Università degli studi di Padova

Facoltà di Scienze Statistiche

Corso di Laurea Specialistica in Scienze Statistiche,
Demografiche e Sociali

Tesi di Laurea

Innovative statistical methods for the construction of composite scales from ordinal variables with application to business and management data

September 15, 2009

Relatore: Ch.mo Prof. Fortunato Pesarin

Correlatore: Ch.mo Prof. Luigi Salmaso

Laureando: Enrico Tonini - 600413

Anno Accademico 2008-09

To my family

Contents

List of Figures

List of Tables

Preface

This work will deal with two important aspects of statistics very useful in particular in management engineering: the use of statistical instruments in management research and the investigation of some methods useful to make rankings of treatments (for example different products).

Quite often in management literature, in the framework of research that uses Likert's scales, some statistical instruments are used because only most of all other people do it this way, even writers of articles published in important scientific magazines, but probably these statistical instruments are not always methodologically correct and the results could be completely wrong; in the first chapter we will present this traditional analysis and we will propose a method to do it in another way.

The second and the third chapters, instead, will deal with some methods useful to order treatments: this helps for example to evaluate different (new) products.

Chapter 1

Analysis of Likert's scales

In this chapter we will eventually discuss two ways of analyzing data which are constructed on Likert's scales: first we will present a more traditional (but probably not correct) analysis and then we will introduce a new (and probably more correct) analysis. But first of all let us see in a better way types of existing variables and their features, in particular variables based on Likert's scale.

1.1 Introduction: a brief summary of types of variables

In order to know how to analyze data based on Likert's scale, it is very useful to do a brief classification of statistical variables' types[1], so we can also better frame our type of data.
We can distinguish substantially these types of variables:

Quantitative variables

Their modalities are numeric and they in turn are distinguished in two ways:

[1] The reader can find it (probably a better one) in all statistics books, in particular we have used [17].

ratio scales vs. interval scales the "zero" of ratio scales is not conventional, but fixed (for example: weight, height, length, etc.), whereas the "zero" of interval scales is merely conventional (for example: temperature). To compare modalities of ratio scales we can use every operation, even the ratio itself[2], in fact we can always say, for example, that two meters is twice as long as one, as we can correctly say that two yards is twice as long as one yard; to compare modalities of interval scales, instead, we cannot use ratio, but we have to stop to the difference, in fact we can say that from 15°C and 30°C there are 15°C, but we cannot absolutely say that 30°C is twice as hot as 15°C, in fact instead of using Celsius scale, if we used Kelvin scale, we would have, instead of 15°C and 30°C, 288.15K and 303.15K and the second is not twice the first, but the difference is the same: 15K, like 15°C.

continuous scales vs. discrete scales in continuous scales modalities of variables can assume all values in one or more real intervals, whereas discrete variables can assume only a few values or a countable infinity of values.

Qualitative variables

Their modalities are not numeric and there are two types of them:

ordinal scales their modalities have a certain order and no numeric operation is possible, we can only order modalities (example: tha variable which describes the mark of some schools: its modalities can be insufficient, sufficient, discrete, good, perfect).

nominal scales their modalities have no order and the only operation possible between them is comparison (example: the variable "faculty of the University of Padua" has thirteen modalities which we can only compare).

[2] It is not strange indeed that the name of this type of variable is "ratio" variable...

Dichotomic variables

They have only two modalities (examples: sex, presence or absence of a particular feature in something, etc.); depending on the situation sometimes are considered quantitative variables, sometimes qualitative variables, therefore we prefer to consider them apart.

1.2 Likert's scales: the peculiarities of this type of data

1.2.1 Features and construction of a Likert's scale

In this chapter we are analyzing a particular type of data that we normally call *Likert's scales*. They seem to be very useful to try to measure attitudes of people. This particular scale was proposed by Rensis Likert (1903-1981), an american educator and organizational psychologist, in his primary work[3]. Likert's scales, as already written, are used to measure people attitudes with respect to an abstract concept and every concept is measured with a series of questions, generally called *items*; every item is normally coded in 5 or 7 levels: "1" indicates the lowest level of attitude, "5" (or "7") indicates the highest level of attitude. For further information on Likert's scales, see [15], [16], [48], [49], [12] and [30].

