- ggplot(clustered_funs6_km_spotname_cluster_2) +
    geom_line(aes(time, Resistance, group = obs, col = cluster)) +
    ggtitle(paste("Filtering FPCA k-means clustering", "\n", "of spotname",spotname_new[i],
```

```
                "of cluster 2's DRC"))
  # filt_fpca_km_spotname_cluster_2
  filt_fpca_mb_spotname_cluster_2 <- ggplot(clustered_funs6_mb_spotname_cluster_2) +
    geom_line(aes(time, Resistance, group = obs, col = cluster)) +
    ggtitle(paste("Filtering FPCA model-based clustering", "\n", "of spotname",spotname_new[i],
                  "of cluster 2's DRC"))
  # filt_fpca_mb_spotname_cluster_2
  filt_fpca_hc_cen_spotname_cluster_2 <- ggplot(centroids6_hc_spotname_cluster_2) +
    geom_line(aes(time, Resistance, col = cluster)) +
    ggtitle(paste("Centroids of filtering FPCA hierarchical clustering", "\n",
                  "of spotname",spotname_new[i],"of cluster 2's DRC"))
  # filt_fpca_hc_cen_spotname_cluster_2
  filt_fpca_km_cen_spotname_cluster_2 <- ggplot(centroids6_km_spotname_cluster_2) +
    geom_line(aes(time, Resistance, col = cluster)) +
    ggtitle(paste("Centroids of filtering FPCA k-means clustering", "\n",
                  "of spotname",spotname_new[i],"of cluster 2's DRC"))
  # filt_fpca_km_cen_spotname_cluster_2
  filt_fpca_mb_cen_spotname_cluster_2 <- ggplot(centroids6_mb_spotname_cluster_2) +
    geom_line(aes(time, Resistance, col = cluster)) +
    ggtitle(paste("Centroids of filtering FPCA model-based clustering", "\n",
                  "of spotname",spotname_new[i],"of cluster 2's DRC"))
  # filt_fpca_mb_cen_spotname_cluster_2

  full_filt_fpca_new <- ggarrange(filt_fpca_hc_spotname_cluster_2,filt_fpca_hc_cen_spotname_cluster_2,
                                  filt_fpca_km_spotname_cluster_2,filt_fpca_km_cen_spotname_cluster_2,
                                  filt_fpca_mb_spotname_cluster_2,filt_fpca_mb_cen_spotname_cluster_2,
                                  ncol=1,nrow=2)
  ggexport(full_filt_fpca_new, filename = paste("Filtering FPCA profili-centroidi of spotname",
                                                spotname_new[i],"of cluster 2's DRC.png"))
  confr_filt_fpca_new <- ggarrange(filt_fpca_hc_spotname_cluster_2,filt_fpca_km_spotname_cluster_2,
                                   filt_fpca_mb_spotname_cluster_2,
                                   filt_fpca_hc_cen_spotname_cluster_2,filt_fpca_km_cen_spotname_cluster_2,
                                   filt_fpca_mb_cen_spotname_cluster_2,ncol=1,nrow=3)
  ggexport(confr_filt_fpca_new, filename = paste("Filtering FPCA technique's comparison of spotname",
                                                 spotname_new[i],"of cluster 2's DRC.png"))
}
#####################




#APPROCCIO GERARCHICO
#B-spline basis - Prima clusterizzazione
#clustering
#################
clustered_funs5_hc_spotname_new_tot <- list()

for (i in 1:length(spotname_new)) {
  mod5_spotname_new <- fil_bspline_ms_nclust(data = data_spotname_new_tot[[i]],
                                             num_cluster_seq = 2:4, nbasis = 28)
  mod5_spotname_new$mod_opt$ind_hc$nbclust ## hierarchical

  cl_5_hc_spotname_new <- mod5_spotname_new$mod_opt$ind_hc$cluster

  clustered_funs5_hc_spotname_new <- get_clustered_funs_df(cl = cl_5_hc_spotname_new,
                                                           data = data_spotname_new_tot[[i]],
                                                           method = "filtering B-spline\nhierarchical")
  centroids5_hc_spotname_new <- get_centroids_df(cl_5_hc_spotname_new, data_spotname_new_tot[[i]],
                                                 "filtering B-spline\nhierarchical")

  clustered_funs5_hc_spotname_new_tot[[i]] <- clustered_funs5_hc_spotname_new

  filt_bspl_hc_spotname_new <- ggplot(clustered_funs5_hc_spotname_new) +
    geom_line(aes(time, Resistance, group = obs, col = cluster)) +
    ggtitle(paste("Filtering B-spline hierarchical clustering", "\n", "of spotname",spotname_new[i],"'s DRC"))
  # filt_bspl_hc_spotname_new
  filt_bspl_hc_cen_spotname_new <- ggplot(centroids5_hc_spotname_new) +
    geom_line(aes(time, Resistance, col = cluster)) +
    ggtitle(paste("Centroids of filtering B-spline hierarchical clustering", "\n", "of spotname",
                  spotname_new[i],"'s DRC"))
  # filt_bspl_hc_cen_spotname_new

  full_filt_bspl_new <- ggarrange(filt_bspl_hc_spotname_new,filt_bspl_hc_cen_spotname_new,ncol=1,nrow=2)
  ggexport(full_filt_bspl__new, filename = paste("Filtering B-spline profili-centroidi of spotname",
                                                 spotname_new[i],"'s DRC.png"))
}
#################

#suddivido i cluster di bspl
#################
scalari_cluster_1_bspl_hc_tot <- list()
scalari_cluster_2_bspl_hc_tot <- list()
scalari_cluster_3_bspl_hc_tot <- list()
scalari_cluster_4_bspl_hc_tot <- list()

for (i in 1:24) {

  profili_cluster_bspl_hc <- clustered_funs5_hc_spotname_new_tot[[i]][seq(
    from = 1, to = ncol(resistance_rawdata_spotname_new_tot[[i]])),]
  profili_cluster_1_bspl_hc <- profili_cluster_bspl_hc[profili_cluster_bspl_hc$cluster == 1,]
  profili_cluster_2_bspl_hc <- profili_cluster_bspl_hc[profili_cluster_bspl_hc$cluster == 2,]
  profili_cluster_3_bspl_hc <- profili_cluster_bspl_hc[profili_cluster_bspl_hc$cluster == 3,]
  profili_cluster_4_bspl_hc <- profili_cluster_bspl_hc[profili_cluster_bspl_hc$cluster == 4,]

  #cambio nome a colonna delle obs e di cluster
```

```
  colnames(profili_cluster_1_bspl_hc)[2] <- "dateTime"
  colnames(profili_cluster_1_bspl_hc)[4] <- "cluster bspl hc"

  colnames(profili_cluster_2_bspl_hc)[2] <- "dateTime"
  colnames(profili_cluster_2_bspl_hc)[4] <- "cluster bspl hc"

  colnames(profili_cluster_3_bspl_hc)[2] <- "dateTime"
  colnames(profili_cluster_3_bspl_hc)[4] <- "cluster bspl hc"

  colnames(profili_cluster_4_bspl_hc)[2] <- "dateTime"
  colnames(profili_cluster_4_bspl_hc)[4] <- "cluster bspl hc"

  #unisco colonna relativa al cluster di appartenenza
  scalari_cluster_1_bspl_hc <- merge(scalari_associazione_new,profili_cluster_1_bspl_hc,by = ("dateTime"))
  scalari_cluster_2_bspl_hc <- merge(scalari_associazione_new,profili_cluster_2_bspl_hc,by = ("dateTime"))
  scalari_cluster_3_bspl_hc <- merge(scalari_associazione_new,profili_cluster_3_bspl_hc,by = ("dateTime"))
  scalari_cluster_4_bspl_hc <- merge(scalari_associazione_new,profili_cluster_4_bspl_hc,by = ("dateTime"))

  scalari_cluster_1_bspl_hc_tot[[i]] <- scalari_cluster_1_bspl_hc
  scalari_cluster_2_bspl_hc_tot[[i]] <- scalari_cluster_2_bspl_hc
  scalari_cluster_3_bspl_hc_tot[[i]] <- scalari_cluster_3_bspl_hc
  scalari_cluster_4_bspl_hc_tot[[i]] <- scalari_cluster_4_bspl_hc
}
#################

#creo matrici resistance rawdata
#################
resistance_rawdata_spotname_bspl_hc_cluster_1_tot <- list()
resistance_rawdata_spotname_bspl_hc_cluster_2_tot <- list()
resistance_rawdata_spotname_bspl_hc_cluster_3_tot <- list()
resistance_rawdata_spotname_bspl_hc_cluster_4_tot <- list()

for (i in 1:24) {

  resistance_rawdata_spotname_bspl_hc_cluster_1 <- data_profili_new %>%
    filter(id %in% scalari_cluster_1_bspl_hc_tot[[i]]$dateTime ) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_bspl_hc_cluster_1<- t(resistance_rawdata_spotname_bspl_hc_cluster_1)

  colnames(resistance_rawdata_spotname_bspl_hc_cluster_1)<- resistance_rawdata_spotname_bspl_hc_cluster_1[1,]
  resistance_rawdata_spotname_bspl_hc_cluster_1<- resistance_rawdata_spotname_bspl_hc_cluster_1[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_bspl_hc_cluster_1[is.na(resistance_rawdata_spotname_bspl_hc_cluster_1)] <- 0

  storage.mode(resistance_rawdata_spotname_bspl_hc_cluster_1) <- "numeric"       #la matrice veniva salvata come character, così diventa numer
  resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[i]] <- resistance_rawdata_spotname_bspl_hc_cluster_1
  names(resistance_rawdata_spotname_bspl_hc_cluster_1_tot)[[i]] <- spotname_new[i]

  resistance_rawdata_spotname_bspl_hc_cluster_2 <- data_profili_new %>%
    filter(id %in% scalari_cluster_2_bspl_hc_tot[[i]]$dateTime) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_bspl_hc_cluster_2<- t(resistance_rawdata_spotname_bspl_hc_cluster_2)

  colnames(resistance_rawdata_spotname_bspl_hc_cluster_2)<- resistance_rawdata_spotname_bspl_hc_cluster_2[1,]
  resistance_rawdata_spotname_bspl_hc_cluster_2<- resistance_rawdata_spotname_bspl_hc_cluster_2[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_bspl_hc_cluster_2[is.na(resistance_rawdata_spotname_bspl_hc_cluster_2)] <- 0

  storage.mode(resistance_rawdata_spotname_bspl_hc_cluster_2) <- "numeric"       #la matrice veniva salvata come character, così diventa numer
  resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[i]] <- resistance_rawdata_spotname_bspl_hc_cluster_2
  names(resistance_rawdata_spotname_bspl_hc_cluster_2_tot)[[i]] <- spotname_new[i]

  resistance_rawdata_spotname_bspl_hc_cluster_3 <- data_profili_new %>%
    filter(id %in% scalari_cluster_3_bspl_hc_tot[[i]]$dateTime ) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_bspl_hc_cluster_3<- t(resistance_rawdata_spotname_bspl_hc_cluster_3)

  colnames(resistance_rawdata_spotname_bspl_hc_cluster_3)<- resistance_rawdata_spotname_bspl_hc_cluster_3[1,]
  resistance_rawdata_spotname_bspl_hc_cluster_3<- resistance_rawdata_spotname_bspl_hc_cluster_3[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_bspl_hc_cluster_3[is.na(resistance_rawdata_spotname_bspl_hc_cluster_3)] <- 0

  storage.mode(resistance_rawdata_spotname_bspl_hc_cluster_3) <- "numeric"       #la matrice veniva salvata come character, così diventa numer
  resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[i]] <- resistance_rawdata_spotname_bspl_hc_cluster_3
  names(resistance_rawdata_spotname_bspl_hc_cluster_3_tot)[[i]] <- spotname_new[i]
```

```
  resistance_rawdata_spotname_bspl_hc_cluster_4 <- data_profili_new %>%
    filter(id %in% scalari_cluster_4_bspl_hc_tot[[i]]$dateTime) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_bspl_hc_cluster_4<- t(resistance_rawdata_spotname_bspl_hc_cluster_4)

  colnames(resistance_rawdata_spotname_bspl_hc_cluster_4)<- resistance_rawdata_spotname_bspl_hc_cluster_4[1,]
  resistance_rawdata_spotname_bspl_hc_cluster_4<- resistance_rawdata_spotname_bspl_hc_cluster_4[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_bspl_hc_cluster_4[is.na(resistance_rawdata_spotname_bspl_hc_cluster_4)] <- 0

  storage.mode(resistance_rawdata_spotname_bspl_hc_cluster_4) <- "numeric"       #la matrice veniva salvata come character, così diventa numeri
  resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[i]] <- resistance_rawdata_spotname_bspl_hc_cluster_4
  names(resistance_rawdata_spotname_bspl_hc_cluster_4_tot)[[i]] <- spotname_new[i]
}
#################

#taglio durata profili come all'inizio (240,280,300,320,340 ms)
####################
resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[1]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[1]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[2]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[2]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[3]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[3]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[4]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[4]][1:280,]
resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[5]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[5]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[6]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[6]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[7]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[7]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[8]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[8]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[9]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[9]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[10]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[10]][1:220,]
resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[11]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[11]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[12]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[12]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[13]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[13]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[14]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[14]][1:340,]
resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[15]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[15]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[16]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[16]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[17]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[17]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[18]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[18]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[19]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[19]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[20]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[20]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[21]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[21]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[22]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[22]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[23]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[23]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[24]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[24]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[25]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[25]][1:320,]

resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[1]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[1]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[2]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[2]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[3]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[3]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[4]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[4]][1:280,]
resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[5]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[5]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[6]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[6]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[7]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[7]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[8]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[8]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[9]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[9]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[10]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[10]][1:220,]
```

```
resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[11]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[11]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[12]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[12]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[13]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[13]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[14]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[14]][1:340,]
resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[15]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[15]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[16]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[16]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[17]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[17]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[18]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[18]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[19]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[19]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[20]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[20]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[21]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[21]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[22]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[22]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[23]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[23]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[24]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[24]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[25]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[25]][1:320,]

resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[1]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[1]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[2]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[2]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[3]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[3]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[4]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[4]][1:280,]
resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[5]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[5]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[6]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[6]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[7]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[7]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[8]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[8]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[9]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[9]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[10]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[10]][1:220,]
resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[11]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[11]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[12]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[12]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[13]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[13]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[14]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[14]][1:340,]
resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[15]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[15]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[16]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[16]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[17]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[17]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[18]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[18]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[19]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[19]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[20]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[20]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[21]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[21]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[22]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[22]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[23]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[23]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[24]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[24]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[25]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[25]][1:320,]

resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[1]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[1]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[2]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[2]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[3]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[3]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[4]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[4]][1:280,]
resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[5]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[5]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[6]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[6]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[7]]<-
```

```
  resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[7]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[8]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[8]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[9]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[9]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[10]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[10]][1:220,]
resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[11]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[11]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[12]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[12]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[13]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[13]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[14]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[14]][1:340,]
resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[15]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[15]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[16]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[16]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[17]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[17]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[18]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[18]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[19]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[19]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[20]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[20]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[21]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[21]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[22]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[22]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[23]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[23]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[24]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[24]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[25]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[25]][1:320,]
####################

#Creazione tabelle di contingenza
#################
tab_conting_filt_bspl <- matrix(nrow = length(spotname_new), ncol = 4+1)
row.names(tab_conting_filt_bspl)<- spotname_new
colnames(tab_conting_filt_bspl)<- c(1,2,3,4,"Totale")

#visto che filt bspl dà errore con spotname 53703 (il 25esimo), levo ultima riga
tab_conting_filt_bspl <- tab_conting_filt_bspl[-25,]

#riempio tabella
for (i in 1:24) {
  tab_conting_filt_bspl[i,] <- c(ncol(resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[i]]),
                                 ncol(resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[i]]),
                                 ncol(resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[i]]),
                                 ncol(resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[i]]),
                                 (ncol(resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[i]])+
                                    ncol(resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[i]])+
                                    ncol(resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[i]])+
                                    ncol(resistance_rawdata_spotname_bspl_hc_cluster_4_tot[[i]])))
}

tab_conting_filt_bspl_freq_intraspotname <- prop.table(tab_conting_filt_bspl[,-5],margin = 1)

tab_conting_filt_bspl <- as.data.frame(tab_conting_filt_bspl)
kable(tab_conting_filt_bspl, format='latex', caption = "Contingency tables for 220ms' group",
      label = "Tabella contingenza 220.png")
#creo tabelle con frequenze

write.csv2(tab_conting_filt_bspl, file="tabella contingenza filtering bspl.csv",
           quote=F, na="", row.names=T, col.names=T)
write.csv2(tab_conting_filt_bspl_freq_intraspotname,
           file="tabella contingenza filtering bspl frequenza intraspotname.csv",
           quote=F, na="", row.names=T, col.names=T)
#################



#Filtering B-spline - Seconda clusterizzazione (uso solo i primi 3 cluster, il 4 contiene pochissimi profili)
#clusterizzo i singoli cluster (1-2-3) di filtering b-spline hierarchical

#Cluster 1

#clustering
#################
clustered_funs5_hc_spotname_new_cluster_1_tot <- list()
centroids5_hc_spotname_new_cluster_1_tot <- list()

#24 errore, 25 errore da prima clusterizzazione
for (i in 1:23) {
  if (i == 11 | 19) {
    breaks <- seq(0,0.5, by = 0.1);
    limits <- c(0,0.5);
  }
  else {
    breaks <- seq(0,0.3, by = 0.1);
    limits <- c(0,0.35);
```

```
  }

  if (length(levels(clustered_funs5_hc_spotname_new_cluster_1_tot[[i]][["cluster"]]))==2) {
    labels <- c("A","B");
  }
  else if (length(levels(clustered_funs5_hc_spotname_new_cluster_1_tot[[1]][["cluster"]]))==3) {
    labels <- c("A","B","C");
  }
  else {
    labels <- c("A","B","C","D");
  }

  mod5_spotname_new_cluster_1 <- fil_bspline_ms_nclust(data = data_spotname_bspl_cluster_1_tot[[i]],
                                                       num_cluster_seq = 2:4, nbasis = 28)
  mod5_spotname_new_cluster_1$mod_opt$ind_hc$nbclust ## hierarchical

  cl_5_hc_spotname_new_cluster_1 <- mod5_spotname_new_cluster_1$mod_opt$ind_hc$cluster

  clustered_funs5_hc_spotname_new_cluster_1 <- get_clustered_funs_df(cl = cl_5_hc_spotname_new_cluster_1,
                                                                     data = data_spotname_bspl_cluster_1_tot[[i]],
                                                                     method = "filtering B-spline\nhierarchical")
  centroids5_hc_spotname_new_cluster_1 <- get_centroids_df(cl_5_hc_spotname_new_cluster_1,
                                                           data_spotname_bspl_cluster_1_tot[[i]],
                                                           "filtering B-spline\nhierarchical")

  clustered_funs5_hc_spotname_new_cluster_1_tot[[i]] <- clustered_funs5_hc_spotname_new_cluster_1

  filt_bspl_hc_spotname_new_cluster_1 <- ggplot(clustered_funs5_hc_spotname_new_cluster_1) +
    geom_line(aes(time, Resistance, group = obs, col = cluster)) +
    scale_y_continuous(breaks = breaks, limits = limits) +
    scale_fill_discrete(labels <- labels) +
    ggtitle(paste("Filtering B-spline hierarchical clustering", "\n", "of bspl cluster 1 of spotname",
                  spotname_new[i],"'s DRC"))
  # filt_bspl_hc_spotname_new_cluster_1
  filt_bspl_hc_cen_spotname_new_cluster_1 <- ggplot(centroids5_hc_spotname_new_cluster_1) +
    geom_line(aes(time, Resistance, col = cluster)) +
    scale_y_continuous(breaks = breaks, limits = limits) +
    scale_fill_discrete(labels <- labels) +
    ggtitle(paste("Centroids of filtering B-spline hierarchical clustering", "\n",
                  "of bspl cluster 1 of spotname",spotname_new[i],"'s DRC"))
  # filt_bspl_hc_cen_spotname_new_cluster_1

  full_filt_bspl__new <- ggarrange(filt_bspl_hc_spotname_new_cluster_1,filt_bspl_hc_cen_spotname_new_cluster_1,
                                   ncol=1,nrow=2)
  ggexport(full_filt_bspl__new,
           filename = paste("Filtering B-spline profili-centroidi of bspl cluster 1 of spotname",
                            spotname_new[i],"'s DRC.png"))
}
##################
```

```
#suddivido i sotto cluster in oggetti diversi
##################
scalari_cluster_1_cluster_A_bspl_hc_tot <- list()
scalari_cluster_1_cluster_B_bspl_hc_tot <- list()
scalari_cluster_1_cluster_C_bspl_hc_tot <- list()
scalari_cluster_1_cluster_D_bspl_hc_tot <- list()

for (i in 1:23) {

  profili_cluster_1_bspl_hc <- clustered_funs5_hc_spotname_new_cluster_1_tot[[i]][seq(
    from = 1, to = ncol(resistance_rawdata_spotname_bspl_hc_cluster_1_tot[[i]])),]
  profili_cluster_1_cluster_A_bspl_hc <- profili_cluster_1_bspl_hc[profili_cluster_1_bspl_hc$cluster == 1,]
  profili_cluster_1_cluster_B_bspl_hc <- profili_cluster_1_bspl_hc[profili_cluster_1_bspl_hc$cluster == 2,]
  profili_cluster_1_cluster_C_bspl_hc <- profili_cluster_1_bspl_hc[profili_cluster_1_bspl_hc$cluster == 3,]
  profili_cluster_1_cluster_D_bspl_hc <- profili_cluster_1_bspl_hc[profili_cluster_1_bspl_hc$cluster == 4,]

  #cambio nome a colonna delle obs e di cluster
  colnames(profili_cluster_1_cluster_A_bspl_hc)[2] <- "dateTime"
  colnames(profili_cluster_1_cluster_A_bspl_hc)[4] <- "cluster bspl hc A"

  colnames(profili_cluster_1_cluster_B_bspl_hc)[2] <- "dateTime"
  colnames(profili_cluster_1_cluster_B_bspl_hc)[4] <- "cluster bspl hc B"

  colnames(profili_cluster_1_cluster_C_bspl_hc)[2] <- "dateTime"
  colnames(profili_cluster_1_cluster_C_bspl_hc)[4] <- "cluster bspl hc C"

  colnames(profili_cluster_1_cluster_D_bspl_hc)[2] <- "dateTime"
  colnames(profili_cluster_1_cluster_D_bspl_hc)[4] <- "cluster bspl hc D"

  #unisco colonna relativa al cluster di appartenenza
  scalari_cluster_1_cluster_A_bspl_hc <- merge(scalari_associazione_new,
                                               profili_cluster_1_cluster_A_bspl_hc,by = ("dateTime"))
  scalari_cluster_1_cluster_B_bspl_hc <- merge(scalari_associazione_new,
                                               profili_cluster_1_cluster_B_bspl_hc,by = ("dateTime"))
  scalari_cluster_1_cluster_C_bspl_hc <- merge(scalari_associazione_new,
                                               profili_cluster_1_cluster_C_bspl_hc,by = ("dateTime"))
  scalari_cluster_1_cluster_D_bspl_hc <- merge(scalari_associazione_new,
                                               profili_cluster_1_cluster_D_bspl_hc,by = ("dateTime"))

  scalari_cluster_1_cluster_A_bspl_hc_tot[[i]] <- scalari_cluster_1_cluster_A_bspl_hc
  scalari_cluster_1_cluster_B_bspl_hc_tot[[i]] <- scalari_cluster_1_cluster_B_bspl_hc
  scalari_cluster_1_cluster_C_bspl_hc_tot[[i]] <- scalari_cluster_1_cluster_C_bspl_hc
  scalari_cluster_1_cluster_D_bspl_hc_tot[[i]] <- scalari_cluster_1_cluster_D_bspl_hc

}
```

```
#################

#creo matrici resistance rawdata
#################
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot <- list()
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot <- list()
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot <- list()
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot <- list()

for (i in 1:23) {

  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A <- data_profili_new %>%
    filter(id %in% scalari_cluster_1_cluster_A_bspl_hc_tot[[i]]$dateTime ) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A<-
    t(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A)

  colnames(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A)<-
    resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A[1,]
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A<-
    resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A[
    is.na(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A)] <- 0

  storage.mode(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A) <- "numeric"
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[i]] <-
    resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A
  names(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot)[[i]] <- spotname_new[i]


  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B <- data_profili_new %>%
    filter(id %in% scalari_cluster_1_cluster_B_bspl_hc_tot[[i]]$dateTime) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B<-
    t(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B)

  colnames(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B)<-
    resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B[1,]
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B<-
    resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B[
    is.na(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B)] <- 0

  storage.mode(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B) <- "numeric"
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[i]] <-
    resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B
  names(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot)[[i]] <- spotname_new[i]

  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C <- data_profili_new %>%
    filter(id %in% scalari_cluster_1_cluster_C_bspl_hc_tot[[i]]$dateTime ) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C<-
    t(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C)

  colnames(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C)<-
    resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C[1,]
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C<-
    resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C[
    is.na(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C)] <- 0

  storage.mode(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C) <- "numeric"
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[i]] <-
    resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C
  names(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot)[[i]] <- spotname_new[i]


  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D <- data_profili_new %>%
    filter(id %in% scalari_cluster_1_cluster_D_bspl_hc_tot[[i]]$dateTime) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)
```

```
  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D<-
    t(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D)

  colnames(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D)<-
    resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D[1,]
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D<-
    resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D[
    is.na(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D)] <- 0

  storage.mode(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D) <- "numeric"
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[i]] <-
    resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D
  names(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot)[[i]] <- spotname_new[i]

}
#################

#taglio durata profili come all'inizio (240,280,300,320,340 ms)
####################
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[1]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[1]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[2]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[2]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[3]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[3]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[4]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[4]][1:280,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[5]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[5]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[6]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[6]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[7]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[7]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[8]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[8]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[9]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[9]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[10]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[10]][1:220,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[11]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[11]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[12]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[12]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[13]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[13]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[14]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[14]][1:340,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[15]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[15]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[16]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[16]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[17]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[17]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[18]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[18]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[19]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[19]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[20]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[20]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[21]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[21]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[22]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[22]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[23]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[23]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[24]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[24]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[25]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[25]][1:320,]

resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[1]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[1]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[2]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[2]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[3]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[3]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[4]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[4]][1:280,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[5]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[5]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[6]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[6]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[7]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[7]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[8]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[8]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[9]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[9]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[10]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[10]][1:220,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[11]]<-
```

```
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[11]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[12]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[12]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[13]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[13]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[14]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[14]][1:340,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[15]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[15]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[16]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[16]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[17]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[17]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[18]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[18]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[19]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[19]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[20]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[20]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[21]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[21]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[22]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[22]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[23]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[23]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[24]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[24]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[25]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[25]][1:320,]

resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[1]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[1]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[2]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[2]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[3]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[3]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[4]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[4]][1:280,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[5]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[5]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[6]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[6]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[7]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[7]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[8]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[8]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[9]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[9]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[10]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[10]][1:220,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[11]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[11]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[12]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[12]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[13]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[13]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[14]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[14]][1:340,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[15]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[15]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[16]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[16]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[17]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[17]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[18]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[18]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[19]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[19]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[20]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[20]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[21]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[21]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[22]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[22]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[23]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[23]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[24]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[24]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[25]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[25]][1:320,]

resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[1]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[1]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[2]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[2]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[3]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[3]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[4]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[4]][1:280,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[5]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[5]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[6]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[6]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[7]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[7]][1:320,]
```

```
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[8]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[8]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[9]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[9]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[10]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[10]][1:220,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[11]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[11]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[12]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[12]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[13]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[13]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[14]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[14]][1:340,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[15]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[15]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[16]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[16]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[17]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[17]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[18]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[18]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[19]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[19]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[20]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[20]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[21]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[21]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[22]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[22]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[23]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[23]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[24]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[24]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[25]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[25]][1:320,]
####################