1.2.2 The literary "disputation" between methodologic statisticians and more pragmatic economists

This type of data is particularly problematic: how can we consider them? From a methodologic-statistical point of view they are merely qualitative ordinal variables, in fact, for example, we know that for a Likert's scale "6" *is more than* "3", so we can at least extablish an ordering between modalities: $7 \succ 6 \succ 5 \succ 4 \succ 3 \succ 2 \succ 1$, where "$\succ$" indicates *is better than*.

[3] See [31].

Instead very often, above all in economical and management literature, they treat this type of data as quantitative discrete variables and in this way they give a numeric significance to the modalities. But this is not correct from a statistical point of view, because the modalities have no numeric significance: for example if an informant answers “4” to an item and another informant answers “2” it does not mean that the first response is twice as satisfied as the second, but only that the first informant is more (or less, if the sense of the question is inverted) satisfied than the second. For further information on this topic, see [43], [44], [28] and, in particular, [25], that is a brief but very interesting paper.

Considering these variables as quantitative discrete, as the main part of management literature does, we can apply to this type of data lots of statistical analysis: mean is good to synthesize the answers, standard deviation is good to synthesize the variability of the answers, all parametric tests can be applied to the data, etc. Instead, if we consider the items of Likert's scale merely qualitative ordinal, as they are indeed, we should change the precedent analysis: median (or mode) instead of mean, interquartile difference (or also other variability indexes) instead of the standard deviation, lots of parametric tests cannot be used any more and must be supplied by non-parametric tests.

Likert himself called the scales *summating scales*, because he said that as the scales respect some assumptions[4], scores on items of the same scale can be correctly summed up. But the problem is that it is almost impossible to verify all assumptions of the scales:

1. unidimensionality: every declared scale (every group of items that are underlining a same concept to investigate) should be subjected to a same concept that should represent only one dimension;

2. equidistance of intervals: the modalities of the items should be equidistant; so, for example, between “3” and “4” there should be the same

[4] See [8].

difference than the one between "6" and "7";

3. validity: every scale should really measure what it is declared to measure;

4. reliability: every scale should have the capability to limit random errors, that is if we repeated the analysis, the new results should be not significantly different from the first ones[5].

Furthermore it is desirable that the items of the scales have another property: they should have an approximated normal (gaussian) shape, or at least symmetric. Items that are too skew or too concentrated cannot be used in the analyses because they are not useful: it would mean that such an item is unuseful to discriminate within the population of interest. In order to check this property it can be useful to perform histograms and boxplots of the answers and to calculate skewness and curtosis indexes: it is adviced to keep only items that have skewness and curtosis indexes between -1 and 1. Even if in literature some statistical instruments have been proposed to assess unidimensionality, validity and reliability of a Likert's scale, we have no way to check the equidistance of intervals; furthermore the instruments to assess previous assumptions are not always correct from a methodological-statistical point of view[6].

So, remembering these considerations and also that Rensis Likert was not a statistician, but a psychologist, we may have some perplexities about how we can treat this particular type of data.

My proposal to the reader for this work and what we were going to do in the next two sections is to consider a real dataset and to perform on these data

[5] Notice that validity and reliability are completely different even if it is very easy to confound them: for example a scale could really measure what it is declared to measure but could also have great random errors and in this case the scale is valid but not reliable; it can also occur that a scale has very small errors, but it does not measure at all what we want it to measure and in this case the scale is reliable, but not valid.

[6] See sections 1.3 and 1.4.

the two types of analysis we are going to present: the first one is the traditional and more simple analysis that the main part of management literature does[7] and that we will see not to be methodologically correct in almost all its steps; the second is a new, in a main part innovative, maybe a little more complicated, but surely methodologically correct analysis that we want to propose to the reader.

In the end, it would also be good to compare the results of the two analyses seeing if, at least in this case, results of the first (very used but not correct) analysis are similar to the results of the second (not used but correct) analysis. We must say that we are only proposing an alternative possible analysis to the traditional one and that we have absolutely no claim to demonstrate anything, but we would like to see how this second alternative analysis could work on real sets of data.[8]

1.3 A first more traditional and pragmatic analysis

In this section we will discuss the first analysis. Reading and analyzing lots of management papers (in particular [38] and also schemes of analysis of the scales quoted in [7]), we have identified the following steps in order to perform an analysis based on Likert's scales:

1. aggregation of multiple informants in the same analysis unit;
2. check of the first aggregation;
3. aggregation of multiple items in a unique scale;

[7] We have used in particular the outline of the paper [38], but we found this sequence of operations in lots of management papers and you can also do it.