#plotto i profili degli spotname relativi ai 4 cluster, non serve
###################
data_spotname_bspl_cluster_1_cluster_A_tot <- list()
data_spotname_bspl_cluster_1_cluster_B_tot <- list()
data_spotname_bspl_cluster_1_cluster_C_tot <- list()
data_spotname_bspl_cluster_1_cluster_D_tot <- list()

#cluster A e B
for (i in 1:23) {
  X_spotname_bspl_cluster_1_cluster_A <- resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[i]]
  storage.mode(X_spotname_bspl_cluster_1_cluster_A) <- "numeric"
  n_obs_spotname_bspl_cluster_1_cluster_A <- ncol(X_spotname_bspl_cluster_1_cluster_A)
  n_point_spotname_bspl_cluster_1_cluster_A <- nrow(X_spotname_bspl_cluster_1_cluster_A)
  grid_spotname_bspl_cluster_1_cluster_A <- seq(from = 0, to = 1,
                                                length.out = n_point_spotname_bspl_cluster_1_cluster_A)
  data_spotname_bspl_cluster_1_cluster_A <- list(X = X_spotname_bspl_cluster_1_cluster_A,
                                                 grid = grid_spotname_bspl_cluster_1_cluster_A)

  data_spotname_bspl_cluster_1_cluster_A_tot[[i]] <- data_spotname_bspl_cluster_1_cluster_A


  X_spotname_bspl_cluster_1_cluster_B <- resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[i]]
  storage.mode(X_spotname_bspl_cluster_1_cluster_B) <- "numeric"
  n_obs_spotname_bspl_cluster_1_cluster_B <- ncol(X_spotname_bspl_cluster_1_cluster_B)
  n_point_spotname_bspl_cluster_1_cluster_B <- nrow(X_spotname_bspl_cluster_1_cluster_B)
  grid_spotname_bspl_cluster_1_cluster_B <- seq(from = 0, to = 1,
                                                length.out = n_point_spotname_bspl_cluster_1_cluster_B)
  data_spotname_bspl_cluster_1_cluster_B <- list(X = X_spotname_bspl_cluster_1_cluster_B,
                                                 grid = grid_spotname_bspl_cluster_1_cluster_B)

  data_spotname_bspl_cluster_1_cluster_B_tot[[i]] <- data_spotname_bspl_cluster_1_cluster_B
}

#cluster C
for (i in c(1,3,4,9:11,13:18,20,22,23)) {
  X_spotname_bspl_cluster_1_cluster_C <- resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[i]]
  storage.mode(X_spotname_bspl_cluster_1_cluster_C) <- "numeric"
  n_obs_spotname_bspl_cluster_1_cluster_C <- ncol(X_spotname_bspl_cluster_1_cluster_C)
  n_point_spotname_bspl_cluster_1_cluster_C <- nrow(X_spotname_bspl_cluster_1_cluster_C)
  grid_spotname_bspl_cluster_1_cluster_C <- seq(from = 0, to = 1,
                                                length.out = n_point_spotname_bspl_cluster_1_cluster_C)
  data_spotname_bspl_cluster_1_cluster_C <- list(X = X_spotname_bspl_cluster_1_cluster_C,
                                                 grid = grid_spotname_bspl_cluster_1_cluster_C)

  data_spotname_bspl_cluster_1_cluster_C_tot[[i]] <- data_spotname_bspl_cluster_1_cluster_C
}

#cluster D
for (i in 18) {
  X_spotname_bspl_cluster_1_cluster_D <- resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[i]]
  storage.mode(X_spotname_bspl_cluster_1_cluster_D) <- "numeric"
  n_obs_spotname_bspl_cluster_1_cluster_D <- ncol(X_spotname_bspl_cluster_1_cluster_D)
  n_point_spotname_bspl_cluster_1_cluster_D <- nrow(X_spotname_bspl_cluster_1_cluster_D)
  grid_spotname_bspl_cluster_1_cluster_D <- seq(from = 0, to = 1,
                                                length.out = n_point_spotname_bspl_cluster_1_cluster_D)
  data_spotname_bspl_cluster_1_cluster_D <- list(X = X_spotname_bspl_cluster_1_cluster_D,
                                                 grid = grid_spotname_bspl_cluster_1_cluster_D)
```

```
  data_spotname_bspl_cluster_1_cluster_D_tot[[i]] <- data_spotname_bspl_cluster_1_cluster_D
}
####################

#Tabelle di contingenza
####################
tab_conting_filt_bspl_cluster_1 <- matrix(nrow = length(spotname_new), ncol = 4+1)

row.names(tab_conting_filt_bspl_cluster_1)<- spotname_new
colnames(tab_conting_filt_bspl_cluster_1)<- c("A","B","C","D","Totale")

#visto che filt bspl non riesce con spotname 53702 e 53703, levo ultime due righe
tab_conting_filt_bspl_cluster_1 <- tab_conting_filt_bspl_cluster_1[-c(24,25),]

#riempio tabella
for (i in 1:23) {
  tab_conting_filt_bspl_cluster_1[i,] <- c(ncol(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[i]]),
                                           ncol(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[i]]),
                                           ncol(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[i]]),
                                           ncol(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[i]]),
                                           (ncol(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_A_tot[[i]])+
                                              ncol(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_B_tot[[i]])+
                                              ncol(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_C_tot[[i]])+
                                              ncol(resistance_rawdata_spotname_bspl_hc_cluster_1_cluster_D_tot[[i]])))
}

#creo tabelle con frequenze
tab_conting_filt_bspl_cluster_1_freq_intraspotname <- prop.table(tab_conting_filt_bspl_cluster_1[,-5],
                                                                 margin = 1)

write.csv2(tab_conting_filt_bspl_cluster_1, file="tabella contingenza filtering bspl cluster 1.csv",
           quote=F, na="", row.names=T, col.names=T)
write.csv2(tab_conting_filt_bspl_cluster_1_freq_intraspotname,
           file="tabella contingenza filtering bspl cluster 1 frequenza intraspotname.csv",
           quote=F, na="", row.names=T, col.names=T)
####################

#cluster 2

#clustering
##################
clustered_funs5_hc_spotname_new_cluster_2_tot <- list()
#24 errore, 25 errore da prima clusterizzazione
for (i in 1:23) {
  if (i == 11 | 19) {
    breaks <- seq(0,0.5, by = 0.1);
    limits <- c(0,0.5);
  }
  else {
    breaks <- seq(0,0.3, by = 0.1);
    limits <- c(0,0.35);
  }

  if (length(levels(clustered_funs5_hc_spotname_new_cluster_2_tot[[i]][["cluster"]]))==2) {
    labels <- c("A","B");
  }
  else if (length(levels(clustered_funs5_hc_spotname_new_cluster_2_tot[[1]][["cluster"]]))==3) {
    labels <- c("A","B","C");
  }
  else {
    labels <- c("A","B","C","D");
  }

  mod5_spotname_new_cluster_2 <- fil_bspline_ms_nclust(data = data_spotname_bspl_cluster_2_tot[[i]],
                                                       num_cluster_seq = 2:4, nbasis = 28)
  mod5_spotname_new_cluster_2$mod_opt$ind_hc$nbclust ## hierarchical

  cl_5_hc_spotname_new_cluster_2 <- mod5_spotname_new_cluster_2$mod_opt$ind_hc$cluster

  clustered_funs5_hc_spotname_new_cluster_2 <- get_clustered_funs_df(cl = cl_5_hc_spotname_new_cluster_2,
                                                                     data = data_spotname_bspl_cluster_2_tot[[i]],
                                                                     method = "filtering B-spline\nhierarchical")
  centroids5_hc_spotname_new_cluster_2 <-
    get_centroids_df(cl_5_hc_spotname_new_cluster_2, data_spotname_bspl_cluster_2_tot[[i]],
                     "filtering B-spline\nhierarchical")

  clustered_funs5_hc_spotname_new_cluster_2_tot[[i]] <- clustered_funs5_hc_spotname_new_cluster_2

  filt_bspl_hc_spotname_new_cluster_2 <- ggplot(clustered_funs5_hc_spotname_new_cluster_2) +
    geom_line(aes(time, Resistance, group = obs, col = cluster)) +
    scale_y_continuous(breaks = breaks, limits = limits) +
    scale_fill_discrete(labels <- labels) +
    ggtitle(paste("Filtering B-spline hierarchical clustering", "\n", "of bspl cluster 2 of spotname",
                  spotname_new[i],"'s DRC"))
  # filt_bspl_hc_spotname_new_cluster_2
  filt_bspl_hc_cen_spotname_new_cluster_2 <- ggplot(centroids5_hc_spotname_new_cluster_2) +
    geom_line(aes(time, Resistance, col = cluster)) +
    scale_y_continuous(breaks = breaks, limits = limits) +
    scale_fill_discrete(labels <- labels) +
    ggtitle(paste("Centroids of filtering B-spline hierarchical clustering", "\n",
                  "of bspl cluster 2 of spotname",spotname_new[i],"'s DRC"))
  # filt_bspl_hc_cen_spotname_new_cluster_2
  full_filt_bspl__new <- ggarrange(filt_bspl_hc_spotname_new_cluster_2,filt_bspl_hc_cen_spotname_new_cluster_2,
                                   ncol=1,nrow=2)
```

```
  ggexport(full_filt_bspl__new,
           filename = paste("Filtering B-spline profili-centroidi of bspl cluster 2 of spotname",
                            spotname_new[i],"'s DRC.png"))
}
#################

#suddivido i sotto cluster in oggetti diversi
#################
scalari_cluster_2_cluster_A_bspl_hc_tot <- list()
scalari_cluster_2_cluster_B_bspl_hc_tot <- list()
scalari_cluster_2_cluster_C_bspl_hc_tot <- list()
scalari_cluster_2_cluster_D_bspl_hc_tot <- list()

for (i in 1:23) {
  profili_cluster_2_bspl_hc <- clustered_funs5_hc_spotname_new_cluster_2_tot[[i]][
    seq(from = 1, to = ncol(resistance_rawdata_spotname_bspl_hc_cluster_2_tot[[i]])),]
  profili_cluster_2_cluster_A_bspl_hc <- profili_cluster_2_bspl_hc[profili_cluster_2_bspl_hc$cluster == 1,]
  profili_cluster_2_cluster_B_bspl_hc <- profili_cluster_2_bspl_hc[profili_cluster_2_bspl_hc$cluster == 2,]
  profili_cluster_2_cluster_C_bspl_hc <- profili_cluster_2_bspl_hc[profili_cluster_2_bspl_hc$cluster == 3,]
  profili_cluster_2_cluster_D_bspl_hc <- profili_cluster_2_bspl_hc[profili_cluster_2_bspl_hc$cluster == 4,]

  #cambio nome a colonna delle obs e di cluster
  colnames(profili_cluster_2_cluster_A_bspl_hc)[2] <- "dateTime"
  colnames(profili_cluster_2_cluster_A_bspl_hc)[4] <- "cluster bspl hc A"

  colnames(profili_cluster_2_cluster_B_bspl_hc)[2] <- "dateTime"
  colnames(profili_cluster_2_cluster_B_bspl_hc)[4] <- "cluster bspl hc B"

  colnames(profili_cluster_2_cluster_C_bspl_hc)[2] <- "dateTime"
  colnames(profili_cluster_2_cluster_C_bspl_hc)[4] <- "cluster bspl hc C"

  colnames(profili_cluster_2_cluster_D_bspl_hc)[2] <- "dateTime"
  colnames(profili_cluster_2_cluster_D_bspl_hc)[4] <- "cluster bspl hc D"

  #unisco colonna relativa al cluster di appartenenza
  scalari_cluster_2_cluster_A_bspl_hc <- merge(scalari_associazione_new,profili_cluster_2_cluster_A_bspl_hc,
                                               by = ("dateTime"))
  scalari_cluster_2_cluster_B_bspl_hc <- merge(scalari_associazione_new,profili_cluster_2_cluster_B_bspl_hc,
                                               by = ("dateTime"))
  scalari_cluster_2_cluster_C_bspl_hc <- merge(scalari_associazione_new,profili_cluster_2_cluster_C_bspl_hc,
                                               by = ("dateTime"))
  scalari_cluster_2_cluster_D_bspl_hc <- merge(scalari_associazione_new,profili_cluster_2_cluster_D_bspl_hc,
                                               by = ("dateTime"))

  scalari_cluster_2_cluster_A_bspl_hc_tot[[i]] <- scalari_cluster_2_cluster_A_bspl_hc
  scalari_cluster_2_cluster_B_bspl_hc_tot[[i]] <- scalari_cluster_2_cluster_B_bspl_hc
  scalari_cluster_2_cluster_C_bspl_hc_tot[[i]] <- scalari_cluster_2_cluster_C_bspl_hc
  scalari_cluster_2_cluster_D_bspl_hc_tot[[i]] <- scalari_cluster_2_cluster_D_bspl_hc
}
#################

#creo matrici resistance rawdata
#################
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot <- list()
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot <- list()
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot <- list()
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot <- list()

for (i in 1:23) {
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A <- data_profili_new %>%
    filter(id %in% scalari_cluster_2_cluster_A_bspl_hc_tot[[i]]$dateTime ) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A<- t(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A)

  colnames(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A)<- resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A[1,]
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A<- resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A[is.na(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A)] <- 0

  storage.mode(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A) <- "numeric"       #la matrice veniva salvata come character, così div
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[i]] <- resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A
  names(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot)[[i]] <- spotname_new[i]


  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B <- data_profili_new %>%
    filter(id %in% scalari_cluster_2_cluster_B_bspl_hc_tot[[i]]$dateTime) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B<- t(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B)

  colnames(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B)<- resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B[1,]
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B<- resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B[is.na(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B)] <- 0
```

```
  storage.mode(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B) <- "numeric"       #la matrice veniva salvata come character, così div
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[i]] <- resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B
  names(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot)[[i]] <- spotname_new[i]

  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C <- data_profili_new %>%
    filter(id %in% scalari_cluster_2_cluster_C_bspl_hc_tot[[i]]$dateTime ) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C<- t(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C)

  colnames(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C)<- resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C[1,]
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C<- resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C[is.na(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C)] <- 0

  storage.mode(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C) <- "numeric"       #la matrice veniva salvata come character, così div
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[i]] <- resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C
  names(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot)[[i]] <- spotname_new[i]


  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D <- data_profili_new %>%
    filter(id %in% scalari_cluster_2_cluster_D_bspl_hc_tot[[i]]$dateTime) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D<- t(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D)

  colnames(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D)<- resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D[1,]
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D<- resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D[is.na(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D)] <- 0

  storage.mode(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D) <- "numeric"       #la matrice veniva salvata come character, così div
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[i]] <- resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D
  names(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot)[[i]] <- spotname_new[i]

}
##################

#taglio durata profili come all'inizio (240,280,300,320,340 ms)
####################
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[1]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[1]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[2]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[2]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[3]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[3]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[4]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[4]][1:280,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[5]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[5]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[6]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[6]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[7]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[7]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[8]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[8]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[9]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[9]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[10]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[10]][1:220,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[11]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[11]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[12]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[12]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[13]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[13]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[14]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[14]][1:340,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[15]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[15]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[16]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[16]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[17]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[17]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[18]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[18]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[19]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[19]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[20]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[20]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[21]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[21]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[22]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[22]][1:240,]
```

```
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[23]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[23]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[24]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[24]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[25]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[25]][1:320,]

resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[1]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[1]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[2]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[2]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[3]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[3]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[4]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[4]][1:280,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[5]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[5]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[6]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[6]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[7]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[7]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[8]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[8]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[9]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[9]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[10]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[10]][1:220,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[11]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[11]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[12]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[12]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[13]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[13]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[14]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[14]][1:340,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[15]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[15]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[16]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[16]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[17]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[17]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[18]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[18]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[19]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[19]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[20]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[20]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[21]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[21]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[22]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[22]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[23]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[23]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[24]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[24]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[25]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[25]][1:320,]

resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[1]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[1]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[2]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[2]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[3]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[3]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[4]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[4]][1:280,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[5]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[5]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[6]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[6]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[7]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[7]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[8]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[8]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[9]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[9]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[10]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[10]][1:220,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[11]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[11]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[12]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[12]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[13]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[13]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[14]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[14]][1:340,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[15]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[15]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[16]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[16]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[17]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[17]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[18]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[18]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[19]]<-
```

```
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[19]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[20]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[20]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[21]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[21]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[22]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[22]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[23]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[23]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[24]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[24]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[25]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[25]][1:320,]

resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[1]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[1]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[2]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[2]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[3]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[3]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[4]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[4]][1:280,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[5]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[5]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[6]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[6]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[7]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[7]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[8]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[8]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[9]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[9]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[10]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[10]][1:220,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[11]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[11]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[12]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[12]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[13]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[13]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[14]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[14]][1:340,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[15]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[15]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[16]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[16]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[17]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[17]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[18]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[18]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[19]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[19]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[20]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[20]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[21]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[21]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[22]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[22]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[23]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[23]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[24]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[24]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[25]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[25]][1:320,]
####################

#Tabelle di contingenza
###############
tab_conting_filt_bspl_cluster_2 <- matrix(nrow = length(spotname_new), ncol = 4+1)

row.names(tab_conting_filt_bspl_cluster_2)<- spotname_new
colnames(tab_conting_filt_bspl_cluster_2)<- c("A","B","C","D","Totale")

#visto che filt bspl non riesce con spotname 53702 e 53703, levo ultime due righe
tab_conting_filt_bspl_cluster_2 <- tab_conting_filt_bspl_cluster_2[-c(24,25),]

#riempio tabella
for (i in 1:23) {
  tab_conting_filt_bspl_cluster_2[i,] <- c(ncol(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[i]]),
                                           ncol(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[i]]),
                                           ncol(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[i]]),
                                           ncol(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[i]]),
                                           (ncol(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_A_tot[[i]])+
                                              ncol(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_B_tot[[i]])+
                                              ncol(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_C_tot[[i]])+
                                              ncol(resistance_rawdata_spotname_bspl_hc_cluster_2_cluster_D_tot[[i]])))
}

#creo tabelle con frequenze
tab_conting_filt_bspl_cluster_2_freq_intraspotname <-
  prop.table(tab_conting_filt_bspl_cluster_2[,-5],margin = 1)

write.csv2(tab_conting_filt_bspl_cluster_2, file="tabella contingenza filtering bspl cluster 2.csv",
           quote=F, na="", row.names=T, col.names=T)
write.csv2(tab_conting_filt_bspl_cluster_2_freq_intraspotname,
```

```
          file="tabella contingenza filtering bspl cluster 2 frequenza intraspotname.csv",
          quote=F, na="", row.names=T, col.names=T)
###############
```

```
#cluster 3

#clustering
#################
clustered_funs5_hc_spotname_new_cluster_3_tot <- list()
#8,9,10,11,24 errore, 25 errore da prima clusterizzazione
for (i in c(3,5:7,12,14,16,17,19,21)) {
  if (i == 11 | 19) {
    breaks <- seq(0,0.5, by = 0.1);
    limits <- c(0,0.5);
  }
  else {
    breaks <- seq(0,0.3, by = 0.1);
    limits <- c(0,0.35);
  }


  mod5_spotname_new_cluster_3 <- fil_bspline_ms_nclust(data = data_spotname_bspl_cluster_3_tot[[i]],
                                                       num_cluster_seq = 2:4, nbasis = 28)
  mod5_spotname_new_cluster_3$mod_opt$ind_hc$nbclust ## hierarchical

  cl_5_hc_spotname_new_cluster_3 <- mod5_spotname_new_cluster_3$mod_opt$ind_hc$cluster

  clustered_funs5_hc_spotname_new_cluster_3 <- get_clustered_funs_df(cl = cl_5_hc_spotname_new_cluster_3,
                                                                     data = data_spotname_bspl_cluster_3_tot[[i]],
                                                                     method = "filtering B-spline\nhierarchical")
  centroids5_hc_spotname_new_cluster_3 <-
    get_centroids_df(cl_5_hc_spotname_new_cluster_3, data_spotname_bspl_cluster_3_tot[[i]],
                     "filtering B-spline\nhierarchical")

  clustered_funs5_hc_spotname_new_cluster_3_tot[[i]] <- clustered_funs5_hc_spotname_new_cluster_3
  if (length(levels(clustered_funs5_hc_spotname_new_cluster_3_tot[[i]][["cluster"]]))==2) {
    labels <- c("A","B");
  }
  else if (length(levels(clustered_funs5_hc_spotname_new_cluster_3_tot[[i]][["cluster"]]))==3) {
    labels <- c("A","B","C");
  }
  else {
    labels <- c("A","B","C","D");
  }

  filt_bspl_hc_spotname_new_cluster_3 <- ggplot(clustered_funs5_hc_spotname_new_cluster_3) +
    geom_line(aes(time, Resistance, group = obs, col = cluster)) +
    scale_y_continuous(breaks = breaks, limits = limits) +
    scale_fill_discrete(labels <- labels) +
    ggtitle(paste("Filtering B-spline hierarchical clustering", "\n", "of bspl cluster 3 of spotname",
                  spotname_new[i],"'s DRC"))
  # filt_bspl_hc_spotname_new_cluster_3
  filt_bspl_hc_cen_spotname_new_cluster_3 <- ggplot(centroids5_hc_spotname_new_cluster_3) +
    geom_line(aes(time, Resistance, col = cluster)) +
    scale_y_continuous(breaks = breaks, limits = limits) +
    scale_fill_discrete(labels <- labels) +
    ggtitle(paste("Centroids of filtering B-spline hierarchical clustering", "\n",
                  "of bspl cluster 3 of spotname",spotname_new[i],"'s DRC"))
  # filt_bspl_hc_cen_spotname_new_cluster_3

  full_filt_bspl__new <- ggarrange(filt_bspl_hc_spotname_new_cluster_3,filt_bspl_hc_cen_spotname_new_cluster_3,
                                   ncol=1,nrow=2)
  ggexport(full_filt_bspl__new,
           filename = paste("Filtering B-spline profili-centroidi of bspl cluster 3 of spotname",
                            spotname_new[i],"'s DRC.png"))
}
#################

#suddivido i sotto cluster in oggetti diversi
#################
scalari_cluster_3_cluster_A_bspl_hc_tot <- list()
scalari_cluster_3_cluster_B_bspl_hc_tot <- list()
scalari_cluster_3_cluster_C_bspl_hc_tot <- list()
scalari_cluster_3_cluster_D_bspl_hc_tot <- list()

for (i in c(3,5:8,12,14,16,17,19,21)) {
  profili_cluster_3_bspl_hc <- clustered_funs5_hc_spotname_new_cluster_3_tot[[i]][
    seq(from = 1, to = ncol(resistance_rawdata_spotname_bspl_hc_cluster_3_tot[[i]])),]
  profili_cluster_3_cluster_A_bspl_hc <- profili_cluster_3_bspl_hc[profili_cluster_3_bspl_hc$cluster == 1,]
  profili_cluster_3_cluster_B_bspl_hc <- profili_cluster_3_bspl_hc[profili_cluster_3_bspl_hc$cluster == 2,]
  profili_cluster_3_cluster_C_bspl_hc <- profili_cluster_3_bspl_hc[profili_cluster_3_bspl_hc$cluster == 3,]
  profili_cluster_3_cluster_D_bspl_hc <- profili_cluster_3_bspl_hc[profili_cluster_3_bspl_hc$cluster == 4,]

  #cambio nome a colonna delle obs e di cluster
  colnames(profili_cluster_3_cluster_A_bspl_hc)[2] <- "dateTime"
  colnames(profili_cluster_3_cluster_A_bspl_hc)[4] <- "cluster bspl hc A"

  colnames(profili_cluster_3_cluster_B_bspl_hc)[2] <- "dateTime"
  colnames(profili_cluster_3_cluster_B_bspl_hc)[4] <- "cluster bspl hc B"

  colnames(profili_cluster_3_cluster_C_bspl_hc)[2] <- "dateTime"
  colnames(profili_cluster_3_cluster_C_bspl_hc)[4] <- "cluster bspl hc C"

  colnames(profili_cluster_3_cluster_D_bspl_hc)[2] <- "dateTime"
  colnames(profili_cluster_3_cluster_D_bspl_hc)[4] <- "cluster bspl hc D"
```

```
  #unisco colonna relativa al cluster di appartenenza
  scalari_cluster_3_cluster_A_bspl_hc <- merge(scalari_associazione_new,profili_cluster_3_cluster_A_bspl_hc,
                                               by = ("dateTime"))
  scalari_cluster_3_cluster_B_bspl_hc <- merge(scalari_associazione_new,profili_cluster_3_cluster_B_bspl_hc,
                                               by = ("dateTime"))
  scalari_cluster_3_cluster_C_bspl_hc <- merge(scalari_associazione_new,profili_cluster_3_cluster_C_bspl_hc,
                                               by = ("dateTime"))
  scalari_cluster_3_cluster_D_bspl_hc <- merge(scalari_associazione_new,profili_cluster_3_cluster_D_bspl_hc,
                                               by = ("dateTime"))

  scalari_cluster_3_cluster_A_bspl_hc_tot[[i]] <- scalari_cluster_3_cluster_A_bspl_hc
  scalari_cluster_3_cluster_B_bspl_hc_tot[[i]] <- scalari_cluster_3_cluster_B_bspl_hc
  scalari_cluster_3_cluster_C_bspl_hc_tot[[i]] <- scalari_cluster_3_cluster_C_bspl_hc
  scalari_cluster_3_cluster_D_bspl_hc_tot[[i]] <- scalari_cluster_3_cluster_D_bspl_hc

}
##################

#creo matrici resistance rawdata
##################
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot <- list()
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot <- list()
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot <- list()
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot <- list()

for (i in c(3,5:8,12,14,16,17,19,21)) {
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A <- data_profili_new %>%
    filter(id %in% scalari_cluster_3_cluster_A_bspl_hc_tot[[i]]$dateTime ) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A<-
    t(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A)

  colnames(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A)<-
    resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A[1,]
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A<-
    resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A[
    is.na(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A)] <- 0

  storage.mode(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A) <- "numeric"
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[i]] <-
    resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A
  names(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot)[[i]] <- spotname_new[i]


  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B <- data_profili_new %>%
    filter(id %in% scalari_cluster_3_cluster_B_bspl_hc_tot[[i]]$dateTime) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B<-
    t(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B)

  colnames(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B)<-
    resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B[1,]
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B<-
    resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B[
    is.na(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B)] <- 0

  storage.mode(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B) <- "numeric"
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[i]] <-
    resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B
  names(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot)[[i]] <- spotname_new[i]

  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C <- data_profili_new %>%
    filter(id %in% scalari_cluster_3_cluster_C_bspl_hc_tot[[i]]$dateTime ) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C<-
    t(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C)

  colnames(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C)<-
    resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C[1,]
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C<-
    resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C[-1,]

  #Sostituisco i valori NA con 0
```

```
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C[
    is.na(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C)] <- 0

  storage.mode(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C) <- "numeric"
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[i]] <-
    resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C
  names(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot)[[i]] <- spotname_new[i]


  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D <- data_profili_new %>%
    filter(id %in% scalari_cluster_3_cluster_D_bspl_hc_tot[[i]]$dateTime) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D<-
    t(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D)

  colnames(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D)<-
    resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D[1,]
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D<-
    resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D[
    is.na(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D)] <- 0

  storage.mode(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D) <- "numeric"
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[i]] <-
    resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D
  names(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot)[[i]] <- spotname_new[i]

}
##################

#taglio durata profili come all'inizio (240,280,300,320,340 ms)
####################
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[1]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[1]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[2]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[2]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[3]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[3]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[4]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[4]][1:280,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[5]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[5]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[6]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[6]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[7]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[7]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[8]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[8]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[9]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[9]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[10]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[10]][1:220,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[11]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[11]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[12]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[12]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[13]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[13]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[14]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[14]][1:340,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[15]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[15]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[16]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[16]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[17]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[17]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[18]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[18]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[19]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[19]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[20]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[20]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[21]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[21]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[22]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[22]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[23]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[23]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[24]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[24]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[25]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[25]][1:320,]

resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[1]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[1]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[2]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[2]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[3]]<-
```