[8] We wanted to perform also the two analyses and the comparison between them in this work, but then we chose not to insert it, because it was too long and time was not enough; we will probably do it in other works. Furthermore a simulation approach, as the one followed in chapters 2 and 3 would be even better than simply comparing the two analyses applied on a single dataset, but this requires really a lot of time.

4. check of reliability;
5. check of validity and unidimensionality;
6. evaluation of presence of multicollinearity;
7. final analysis.

In the paragraphs of this section we will see all previous steps with the first method (the traditional, but not correct one), in the paragraphs of the following section, instead, we will do the same with the second method (the unused, but innovative and correct one).

1.3.1 Aggregation of multiple informants in the same analysis unit

Before analyzing data we very often have to do a preliminar operation; in fact in management research normally the unit of analysis is the firm, but for every firm there are more informants: so items are referred to the firms, but we have more than one answer for every analysis unit and, afterall, not all the items have the same informants; so, the beginning datasets in this particular situation are presented in a way like the one you can see in table 1.1[9], where u_i indicates the unit of analysis (for example the firm), r_{ij} indicates the j-th informant of the i-th unit of analysis, s_k indicates the k-th scale and i_{kl} indicates the l-th item of the k-th scale.

The first step is therefore to aggregate the answers of different informants in a unique answer for the units of analysis in order to have one answer to every item for every unit (firm); look at the example in table 1.2[10].

In this first analysis we advise you to do it with arithmetic mean of the answers; for more details about how to perform this aggregation with different types of mean, see [45].

[9] In this example items are based only on 5 and not on 7 modalities.

[10] In the example we have used arithmetic mean in order to perform this aggregation.

		s_1			s_2		s_3				
		i_{1a}	i_{1b}	i_{1c}	i_{2a}	i_{2b}	i_{3a}	i_{3b}	i_{3c}	i_{3d}	i_{3e}
u_1	r_{11}	3	4	3	1	1					
u_1	r_{12}				1	3	3	3	4	3	2
u_1	r_{13}	1	3	5	4	2					
u_1	r_{14}				2	1	4	4	3	1	4
u_2	r_{21}	3	5	5			5	5	5	5	5
u_2	r_{22}				5	5	4	3	4	3	3
u_2	r_{23}				3	4					
u_3	r_{31}	3	4	5			4	3	1	3	2
u_3	r_{32}	4	4	5	3	2					
u_3	r_{33}	5	5	4	1	4					
u_3	r_{34}	3	3	4	4	4	3	4	2	5	4
u_3	r_{35}						4	4	4	3	5
u_3	r_{36}				2	3					
u_4	r_{41}	5	5	5	3	5					
u_4	r_{42}	4	3	5	4	4	2	2	3	4	1
u_5	r_{51}	1	2	4	2	1	5	5	5	3	4

Table 1.1: Example of how data based on Likert's scales can appear.

	s_1			s_2		s_3				
	i_{1a}	i_{1b}	i_{1c}	i_{2a}	i_{2b}	i_{3a}	i_{3b}	i_{3c}	i_{3d}	i_{3e}
u_1	2.50	3.50	4.00	2.00	1.75	3.00	3.00	4.00	3.00	2.00
u_2	3.00	5.00	5.00	4.00	4.50	4.50	4.00	4.50	4.00	4.00
u_3	3.75	4.00	4.50	2.50	3.25	3.67	3.67	2.33	3.67	3.67
u_4	4.50	4.00	5.00	3.50	4.50	2.00	2.00	3.00	4.00	1.00
u_5	1.00	2.00	4.00	2.00	1.00	5.00	5.00	5.00	3.00	4.00

Table 1.2: Example of aggregation of more informants on the same unit of analysis.

1.3.2 Missing data and exploration of distributions of answers

We suggest to deal with missing data only after the first aggregation, because it would be too onerous to deal with them on individual data[11]; instead, after

[11] Notice that at the individual level we consider missing data only cases when an informant that should answer does not answer; obviously data on items where an informant

the first aggregation, we have missing data only where no informant in a firm for a certain item has answered: in this way we have not so many missing data and it is very much simpler to deal with them[12]...
As we have already said in the previous section, they say that distributions of answers should be at least symmetric: we advise you to perform a histogram, a boxplot and the calculation of skewness and curtosis indexes for every item.