```
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[3]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[4]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[4]][1:280,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[5]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[5]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[6]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[6]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[7]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[7]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[8]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[8]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[9]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[9]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[10]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[10]][1:220,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[11]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[11]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[12]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[12]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[13]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[13]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[14]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[14]][1:340,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[15]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[15]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[16]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[16]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[17]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[17]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[18]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[18]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[19]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[19]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[20]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[20]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[21]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[21]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[22]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[22]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[23]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[23]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[24]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[24]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[25]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[25]][1:320,]

resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[1]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[1]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[2]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[2]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[3]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[3]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[4]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[4]][1:280,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[5]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[5]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[6]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[6]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[7]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[7]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[8]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[8]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[9]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[9]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[10]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[10]][1:220,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[11]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[11]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[12]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[12]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[13]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[13]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[14]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[14]][1:340,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[15]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[15]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[16]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[16]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[17]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[17]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[18]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[18]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[19]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[19]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[20]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[20]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[21]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[21]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[22]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[22]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[23]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[23]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[24]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[24]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[25]]<-
```

```
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[25]][1:320,]

resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[1]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[1]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[2]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[2]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[3]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[3]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[4]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[4]][1:280,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[5]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[5]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[6]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[6]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[7]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[7]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[8]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[8]][1:300,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[9]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[9]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[10]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[10]][1:220,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[11]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[11]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[12]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[12]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[13]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[13]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[14]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[14]][1:340,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[15]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[15]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[16]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[16]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[17]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[17]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[18]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[18]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[19]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[19]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[20]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[20]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[21]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[21]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[22]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[22]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[23]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[23]][1:240,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[24]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[24]][1:320,]
resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[25]]<-
  resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[25]][1:320,]
####################

#Tabelle di contingenza
###############
tab_conting_filt_bspl_cluster_3 <- matrix(nrow = length(spotname_new), ncol = 4+1)

row.names(tab_conting_filt_bspl_cluster_3)<- spotname_new
colnames(tab_conting_filt_bspl_cluster_3)<- c("A","B","C","D","Totale")

#visto che filt bspl non riesce con spotname 53702 e 53703, levo ultime due righe
tab_conting_filt_bspl_cluster_3 <- tab_conting_filt_bspl_cluster_3[-c(24,25),]

#riempio tabella
for (i in c(3,5:8,12,14,16,17,19,21)) {
 tab_conting_filt_bspl_cluster_3[i,]<-c(ncol(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[i]]),
                                        ncol(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[i]]),
                                        ncol(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[i]]),
                                        ncol(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[i]]),
                                        (ncol(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_A_tot[[i]])+
                                           ncol(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_B_tot[[i]])+
                                           ncol(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_C_tot[[i]])+
                                           ncol(resistance_rawdata_spotname_bspl_hc_cluster_3_cluster_D_tot[[i]])))
}

#creo tabelle con frequenze
tab_conting_filt_bspl_cluster_3_freq_intraspotname <- prop.table(tab_conting_filt_bspl_cluster_3[,-5],margin = 1)

write.csv2(tab_conting_filt_bspl_cluster_3, file="tabella contingenza filtering bspl cluster 3.csv",
           quote=F, na="", row.names=T, col.names=T)
write.csv2(tab_conting_filt_bspl_cluster_3_freq_intraspotname,
           file="tabella contingenza filtering bspl cluster 3 frequenza intraspotname.csv",
           quote=F, na="", row.names=T, col.names=T)
###############




#FPCA basis - Prima clusterizzazione
###############
clustered_funs6_hc_spotname_new_tot <- list()

for (i in 1:length(spotname_new)) {
```

```r
  mod6_spotname_new <- fil_fpca_ss_nbclust(data = data_spotname_new_tot[[i]],
                                           num_cluster_seq = 2:4, per_comp = 0.8)
  mod6_spotname_new$mod_opt$ind_hc$nbclust ## hierarchical

  cl_6_hc_spotname_new <- mod6_spotname_new$mod_opt$ind_hc$cluster

  clustered_funs6_hc_spotname_new <- get_clustered_funs_df(cl = cl_6_hc_spotname_new,
                                                           data = data_spotname_new_tot[[i]],
                                                           method = "filtering FPCA\nhierarchical")
  centroids6_hc_spotname_new <- get_centroids_df(cl_6_hc_spotname_new, data_spotname_new_tot[[i]],
                                                 "filtering FPCA\nhierarchical")

  clustered_funs6_hc_spotname_new_tot[[i]] <- clustered_funs6_hc_spotname_new

  filt_fpca_hc_spotname_new <- ggplot(clustered_funs6_hc_spotname_new) +
    geom_line(aes(time, Resistance, group = obs, col = cluster)) +
    ggtitle(paste("Filtering FPCA hierarchical clustering", "\n", "of spotname",spotname_new[i],"'s DRC"))
  # filt_fpca_hc_spotname_new
  filt_fpca_hc_cen_spotname_new <- ggplot(centroids6_hc_spotname_new) +
    geom_line(aes(time, Resistance, col = cluster)) +
    ggtitle(paste("Centroids of filtering FPCA hierarchical clustering", "\n", "of spotname",
                  spotname_new[i],"'s DRC"))
  # filt_fpca_hc_cen_spotname_new

  full_filt_fpca__new <- ggarrange(filt_fpca_hc_spotname_new,filt_fpca_hc_cen_spotname_new,ncol=1,nrow=2)
  ggexport(full_filt_fpca__new, filename = paste("Filtering FPCA profili-centroidi of spotname",
                                                 spotname_new[i],"'s DRC.png"))
}
###############

#suddivido i cluster in oggetti diversi
###############
scalari_cluster_1_fpca_hc_tot <- list()
scalari_cluster_2_fpca_hc_tot <- list()
scalari_cluster_3_fpca_hc_tot <- list()
scalari_cluster_4_fpca_hc_tot <- list()

for (i in 1:25) {
  profili_cluster_fpca_hc <- clustered_funs6_hc_spotname_new_tot[[i]][
    seq(from = 1, to = ncol(resistance_rawdata_spotname_new_tot[[i]])),]
  profili_cluster_1_fpca_hc <- profili_cluster_fpca_hc[profili_cluster_fpca_hc$cluster == 1,]
  profili_cluster_2_fpca_hc <- profili_cluster_fpca_hc[profili_cluster_fpca_hc$cluster == 2,]
  profili_cluster_3_fpca_hc <- profili_cluster_fpca_hc[profili_cluster_fpca_hc$cluster == 3,]
  profili_cluster_4_fpca_hc <- profili_cluster_fpca_hc[profili_cluster_fpca_hc$cluster == 4,]

  #cambio nome a colonna delle obs e di cluster
  colnames(profili_cluster_1_fpca_hc)[2] <- "dateTime"
  colnames(profili_cluster_1_fpca_hc)[4] <- "cluster fpca hc"

  colnames(profili_cluster_2_fpca_hc)[2] <- "dateTime"
  colnames(profili_cluster_2_fpca_hc)[4] <- "cluster fpca hc"

  colnames(profili_cluster_3_fpca_hc)[2] <- "dateTime"
  colnames(profili_cluster_3_fpca_hc)[4] <- "cluster fpca hc"

  colnames(profili_cluster_4_fpca_hc)[2] <- "dateTime"
  colnames(profili_cluster_4_fpca_hc)[4] <- "cluster fpca hc"

  #unisco colonna relativa al cluster di appartenenza
  scalari_cluster_1_fpca_hc <- merge(scalari_associazione_new,profili_cluster_1_fpca_hc,by = ("dateTime"))
  scalari_cluster_2_fpca_hc <- merge(scalari_associazione_new,profili_cluster_2_fpca_hc,by = ("dateTime"))
  scalari_cluster_3_fpca_hc <- merge(scalari_associazione_new,profili_cluster_3_fpca_hc,by = ("dateTime"))
  scalari_cluster_4_fpca_hc <- merge(scalari_associazione_new,profili_cluster_4_fpca_hc,by = ("dateTime"))

  # profili_cluster_1_fpca_hc_tot[[i]] <- profili_cluster_1_fpca_hc
  # profili_cluster_2_fpca_hc_tot[[i]] <- profili_cluster_2_fpca_hc
  # profili_cluster_3_fpca_hc_tot[[i]] <- profili_cluster_3_fpca_hc
  # profili_cluster_4_fpca_hc_tot[[i]] <- profili_cluster_4_fpca_hc

  scalari_cluster_1_fpca_hc_tot[[i]] <- scalari_cluster_1_fpca_hc
  scalari_cluster_2_fpca_hc_tot[[i]] <- scalari_cluster_2_fpca_hc
  scalari_cluster_3_fpca_hc_tot[[i]] <- scalari_cluster_3_fpca_hc
  scalari_cluster_4_fpca_hc_tot[[i]] <- scalari_cluster_4_fpca_hc

}
###############

#creo matrici resistance rawdata
###############
resistance_rawdata_spotname_fpca_hc_cluster_1_tot <- list()
resistance_rawdata_spotname_fpca_hc_cluster_2_tot <- list()
resistance_rawdata_spotname_fpca_hc_cluster_3_tot <- list()
resistance_rawdata_spotname_fpca_hc_cluster_4_tot <- list()

for (i in 1:25) {
  resistance_rawdata_spotname_fpca_hc_cluster_1 <- data_profili_new %>%
    filter(id %in% scalari_cluster_1_fpca_hc_tot[[i]]$dateTime ) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_fpca_hc_cluster_1<- t(resistance_rawdata_spotname_fpca_hc_cluster_1)
```

```
  colnames(resistance_rawdata_spotname_fpca_hc_cluster_1)<- resistance_rawdata_spotname_fpca_hc_cluster_1[1,]
  resistance_rawdata_spotname_fpca_hc_cluster_1<- resistance_rawdata_spotname_fpca_hc_cluster_1[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_fpca_hc_cluster_1[is.na(resistance_rawdata_spotname_fpca_hc_cluster_1)] <- 0

  storage.mode(resistance_rawdata_spotname_fpca_hc_cluster_1) <- "numeric"
  resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[i]] <- resistance_rawdata_spotname_fpca_hc_cluster_1
  names(resistance_rawdata_spotname_fpca_hc_cluster_1_tot)[[i]] <- spotname_new[i]


  resistance_rawdata_spotname_fpca_hc_cluster_2 <- data_profili_new %>%
    filter(id %in% scalari_cluster_2_fpca_hc_tot[[i]]$dateTime) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_fpca_hc_cluster_2<- t(resistance_rawdata_spotname_fpca_hc_cluster_2)

  colnames(resistance_rawdata_spotname_fpca_hc_cluster_2)<- resistance_rawdata_spotname_fpca_hc_cluster_2[1,]
  resistance_rawdata_spotname_fpca_hc_cluster_2<- resistance_rawdata_spotname_fpca_hc_cluster_2[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_fpca_hc_cluster_2[is.na(resistance_rawdata_spotname_fpca_hc_cluster_2)] <- 0

  storage.mode(resistance_rawdata_spotname_fpca_hc_cluster_2) <- "numeric"
  resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[i]] <- resistance_rawdata_spotname_fpca_hc_cluster_2
  names(resistance_rawdata_spotname_fpca_hc_cluster_2_tot)[[i]] <- spotname_new[i]

  resistance_rawdata_spotname_fpca_hc_cluster_3 <- data_profili_new %>%
    filter(id %in% scalari_cluster_3_fpca_hc_tot[[i]]$dateTime ) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_fpca_hc_cluster_3<- t(resistance_rawdata_spotname_fpca_hc_cluster_3)

  colnames(resistance_rawdata_spotname_fpca_hc_cluster_3)<- resistance_rawdata_spotname_fpca_hc_cluster_3[1,]
  resistance_rawdata_spotname_fpca_hc_cluster_3<- resistance_rawdata_spotname_fpca_hc_cluster_3[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_fpca_hc_cluster_3[is.na(resistance_rawdata_spotname_fpca_hc_cluster_3)] <- 0

  storage.mode(resistance_rawdata_spotname_fpca_hc_cluster_3) <- "numeric"
  resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[i]] <- resistance_rawdata_spotname_fpca_hc_cluster_3
  names(resistance_rawdata_spotname_fpca_hc_cluster_3_tot)[[i]] <- spotname_new[i]


  resistance_rawdata_spotname_fpca_hc_cluster_4 <- data_profili_new %>%
    filter(id %in% scalari_cluster_4_fpca_hc_tot[[i]]$dateTime) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_fpca_hc_cluster_4<- t(resistance_rawdata_spotname_fpca_hc_cluster_4)

  colnames(resistance_rawdata_spotname_fpca_hc_cluster_4)<- resistance_rawdata_spotname_fpca_hc_cluster_4[1,]
  resistance_rawdata_spotname_fpca_hc_cluster_4<- resistance_rawdata_spotname_fpca_hc_cluster_4[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_fpca_hc_cluster_4[is.na(resistance_rawdata_spotname_fpca_hc_cluster_4)] <- 0

  storage.mode(resistance_rawdata_spotname_fpca_hc_cluster_4) <- "numeric"
  resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[i]] <- resistance_rawdata_spotname_fpca_hc_cluster_4
  names(resistance_rawdata_spotname_fpca_hc_cluster_4_tot)[[i]] <- spotname_new[i]
}
###############

#taglio durata profili come all'inizio (240,280,300,320,340 ms)
####################
resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[1]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[1]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[2]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[2]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[3]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[3]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[4]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[4]][1:280,]
resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[5]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[5]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[6]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[6]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[7]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[7]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[8]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[8]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[9]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[9]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[10]]<-
```

```
  resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[10]][1:220,]
resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[11]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[11]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[12]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[12]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[13]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[13]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[14]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[14]][1:340,]
resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[15]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[15]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[16]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[16]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[17]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[17]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[18]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[18]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[19]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[19]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[20]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[20]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[21]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[21]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[22]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[22]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[23]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[23]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[24]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[24]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[25]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[25]][1:320,]

resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[1]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[1]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[2]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[2]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[3]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[3]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[4]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[4]][1:280,]
resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[5]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[5]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[6]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[6]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[7]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[7]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[8]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[8]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[9]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[9]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[10]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[10]][1:220,]
resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[11]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[11]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[12]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[12]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[13]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[13]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[14]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[14]][1:340,]
resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[15]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[15]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[16]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[16]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[17]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[17]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[18]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[18]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[19]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[19]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[20]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[20]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[21]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[21]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[22]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[22]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[23]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[23]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[24]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[24]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[25]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[25]][1:320,]

resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[1]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[1]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[2]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[2]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[3]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[3]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[4]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[4]][1:280,]
resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[5]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[5]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[6]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[6]][1:320,]
```

```
resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[7]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[7]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[8]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[8]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[9]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[9]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[10]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[10]][1:220,]
resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[11]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[11]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[12]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[12]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[13]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[13]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[14]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[14]][1:340,]
resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[15]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[15]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[16]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[16]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[17]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[17]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[18]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[18]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[19]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[19]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[20]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[20]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[21]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[21]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[22]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[22]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[23]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[23]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[24]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[24]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[25]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[25]][1:320,]

resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[1]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[1]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[2]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[2]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[3]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[3]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[4]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[4]][1:280,]
resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[5]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[5]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[6]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[6]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[7]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[7]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[8]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[8]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[9]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[9]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[10]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[10]][1:220,]
resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[11]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[11]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[12]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[12]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[13]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[13]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[14]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[14]][1:340,]
resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[15]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[15]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[16]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[16]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[17]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[17]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[18]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[18]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[19]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[19]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[20]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[20]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[21]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[21]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[22]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[22]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[23]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[23]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[24]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[24]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[25]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[25]][1:320,]
####################

#Creazione tabelle di contingenza: per ogni spotname avrò la ripartizione nei vari cluster
##############
tab_conting_filt_fpca <- matrix(nrow = length(spotname_new), ncol = 4+1)
```

```r
row.names(tab_conting_filt_fpca)<- spotname_new
colnames(tab_conting_filt_fpca)<- c(1,2,3,4,"Totale")

#riempio tabella
for (i in 1:25) {
  tab_conting_filt_fpca[i,] <- c(ncol(resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[i]]),
                                 ncol(resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[i]]),
                                 ncol(resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[i]]),
                                 ncol(resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[i]]),
                                 (ncol(resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[i]])+
                                    ncol(resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[i]])+
                                    ncol(resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[i]])+
                                    ncol(resistance_rawdata_spotname_fpca_hc_cluster_4_tot[[i]])))
}

#creo tabelle con frequenze
tab_conting_filt_fpca_freq_intraspotname <- prop.table(tab_conting_filt_fpca[,-5],margin = 1)

write.csv2(tab_conting_filt_fpca, file="tabella contingenza filtering fpca.csv",
           quote=F, na="", row.names=T, col.names=T)
write.csv2(tab_conting_filt_fpca_freq_intraspotname,
           file="tabella contingenza filtering fpca frequenza intraspotname.csv",
           quote=F, na="", row.names=T, col.names=T)
###############



#Filtering FPCA - Seconda clusterizzazione
#clusterizzo i singoli cluster

#cluster 1

#clustering
###############
clustered_funs6_hc_spotname_new_cluster_1_tot <- list()

for (i in 1:23) {
  mod6_spotname_new_cluster_1 <-
    fil_fpca_ss_nbclust(data = data_spotname_fpca_cluster_1_tot[[i]], num_cluster_seq = 2:4, per_comp = 0.8)
  mod6_spotname_new_cluster_1$mod_opt$ind_hc$nbclust ## hierarchical

  cl_6_hc_spotname_new_cluster_1 <- mod6_spotname_new_cluster_1$mod_opt$ind_hc$cluster

  clustered_funs6_hc_spotname_new_cluster_1 <-
    get_clustered_funs_df(cl = cl_6_hc_spotname_new_cluster_1,
                          data = data_spotname_fpca_cluster_1_tot[[i]],
                          method = "filtering FPCA\nhierarchical")
  centroids6_hc_spotname_new_cluster_1 <-
    get_centroids_df(cl_6_hc_spotname_new_cluster_1, data_spotname_fpca_cluster_1_tot[[i]],
                     "filtering FPCA\nhierarchical")

  clustered_funs6_hc_spotname_new_cluster_1_tot[[i]] <- clustered_funs6_hc_spotname_new_cluster_1
}
for (i in 1:23) {

  if (length(levels(clustered_funs6_hc_spotname_new_cluster_1_tot[[i]][["cluster"]]))==2) {
    labels1 <- c("A","B");
    values1 <- c("#F8766D", "#00BFC4");
  }
  else if (length(levels(clustered_funs6_hc_spotname_new_cluster_1_tot[[i]][["cluster"]]))==3) {
    labels1 <- c("A","B","C");
    values1 <- c("#F8766D", "#00BFC4", "#7CAE00");
  }
  else if (length(levels(clustered_funs6_hc_spotname_new_cluster_1_tot[[i]][["cluster"]]))==4) {
    labels1 <- c("A","B","C","D");
    values1 <- c("#F8766D", "#00BFC4", "#7CAE00", "#C77CFF");
  }
  else {
    labels1 <- c("A","B","C","D","E");
    values1 <- c("#F8766D", "#00BFC4", "#7CAE00", "#C77CFF","#7FFFD4");
  }

  filt_fpca_hc_spotname_new_cluster_1 <- ggplot(clustered_funs6_hc_spotname_new_cluster_1_tot[[i]]) +
    geom_line(aes(time, Resistance, group = obs, col = cluster)) +
    scale_y_continuous(breaks = seq(0.05,0.35, by = 0.05), limits = c(0.05,0.35)) +
    labs(title = "DRC\n", x = "Time", y = "Resistance", color = "Sotto cluster\n") +
    scale_color_manual(labels = labels1, values = values1) +
    ggtitle(paste("Filtering FPCA hierarchical clustering", "\n",
                  "of fpca cluster 1 of spotname",spotname_new[i],"'s DRC"))
  # filt_fpca_hc_spotname_new_cluster_1
  filt_fpca_hc_cen_spotname_new_cluster_1 <- ggplot(clustered_funs6_hc_spotname_new_cluster_1_tot[[i]]) +
    geom_line(aes(time, Resistance, col = cluster)) +
    scale_y_continuous(breaks = seq(0.05,0.35, by = 0.05), limits = c(0.05,0.35)) +
    labs(title = "DRC\n", x = "Time", y = "Resistance", color = "Sotto cluster\n") +
    scale_color_manual(labels = labels1, values = values1) +
    ggtitle(paste("Centroids of filtering FPCA hierarchical clustering", "\n",
                  "of fpca cluster 1 of spotname",spotname_new[i],"'s DRC"))
  # filt_fpca_hc_cen_spotname_new_cluster_1

  full_filt_fpca__new <- ggarrange(filt_fpca_hc_spotname_new_cluster_1,
                                   filt_fpca_hc_cen_spotname_new_cluster_1,ncol=1,nrow=2)
  ggexport(full_filt_fpca__new,
           filename = paste("Filtering FPCA profili-centroidi of fpca cluster 1 of spotname",
                            spotname_new[i],"'s DRC.png"))
```

```
}
###############

#suddivido i sotto cluster in oggetti diversi
#################
scalari_cluster_1_cluster_A_fpca_hc_tot <- list()
scalari_cluster_1_cluster_B_fpca_hc_tot <- list()
scalari_cluster_1_cluster_C_fpca_hc_tot <- list()
scalari_cluster_1_cluster_D_fpca_hc_tot <- list()

for (i in 1:23) {
  profili_cluster_1_fpca_hc <-
    clustered_funs6_hc_spotname_new_cluster_1_tot[[i]][
      seq(from = 1, to = ncol(resistance_rawdata_spotname_fpca_hc_cluster_1_tot[[i]])),]
  profili_cluster_1_cluster_A_fpca_hc <- profili_cluster_1_fpca_hc[profili_cluster_1_fpca_hc$cluster == 1,]
  profili_cluster_1_cluster_B_fpca_hc <- profili_cluster_1_fpca_hc[profili_cluster_1_fpca_hc$cluster == 2,]
  profili_cluster_1_cluster_C_fpca_hc <- profili_cluster_1_fpca_hc[profili_cluster_1_fpca_hc$cluster == 3,]
  profili_cluster_1_cluster_D_fpca_hc <- profili_cluster_1_fpca_hc[profili_cluster_1_fpca_hc$cluster == 4,]

  #cambio nome a colonna delle obs e di cluster
  colnames(profili_cluster_1_cluster_A_fpca_hc)[2] <- "dateTime"
  colnames(profili_cluster_1_cluster_A_fpca_hc)[4] <- "cluster fpca hc A"

  colnames(profili_cluster_1_cluster_B_fpca_hc)[2] <- "dateTime"
  colnames(profili_cluster_1_cluster_B_fpca_hc)[4] <- "cluster fpca hc B"

  colnames(profili_cluster_1_cluster_C_fpca_hc)[2] <- "dateTime"
  colnames(profili_cluster_1_cluster_C_fpca_hc)[4] <- "cluster fpca hc C"

  colnames(profili_cluster_1_cluster_D_fpca_hc)[2] <- "dateTime"
  colnames(profili_cluster_1_cluster_D_fpca_hc)[4] <- "cluster fpca hc D"

  #unisco colonna relativa al cluster di appartenenza
  scalari_cluster_1_cluster_A_fpca_hc <-
    merge(scalari_associazione_new,profili_cluster_1_cluster_A_fpca_hc,by = ("dateTime"))
  scalari_cluster_1_cluster_B_fpca_hc <-
    merge(scalari_associazione_new,profili_cluster_1_cluster_B_fpca_hc,by = ("dateTime"))
  scalari_cluster_1_cluster_C_fpca_hc <-
    merge(scalari_associazione_new,profili_cluster_1_cluster_C_fpca_hc,by = ("dateTime"))
  scalari_cluster_1_cluster_D_fpca_hc <-
    merge(scalari_associazione_new,profili_cluster_1_cluster_D_fpca_hc,by = ("dateTime"))

  scalari_cluster_1_cluster_A_fpca_hc_tot[[i]] <- scalari_cluster_1_cluster_A_fpca_hc
  scalari_cluster_1_cluster_B_fpca_hc_tot[[i]] <- scalari_cluster_1_cluster_B_fpca_hc
  scalari_cluster_1_cluster_C_fpca_hc_tot[[i]] <- scalari_cluster_1_cluster_C_fpca_hc
  scalari_cluster_1_cluster_D_fpca_hc_tot[[i]] <- scalari_cluster_1_cluster_D_fpca_hc
}
#################
```

```
#creo matrici resistance rawdata
#################
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot <- list()
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot <- list()
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot <- list()
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot <- list()

for (i in 1:23) {
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A <- data_profili_new %>%
    filter(id %in% scalari_cluster_1_cluster_A_fpca_hc_tot[[i]]$dateTime ) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A<-
    t(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A)

  colnames(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A)<-
    resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A[1,]
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A<-
    resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A[
    is.na(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A)] <- 0

  storage.mode(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A) <- "numeric"
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[i]] <-
    resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A
  names(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot)[[i]] <- spotname_new[i]


  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B <- data_profili_new %>%
    filter(id %in% scalari_cluster_1_cluster_B_fpca_hc_tot[[i]]$dateTime) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B<-
    t(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B)

  colnames(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B)<-
```

```
    resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B[1,]
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B<-
    resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B[
    is.na(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B)] <- 0

  storage.mode(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B) <- "numeric"
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[i]] <-
    resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B
  names(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot)[[i]] <- spotname_new[i]

  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C <- data_profili_new %>%
    filter(id %in% scalari_cluster_1_cluster_C_fpca_hc_tot[[i]]$dateTime ) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C<-
    t(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C)

  colnames(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C)<-
    resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C[1,]
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C<-
    resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C[
    is.na(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C)] <- 0

  storage.mode(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C) <- "numeric"
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[i]] <-
    resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C
  names(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot)[[i]] <- spotname_new[i]


  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D <- data_profili_new %>%
    filter(id %in% scalari_cluster_1_cluster_D_fpca_hc_tot[[i]]$dateTime) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D<-
    t(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D)

  colnames(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D)<-
    resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D[1,]
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D<-
    resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D[
    is.na(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D)] <- 0

  storage.mode(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D) <- "numeric"
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[i]] <-
    resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D
  names(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot)[[i]] <- spotname_new[i]

}
##################

#taglio durata profili come all'inizio (240,280,300,320,340 ms)
####################
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[1]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[1]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[2]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[2]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[3]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[3]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[4]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[4]][1:280,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[5]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[5]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[6]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[6]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[7]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[7]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[8]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[8]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[9]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[9]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[10]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[10]][1:220,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[11]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[11]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[12]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[12]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[13]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[13]][1:240,]
```

```
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[14]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[14]][1:340,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[15]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[15]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[16]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[16]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[17]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[17]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[18]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[18]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[19]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[19]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[20]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[20]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[21]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[21]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[22]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[22]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[23]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[23]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[24]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[24]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[25]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[25]][1:320,]

resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[1]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[1]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[2]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[2]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[3]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[3]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[4]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[4]][1:280,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[5]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[5]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[6]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[6]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[7]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[7]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[8]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[8]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[9]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[9]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[10]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[10]][1:220,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[11]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[11]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[12]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[12]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[13]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[13]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[14]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[14]][1:340,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[15]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[15]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[16]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[16]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[17]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[17]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[18]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[18]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[19]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[19]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[20]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[20]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[21]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[21]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[22]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[22]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[23]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[23]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[24]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[24]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[25]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[25]][1:320,]

resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[1]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[1]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[2]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[2]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[3]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[3]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[4]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[4]][1:280,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[5]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[5]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[6]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[6]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[7]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[7]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[8]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[8]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[9]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[9]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[10]]<-
```

```
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[10]][1:220,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[11]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[11]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[12]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[12]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[13]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[13]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[14]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[14]][1:340,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[15]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[15]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[16]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[16]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[17]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[17]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[18]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[18]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[19]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[19]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[20]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[20]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[21]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[21]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[22]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[22]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[23]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[23]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[24]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[24]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[25]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[25]][1:320,]

resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[1]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[1]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[2]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[2]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[3]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[3]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[4]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[4]][1:280,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[5]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[5]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[6]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[6]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[7]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[7]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[8]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[8]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[9]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[9]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[10]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[10]][1:220,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[11]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[11]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[12]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[12]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[13]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[13]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[14]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[14]][1:340,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[15]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[15]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[16]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[16]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[17]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[17]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[18]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[18]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[19]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[19]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[20]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[20]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[21]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[21]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[22]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[22]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[23]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[23]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[24]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[24]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[25]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[25]][1:320,]
####################