1.3.3 Assessing the first aggregation: some indicators

After aggregating multiple answers in a unique answer for every unit of analysis, we should check if this first aggregation was possible, that is if the *agreement* between informants of the same firm is high or not. If answers of informants of the same firm are similar, we are more confident that we have not lost a great piece of information. We can check the previous aggregation by evaluating the values of some indexes[13]: F test, Interclass Correlation (ICC), η^2 and the different versions of Inter-Rater Agreement (IRA).
In literature there are some criteria[14] that tell us how good is the agreement among informants of the same unit of analysis:

- η^2 should be greater than 0.16 (others say greater than 0.20);
- F-value should be greater than 1.00 (or, better, relative p-value should be statistically significant, traditionally less than 0.05);
- ICC should be greater than 0.60;
- IRA should be greater than 0.80.

must not answer are not missing data (for example the empty cells of table 1.1 at page 10).

[12] An interesting paper where you can find lots of choices, also more sophisticated than ours, in order to solve the problem of missing data is [41]; you can also have a look at [17].

[13] Some relevant papers where these indexes are calculated are [34], [38], [32], [11], [20], [27] and [10]; the last two ones, in particular, are very useful to learn something more about different types of IRA.

[14] see, for example, [34] and [40].

1.3.4 Some used methods to assess reliability of scales

To evaluate the reliability of scales to measure attitudes, management literature proposes more statistical instruments.

Probably the best method to assess reliability of a scale is the *test-retest* tecnique: in order to see how much the analysis is repeatable, we should repeat it and compare the results, seeing (for example with a paired t-test) if the differences are significative. Obviously if we want to perform this analysis on a given dataset, we cannot perform this technique, therefore we are using other methods.

First of all the main part of this literature calculates for every scale (for every group of items) the coefficient *Cronbach's* α[15], which formula is:

$$\alpha = \frac{n\overline{r}}{1+\overline{r}(n-1)} \tag{1.1}$$

where $\overline{r}$ is the average Pearson's correlation coefficient between the items and the scale and n is the number of the items of the scale. Some researches say that α should be over 0.7, therefore if for a certain scale it is not so, we should exclude from the analysis items that are not so much correlated with the scale, in order to increase the value of α.

In the final analysis it is better using the scales and not the single items, because we should lose reliability in order to capture the concept.

1.3.5 Some used methods to assess validity and unidimensionality of scales

To check if all scales capture only one dimension each, in management literature very often an Explorative Factor Analysis (EFA) is performed[16] for every scale, checking if every scale loads on only one factor. To assess validity, instead, in management literature we find that an explorative factor

[15] For further details on how using it, see [21].

[16] Sometimes also a Confirmative Factor Analysis (CFA) is performed, but we will not discuss it in this context.

analysis is still performed, but this time with all items of all scales and we must check if every scale loads on a different dimension.

1.3.6 Aggregation of multiple items in a unique scale

At this point researchers normally want to deal with some variables that can represent some identified concepts; essentially at this point we need to join the items (only the ones kept from previous analyses) in scales in order to have one variable (scale) for every concept; look at the example below[17]:

	s_1	s_2	s_3
u_1	3.17	1.88	3.00
u_2	4.33	4.25	4.20
u_3	4.08	2.88	3.40
u_4	4.50	4.00	2.40
u_5	2.33	1.50	4.40

Table 1.3: Example of aggregation of more items on the same scale.

One could question that we could deal also with single items, but the problem is that very often a single item cannot catch one whole concept, because usually a concept is something too much complex; so, if we used single items in the analyses we want to perform, probably we would not have consistent[18] variables and also we would have too many of them.

In most of management papers this second aggregation is performed with sum (in fact, as we have already said, Likert himself referred to these data as summating scales) or arithmetic mean; in this work we have chosen to use arithmetic mean, because the scales of our dataset have not the same

[17] In order to perform this second aggregation on the example of table 1.2 at page 10, we suggest to use arithmetic mean.

[18] We are not using this word in a strict statistical sense, but for us it only means that single items probably do not represent whole abstract concepts.

number of items and so, using mean and not sum, all scales have a range in the interval $[1, 7]$.

1.3.7 Assessing presence of multicollinearity in indipendent variables

It can be very useful to check if the indipendent variables (scales, not single items anymore) of our future model are indipendent, because in this way every variable explains an original part of the variability of the response variable. In management literature this evaluation is performed by a simple table of Pearson's correlation coefficients among indipendent variables; if two variables are very correlated, in the future analysis we will use only one of them.