#Tabelle di contingenza
####################
tab_conting_filt_fpca_cluster_1 <- matrix(nrow = length(spotname_new), ncol = 4+1)

row.names(tab_conting_filt_fpca_cluster_1)<- spotname_new
colnames(tab_conting_filt_fpca_cluster_1)<- c("A","B","C","D","Totale")

#visto che filt fpca non riesce con spotname 53702 e 53703, levo ultime due righe
tab_conting_filt_fpca_cluster_1 <- tab_conting_filt_fpca_cluster_1[-c(24,25),]

#riempio tabella
```

```
for (i in 1:23) {
  tab_conting_filt_fpca_cluster_1[i,] <- c(ncol(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[i]]),
                                           ncol(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[i]]),
                                           ncol(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[i]]),
                                           ncol(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[i]]),
                                           (ncol(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_A_tot[[i]])+
                                              ncol(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_B_tot[[i]])+
                                              ncol(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_C_tot[[i]])+
                                              ncol(resistance_rawdata_spotname_fpca_hc_cluster_1_cluster_D_tot[[i]])))
}

#creo tabelle con frequenze
tab_conting_filt_fpca_cluster_1_freq_intraspotname <-
  prop.table(tab_conting_filt_fpca_cluster_1[,-5],margin = 1)

write.csv2(tab_conting_filt_fpca_cluster_1, file="tabella contingenza filtering fpca cluster 1.csv",
           quote=F, na="", row.names=T, col.names=T)
write.csv2(tab_conting_filt_fpca_cluster_1_freq_intraspotname,
           file="tabella contingenza filtering fpca cluster 1 frequenza intraspotname.csv",
           quote=F, na="", row.names=T, col.names=T)
####################


#cluster 2

#clustering
###############
clustered_funs6_hc_spotname_new_cluster_2_tot <- list()

for (i in 1:25) {
  mod6_spotname_new_cluster_2 <-
    fil_fpca_ss_nbclust(data = data_spotname_fpca_cluster_2_tot[[i]], num_cluster_seq = 2:4, per_comp = 0.8)
  mod6_spotname_new_cluster_2$mod_opt$ind_hc$nbclust ## hierarchical

  cl_6_hc_spotname_new_cluster_2 <- mod6_spotname_new_cluster_2$mod_opt$ind_hc$cluster

  clustered_funs6_hc_spotname_new_cluster_2 <-
    get_clustered_funs_df(cl = cl_6_hc_spotname_new_cluster_2,
                          data = data_spotname_fpca_cluster_2_tot[[i]],
                          method = "filtering FPCA\nhierarchical")
  centroids6_hc_spotname_new_cluster_2 <-
    get_centroids_df(cl_6_hc_spotname_new_cluster_2, data_spotname_fpca_cluster_2_tot[[i]],
                     "filtering FPCA\nhierarchical")

  clustered_funs6_hc_spotname_new_cluster_2_tot[[i]] <- clustered_funs6_hc_spotname_new_cluster_2
}
for (i in 1:25) {
  if (length(levels(clustered_funs6_hc_spotname_new_cluster_2_tot[[i]][["cluster"]]))==2) {
    labels1 <- c("A","B");
    values1 <- c("#F8766D", "#00BFC4");
  }
  else if (length(levels(clustered_funs6_hc_spotname_new_cluster_2_tot[[i]][["cluster"]]))==3) {
    labels1 <- c("A","B","C");
    values1 <- c("#F8766D", "#00BFC4", "#7CAE00");
  }
  else {
    labels1 <- c("A","B","C","D");
    values1 <- c("#F8766D", "#00BFC4", "#7CAE00", "#C77CFF");
  }

  filt_fpca_hc_spotname_new_cluster_2 <- ggplot(clustered_funs6_hc_spotname_new_cluster_2_tot[[i]]) +
    geom_line(aes(time, Resistance, group = obs, col = cluster)) +
    scale_y_continuous(breaks = seq(0.05,0.35, by = 0.05), limits = c(0.05,0.35)) +
    labs(title = "DRC\n", x = "Time", y = "Resistance", color = "Sotto cluster\n") +
    scale_color_manual(labels = labels1, values = values1) +
    ggtitle(paste("Filtering FPCA hierarchical clustering", "\n",
                  "of fpca cluster 2 of spotname",spotname_new[i],"'s DRC"))
  # filt_fpca_hc_spotname_new_cluster_2
  filt_fpca_hc_cen_spotname_new_cluster_2 <- ggplot(clustered_funs6_hc_spotname_new_cluster_2_tot[[i]]) +
    geom_line(aes(time, Resistance, col = cluster)) +
    scale_y_continuous(breaks = seq(0.05,0.35, by = 0.05), limits = c(0.05,0.35)) +
    labs(title = "DRC\n", x = "Time", y = "Resistance", color = "Sotto cluster\n") +
    scale_color_manual(labels = labels1, values = values1) +
    ggtitle(paste("Centroids of filtering FPCA hierarchical clustering", "\n",
                  "of fpca cluster 2 of spotname",spotname_new[i],"'s DRC"))
  # filt_fpca_hc_cen_spotname_new_cluster_2

  full_filt_fpca__new <- ggarrange(filt_fpca_hc_spotname_new_cluster_2,
                                   filt_fpca_hc_cen_spotname_new_cluster_2,ncol=1,nrow=2)
  ggexport(full_filt_fpca__new,
           filename = paste("Filtering FPCA profili-centroidi of fpca cluster 2 of spotname",
                            spotname_new[i],"'s DRC.png"))
}
###############

#suddivido i sotto cluster in oggetti diversi
#################
scalari_cluster_2_cluster_A_fpca_hc_tot <- list()
scalari_cluster_2_cluster_B_fpca_hc_tot <- list()
scalari_cluster_2_cluster_C_fpca_hc_tot <- list()
scalari_cluster_2_cluster_D_fpca_hc_tot <- list()

for (i in 1:25) {
  profili_cluster_2_fpca_hc <- clustered_funs6_hc_spotname_new_cluster_2_tot[[i]][
    seq(from = 1, to = ncol(resistance_rawdata_spotname_fpca_hc_cluster_2_tot[[i]])),]
```

```
  profili_cluster_2_cluster_A_fpca_hc <- profili_cluster_2_fpca_hc[profili_cluster_2_fpca_hc$cluster == 1,]
  profili_cluster_2_cluster_B_fpca_hc <- profili_cluster_2_fpca_hc[profili_cluster_2_fpca_hc$cluster == 2,]
  profili_cluster_2_cluster_C_fpca_hc <- profili_cluster_2_fpca_hc[profili_cluster_2_fpca_hc$cluster == 3,]
  profili_cluster_2_cluster_D_fpca_hc <- profili_cluster_2_fpca_hc[profili_cluster_2_fpca_hc$cluster == 4,]

  #cambio nome a colonna delle obs e di cluster
  colnames(profili_cluster_2_cluster_A_fpca_hc)[2] <- "dateTime"
  colnames(profili_cluster_2_cluster_A_fpca_hc)[4] <- "cluster fpca hc A"

  colnames(profili_cluster_2_cluster_B_fpca_hc)[2] <- "dateTime"
  colnames(profili_cluster_2_cluster_B_fpca_hc)[4] <- "cluster fpca hc B"

  colnames(profili_cluster_2_cluster_C_fpca_hc)[2] <- "dateTime"
  colnames(profili_cluster_2_cluster_C_fpca_hc)[4] <- "cluster fpca hc C"

  colnames(profili_cluster_2_cluster_D_fpca_hc)[2] <- "dateTime"
  colnames(profili_cluster_2_cluster_D_fpca_hc)[4] <- "cluster fpca hc D"

  #unisco colonna relativa al cluster di appartenenza
  scalari_cluster_2_cluster_A_fpca_hc <-
    merge(scalari_associazione_new,profili_cluster_2_cluster_A_fpca_hc,by = ("dateTime"))
  scalari_cluster_2_cluster_B_fpca_hc <-
    merge(scalari_associazione_new,profili_cluster_2_cluster_B_fpca_hc,by = ("dateTime"))
  scalari_cluster_2_cluster_C_fpca_hc <-
    merge(scalari_associazione_new,profili_cluster_2_cluster_C_fpca_hc,by = ("dateTime"))
  scalari_cluster_2_cluster_D_fpca_hc <-
    merge(scalari_associazione_new,profili_cluster_2_cluster_D_fpca_hc,by = ("dateTime"))

  scalari_cluster_2_cluster_A_fpca_hc_tot[[i]] <- scalari_cluster_2_cluster_A_fpca_hc
  scalari_cluster_2_cluster_B_fpca_hc_tot[[i]] <- scalari_cluster_2_cluster_B_fpca_hc
  scalari_cluster_2_cluster_C_fpca_hc_tot[[i]] <- scalari_cluster_2_cluster_C_fpca_hc
  scalari_cluster_2_cluster_D_fpca_hc_tot[[i]] <- scalari_cluster_2_cluster_D_fpca_hc
}
##################

#creo matrici resistance rawdata
##################
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot <- list()
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot <- list()
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot <- list()
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot <- list()

for (i in 1:25) {
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A <- data_profili_new %>%
    filter(id %in% scalari_cluster_2_cluster_A_fpca_hc_tot[[i]]$dateTime ) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A<-
    t(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A)

  colnames(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A)<-
    resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A[1,]
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A<-
    resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A[
    is.na(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A)] <- 0

  storage.mode(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A) <- "numeric"
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[i]] <-
    resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A
  names(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot)[[i]] <- spotname_new[i]


  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B <- data_profili_new %>%
    filter(id %in% scalari_cluster_2_cluster_B_fpca_hc_tot[[i]]$dateTime) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B<-
    t(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B)

  colnames(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B)<-
    resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B[1,]
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B<-
    resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B[
    is.na(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B)] <- 0

  storage.mode(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B) <- "numeric"
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[i]] <-
    resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B
  names(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot)[[i]] <- spotname_new[i]

  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C <- data_profili_new %>%
```

```
    filter(id %in% scalari_cluster_2_cluster_C_fpca_hc_tot[[i]]$dateTime ) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C<-
    t(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C)

  colnames(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C)<-
    resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C[1,]
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C<-
    resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C[
    is.na(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C)] <- 0

  storage.mode(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C) <- "numeric"
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[i]] <-
    resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C
  names(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot)[[i]] <- spotname_new[i]


  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D <- data_profili_new %>%
    filter(id %in% scalari_cluster_2_cluster_D_fpca_hc_tot[[i]]$dateTime) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D<-
    t(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D)

  colnames(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D)<-
    resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D[1,]
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D<-
    resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D[
    is.na(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D)] <- 0

  storage.mode(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D) <- "numeric"
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[i]] <-
    resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D
  names(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot)[[i]] <- spotname_new[i]

}
##################

#taglio durata profili come all'inizio (240,280,300,320,340 ms)
####################
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[1]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[1]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[2]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[2]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[3]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[3]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[4]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[4]][1:280,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[5]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[5]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[6]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[6]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[7]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[7]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[8]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[8]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[9]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[9]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[10]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[10]][1:220,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[11]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[11]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[12]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[12]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[13]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[13]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[14]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[14]][1:340,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[15]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[15]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[16]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[16]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[17]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[17]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[18]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[18]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[19]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[19]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[20]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[20]][1:240,]
```

```
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[21]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[21]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[22]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[22]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[23]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[23]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[24]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[24]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[25]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[25]][1:320,]

resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[1]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[1]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[2]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[2]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[3]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[3]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[4]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[4]][1:280,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[5]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[5]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[6]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[6]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[7]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[7]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[8]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[8]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[9]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[9]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[10]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[10]][1:220,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[11]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[11]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[12]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[12]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[13]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[13]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[14]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[14]][1:340,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[15]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[15]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[16]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[16]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[17]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[17]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[18]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[18]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[19]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[19]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[20]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[20]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[21]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[21]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[22]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[22]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[23]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[23]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[24]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[24]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[25]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[25]][1:320,]

resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[1]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[1]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[2]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[2]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[3]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[3]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[4]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[4]][1:280,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[5]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[5]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[6]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[6]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[7]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[7]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[8]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[8]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[9]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[9]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[10]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[10]][1:220,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[11]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[11]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[12]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[12]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[13]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[13]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[14]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[14]][1:340,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[15]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[15]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[16]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[16]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[17]]<-
```

```
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[17]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[18]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[18]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[19]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[19]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[20]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[20]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[21]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[21]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[22]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[22]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[23]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[23]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[24]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[24]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[25]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[25]][1:320,]

resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[1]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[1]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[2]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[2]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[3]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[3]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[4]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[4]][1:280,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[5]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[5]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[6]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[6]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[7]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[7]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[8]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[8]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[9]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[9]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[10]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[10]][1:220,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[11]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[11]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[12]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[12]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[13]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[13]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[14]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[14]][1:340,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[15]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[15]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[16]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[16]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[17]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[17]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[18]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[18]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[19]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[19]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[20]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[20]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[21]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[21]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[22]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[22]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[23]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[23]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[24]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[24]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[25]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[25]][1:320,]
####################

#Tabelle di contingenza
####################
tab_conting_filt_fpca_cluster_2 <- matrix(nrow = length(spotname_new), ncol = 4+1)

row.names(tab_conting_filt_fpca_cluster_2)<- spotname_new
colnames(tab_conting_filt_fpca_cluster_2)<- c("A","B","C","D","Totale")

#riempio tabella
for (i in 1:25) {
  tab_conting_filt_fpca_cluster_2[i,] <- c(ncol(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[i]]),
                                           ncol(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[i]]),
                                           ncol(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[i]]),
                                           ncol(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[i]]),
                                           (ncol(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_A_tot[[i]])+
                                              ncol(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_B_tot[[i]])+
                                              ncol(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_C_tot[[i]])+
                                              ncol(resistance_rawdata_spotname_fpca_hc_cluster_2_cluster_D_tot[[i]])))
}

#creo tabelle con frequenze
tab_conting_filt_fpca_cluster_2_freq_intraspotname <-
  prop.table(tab_conting_filt_fpca_cluster_2[,-5],margin = 1)

write.csv2(tab_conting_filt_fpca_cluster_2, file="tabella contingenza filtering fpca cluster 2.csv",
           quote=F, na="", row.names=T, col.names=T)
```

```r
write.csv2(tab_conting_filt_fpca_cluster_2_freq_intraspotname,
           file="tabella contingenza filtering fpca cluster 2 frequenza intraspotname.csv",
           quote=F, na="", row.names=T, col.names=T)
###################


#cluster 3

#clustering
###############
clustered_funs6_hc_spotname_new_cluster_3_tot <- list()

for (i in c(1:9,12,14,18:19,21,23,25)) {
  mod6_spotname_new_cluster_3 <- fil_fpca_ss_nbclust(data = data_spotname_fpca_cluster_3_tot[[i]],
                                                     num_cluster_seq = 2:4, per_comp = 0.8)
  mod6_spotname_new_cluster_3$mod_opt$ind_hc$nbclust ## hierarchical

  cl_6_hc_spotname_new_cluster_3 <- mod6_spotname_new_cluster_3$mod_opt$ind_hc$cluster

  clustered_funs6_hc_spotname_new_cluster_3 <-
    get_clustered_funs_df(cl = cl_6_hc_spotname_new_cluster_3,
                          data = data_spotname_fpca_cluster_3_tot[[i]],
                          method = "filtering FPCA\nhierarchical")
  centroids6_hc_spotname_new_cluster_3 <-
    get_centroids_df(cl_6_hc_spotname_new_cluster_3, data_spotname_fpca_cluster_3_tot[[i]],
                     "filtering FPCA\nhierarchical")

  clustered_funs6_hc_spotname_new_cluster_3_tot[[i]] <- clustered_funs6_hc_spotname_new_cluster_3
}
for (i in c(1:9,12,14,18:19,21,23,25)) {
  if (length(levels(clustered_funs6_hc_spotname_new_cluster_3_tot[[i]][["cluster"]]))==2) {
    labels1 <- c("A","B");
    values1 <- c("#F8766D", "#00BFC4");
  }
  else if (length(levels(clustered_funs6_hc_spotname_new_cluster_3_tot[[i]][["cluster"]]))==3) {
    labels1 <- c("A","B","C");
    values1 <- c("#F8766D", "#00BFC4", "#7CAE00");
  }
  else {
    labels1 <- c("A","B","C","D");
    values1 <- c("#F8766D", "#00BFC4", "#7CAE00", "#C77CFF");
  }

  filt_fpca_hc_spotname_new_cluster_3 <- ggplot(clustered_funs6_hc_spotname_new_cluster_3_tot[[i]]) +
    geom_line(aes(time, Resistance, group = obs, col = cluster)) +
    scale_y_continuous(breaks = seq(0.05,0.35, by = 0.05), limits = c(0.05,0.35)) +
    labs(title = "DRC\n", x = "Time", y = "Resistance", color = "Sotto cluster\n") +
    scale_color_manual(labels = labels1, values = values1) +
    ggtitle(paste("Filtering FPCA hierarchical clustering", "\n",
                  "of fpca cluster 3 of spotname",spotname_new[i],"'s DRC"))
  # filt_fpca_hc_spotname_new_cluster_3
  filt_fpca_hc_cen_spotname_new_cluster_3 <- ggplot(clustered_funs6_hc_spotname_new_cluster_3_tot[[i]]) +
    geom_line(aes(time, Resistance, col = cluster)) +
    scale_y_continuous(breaks = seq(0.05,0.35, by = 0.05), limits = c(0.05,0.35)) +
    labs(title = "DRC\n", x = "Time", y = "Resistance", color = "Sotto cluster\n") +
    scale_color_manual(labels = labels1, values = values1) +
    ggtitle(paste("Centroids of filtering FPCA hierarchical clustering", "\n",
                  "of fpca cluster 3 of spotname",spotname_new[i],"'s DRC"))
  # filt_fpca_hc_cen_spotname_new_cluster_3
  full_filt_fpca__new <- ggarrange(filt_fpca_hc_spotname_new_cluster_3,
                                   filt_fpca_hc_cen_spotname_new_cluster_3,ncol=1,nrow=2)
  ggexport(full_filt_fpca__new,
           filename = paste("Filtering FPCA profili-centroidi of fpca cluster 3 of spotname",
                            spotname_new[i],"'s DRC.png"))
}
###############

#suddivido i sotto cluster in oggetti diversi
#################
scalari_cluster_3_cluster_A_fpca_hc_tot <- list()
scalari_cluster_3_cluster_B_fpca_hc_tot <- list()
scalari_cluster_3_cluster_C_fpca_hc_tot <- list()
scalari_cluster_3_cluster_D_fpca_hc_tot <- list()

for (i in c(1:9,12,14,18:19,21,23,25)) {
  profili_cluster_3_fpca_hc <- clustered_funs6_hc_spotname_new_cluster_3_tot[[i]][
    seq(from = 1, to = ncol(resistance_rawdata_spotname_fpca_hc_cluster_3_tot[[i]])),]
  profili_cluster_3_cluster_A_fpca_hc <- profili_cluster_3_fpca_hc[profili_cluster_3_fpca_hc$cluster == 1,]
  profili_cluster_3_cluster_B_fpca_hc <- profili_cluster_3_fpca_hc[profili_cluster_3_fpca_hc$cluster == 2,]
  profili_cluster_3_cluster_C_fpca_hc <- profili_cluster_3_fpca_hc[profili_cluster_3_fpca_hc$cluster == 3,]
  profili_cluster_3_cluster_D_fpca_hc <- profili_cluster_3_fpca_hc[profili_cluster_3_fpca_hc$cluster == 4,]

  #cambio nome a colonna delle obs e di cluster
  colnames(profili_cluster_3_cluster_A_fpca_hc)[2] <- "dateTime"
  colnames(profili_cluster_3_cluster_A_fpca_hc)[4] <- "cluster fpca hc A"

  colnames(profili_cluster_3_cluster_B_fpca_hc)[2] <- "dateTime"
  colnames(profili_cluster_3_cluster_B_fpca_hc)[4] <- "cluster fpca hc B"

  colnames(profili_cluster_3_cluster_C_fpca_hc)[2] <- "dateTime"
  colnames(profili_cluster_3_cluster_C_fpca_hc)[4] <- "cluster fpca hc C"

  colnames(profili_cluster_3_cluster_D_fpca_hc)[2] <- "dateTime"
  colnames(profili_cluster_3_cluster_D_fpca_hc)[4] <- "cluster fpca hc D"
```

```
  #unisco colonna relativa al cluster di appartenenza
  scalari_cluster_3_cluster_A_fpca_hc <-
    merge(scalari_associazione_new,profili_cluster_3_cluster_A_fpca_hc,by = ("dateTime"))
  scalari_cluster_3_cluster_B_fpca_hc <-
    merge(scalari_associazione_new,profili_cluster_3_cluster_B_fpca_hc,by = ("dateTime"))
  scalari_cluster_3_cluster_C_fpca_hc <-
    merge(scalari_associazione_new,profili_cluster_3_cluster_C_fpca_hc,by = ("dateTime"))
  scalari_cluster_3_cluster_D_fpca_hc <-
    merge(scalari_associazione_new,profili_cluster_3_cluster_D_fpca_hc,by = ("dateTime"))

  scalari_cluster_3_cluster_A_fpca_hc_tot[[i]] <- scalari_cluster_3_cluster_A_fpca_hc
  scalari_cluster_3_cluster_B_fpca_hc_tot[[i]] <- scalari_cluster_3_cluster_B_fpca_hc
  scalari_cluster_3_cluster_C_fpca_hc_tot[[i]] <- scalari_cluster_3_cluster_C_fpca_hc
  scalari_cluster_3_cluster_D_fpca_hc_tot[[i]] <- scalari_cluster_3_cluster_D_fpca_hc
}
##################
```

```
#creo matrici resistance rawdata
##################
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot <- list()
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot <- list()
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot <- list()
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot <- list()

for (i in c(1:9,12,14,18:19,21,23,25)) {
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A <- data_profili_new %>%
    filter(id %in% scalari_cluster_3_cluster_A_fpca_hc_tot[[i]]$dateTime ) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A<-
    t(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A)

  colnames(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A)<-
    resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A[1,]
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A<-
    resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A[
    is.na(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A)] <- 0

  storage.mode(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A) <- "numeric"
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[i]] <-
    resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A
  names(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot)[[i]] <- spotname_new[i]


  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B <- data_profili_new %>%
    filter(id %in% scalari_cluster_3_cluster_B_fpca_hc_tot[[i]]$dateTime) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B<-
    t(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B)

  colnames(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B)<-
    resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B[1,]
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B<-
    resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B[
    is.na(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B)] <- 0

  storage.mode(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B) <- "numeric"
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[i]] <-
    resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B
  names(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot)[[i]] <- spotname_new[i]

  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C <- data_profili_new %>%
    filter(id %in% scalari_cluster_3_cluster_C_fpca_hc_tot[[i]]$dateTime ) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C<-
    t(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C)

  colnames(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C)<-
    resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C[1,]
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C<-
    resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C[
    is.na(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C)] <- 0
```

```
  storage.mode(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C) <- "numeric"
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[i]] <-
    resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C
  names(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot)[[i]] <- spotname_new[i]


  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D <- data_profili_new %>%
    filter(id %in% scalari_cluster_3_cluster_D_fpca_hc_tot[[i]]$dateTime) %>%
    pivot_wider(names_from = time,
                values_from = ResistanceCurve,
                id_cols = id,
                values_fill = NA)

  #Faccio la trasposta per avere sulle righe i millisecondi e sulle colonne i "profili"
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D<-
    t(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D)

  colnames(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D)<-
    resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D[1,]
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D<-
    resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D[-1,]

  #Sostituisco i valori NA con 0
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D[
    is.na(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D)] <- 0

  storage.mode(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D) <- "numeric"
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[i]] <-
    resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D
  names(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot)[[i]] <- spotname_new[i]

}
#################

#taglio durata profili come all'inizio (240,280,300,320,340 ms)
####################
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[1]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[1]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[2]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[2]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[3]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[3]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[4]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[4]][1:280,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[5]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[5]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[6]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[6]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[7]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[7]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[8]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[8]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[9]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[9]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[10]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[10]][1:220,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[11]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[11]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[12]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[12]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[13]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[13]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[14]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[14]][1:340,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[15]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[15]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[16]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[16]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[17]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[17]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[18]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[18]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[19]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[19]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[20]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[20]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[21]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[21]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[22]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[22]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[23]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[23]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[24]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[24]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[25]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[25]][1:320,]

resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[1]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[1]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[2]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[2]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[3]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[3]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[4]]<-
```

```
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[4]][1:280,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[5]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[5]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[6]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[6]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[7]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[7]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[8]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[8]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[9]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[9]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[10]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[10]][1:220,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[11]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[11]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[12]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[12]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[13]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[13]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[14]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[14]][1:340,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[15]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[15]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[16]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[16]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[17]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[17]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[18]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[18]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[19]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[19]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[20]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[20]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[21]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[21]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[22]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[22]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[23]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[23]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[24]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[24]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[25]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[25]][1:320,]

resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[1]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[1]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[2]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[2]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[3]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[3]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[4]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[4]][1:280,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[5]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[5]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[6]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[6]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[7]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[7]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[8]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[8]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[9]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[9]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[10]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[10]][1:220,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[11]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[11]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[12]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[12]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[13]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[13]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[14]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[14]][1:340,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[15]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[15]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[16]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[16]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[17]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[17]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[18]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[18]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[19]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[19]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[20]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[20]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[21]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[21]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[22]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[22]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[23]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[23]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[24]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[24]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[25]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[25]][1:320,]
```

```
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[1]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[1]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[2]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[2]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[3]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[3]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[4]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[4]][1:280,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[5]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[5]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[6]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[6]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[7]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[7]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[8]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[8]][1:300,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[9]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[9]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[10]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[10]][1:220,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[11]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[11]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[12]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[12]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[13]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[13]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[14]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[14]][1:340,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[15]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[15]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[16]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[16]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[17]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[17]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[18]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[18]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[19]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[19]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[20]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[20]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[21]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[21]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[22]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[22]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[23]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[23]][1:240,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[24]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[24]][1:320,]
resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[25]]<-
  resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[25]][1:320,]
####################

#Tabelle di contingenza
###################
tab_conting_filt_fpca_cluster_3 <- matrix(nrow = length(spotname_new), ncol = 4+1)

row.names(tab_conting_filt_fpca_cluster_3)<- spotname_new
colnames(tab_conting_filt_fpca_cluster_3)<- c("A","B","C","D","Totale")

#visto che filt fpca non riesce con spotname 53702 e 53703, levo ultime due righe
tab_conting_filt_fpca_cluster_3 <- tab_conting_filt_fpca_cluster_3[-c(18:25),]

#riempio tabella
for (i in c(1:9,12,14,16,18,19,21,23,25)) {
  tab_conting_filt_fpca_cluster_3[i,] <- c(ncol(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[i]]),
                                           ncol(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[i]]),
                                           ncol(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[i]]),
                                           ncol(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[i]]),
                                           (ncol(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_A_tot[[i]])+
                                              ncol(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_B_tot[[i]])+
                                              ncol(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_C_tot[[i]])+
                                              ncol(resistance_rawdata_spotname_fpca_hc_cluster_3_cluster_D_tot[[i]])))
}

#creo tabelle con frequenze
tab_conting_filt_fpca_cluster_3_freq_intraspotname <-
  prop.table(tab_conting_filt_fpca_cluster_3[,-5],margin = 1)

write.csv2(tab_conting_filt_fpca_cluster_3, file="tabella contingenza filtering fpca cluster 3.csv",
           quote=F, na="", row.names=T, col.names=T)
write.csv2(tab_conting_filt_fpca_cluster_3_freq_intraspotname,
           file="tabella contingenza filtering fpca cluster 3 frequenza intraspotname.csv",
           quote=F, na="", row.names=T, col.names=T)
####################
```