1.3.8 Final analysis

Very often we are interested in performing a regression, in order to see if and how much a response variable is influenced by some explicative variables. In management literature almost always a linear regression is performed; sometimes also an analysis of variance is performed, because it is often useful to know if some groups have a significantly different behaviour or not.

1.4 A second less traditional but more methodologically correct analysis

1.4.1 Aggregation of multiple informants in the same analysis unit

Considering items merely ordinal variables as they indeed are, we cannot apply to the answers the mean anymore; in this second analysis we advise to aggregate multiple answers in a unique one using median; other possible choices are using mode or keeping only one answer (maybe the answer of the informant that we think to be, for some motivation, the most reliable[19]), but each of these choices has its pros and cons. Probably it would be interesting to investigate better this point of the analysis in order to use a method that can keep all information we have.

1.4.2 Assessing the first aggregation: some indicators

Probably only some versions of IRA indexes are methodologically correct.

1.4.3 Explorating distributions of answers

Skewness and curtosis indexes and boxplot are not correct from a statistical point of view, because they are applicable only to quantitative variables and not to qualitative ones; the histogram instead can be still performed.

1.4.4 Some used methods to assess reliability of scales

Pearson's correlation coefficients are not statistically correct, because they are performed only with quantitative variables. With qualitative variables we should use Kendall's τ, Goodman and Kruskal's γ or Spearman's ρ[20] correlation indexes.

Therefore we suggest to do the same analysis we did before, but changing Pearson's correlation coefficients with Spearman's ρ correlation coefficients.

[19] For example it could be possible to insert a particular question in every questionnaire in order to determine a ranking of the informants and, in this way, the best informant for every item (see [23]); but with our example this is not possible any more because the survey has been already done and data has been already collected.

[20] As a matter of fact we should use this third index only with distinct ranks; nevertheless the index seems to work correctly also if there are repeated ranks.

1.4.5 Some used methods to assess validity and unidimensionality of scales

If we want to do also this step in a methodologically correct way, we propose to use corrispondence analysis instead of factor analysis; in fact corrispondence analysis is substantially a factor analysis for not quantitative variables.

1.4.6 Aggregation of multiple items in a unique scale

From a methodologic point of view, the simple sum (or mean) of the items in order to obtain a scale is not so correct; the main problem is that, as we have already said in section 1.2, we cannot check the equidistance of intervals, furthermore we can obtain the same score in several ways: for example, if we have four items (with all modalities in the discrete range 1-7), the summating scale we can construct has the discrete range 4-28, but we can obtain a same number (for example 22) in several ways (for example: 7 7 7 1, 7 2 6 7, 5 5 5 7, etc.).

Some recently proposed methods that we could use to perform this second aggregation in a more sophisticated (and maybe correct) way are: fuzzy clustering methods, methods based on ordering functions, methods based on multi-criteria analysis, methods based on non parametric techniques[21]. If you want to do a simple choice we advise to use Fisher's combining function[22] after reducing all items in the interval $(0, 1)$ thanks to this transformation:

$$z = \frac{x + 0.5}{C + 1} \tag{1.2}$$

where C is the number of modalities of the items (7 in our example), x is the previous score and z is the new score; the constants added at the numerator (0.5) and at the denominator (1) are useful to prevent z to assume border

[21] But we will not treat them hear analitically, a part from multi-criteria analysis (see subsection 1.4.6 at 58); for further information about them, it is enough you see [5].

[22] $\lambda_{ik} = -2\sum_{j=1}^{J_k} z_{ijk}$, where J_k is the number of items of k-th scale, z_{ijk} is the score of the i-th unit of analysis on j-th item of k-th scale and λ_{ik} is the score of the i-th unit on the whole k-th scale.

values 0 and 1; 0 would be problematic because log(0) is not defined, 1 could be problematic in further analysis because $\log(1) = 0$.

1.4.7 Assessing presence of multicollinearity in indipendent variables

In order to check if future indipendent variables of our model are very correlated or not, we could use another index instead of Pearson's correlation index; as we have already said, correlation indexes more correct for qualitative ordinal variables are Spearman's ρ[23], Kendall's τ or Goodman and Kruskal's γ. Anyway, afterall using Spearman's ρ, results should not be so different from the ones using Pearson's index, but we propose this way only because we want the second analysis to be completely (or at least almost[24] completely) correct from a statistical-methodological point of view.

1.4.8 Final analysis

Linear regression has some particular assumptions that cannot be respected in Likert's scales: the main one is normality of random errors and, therefore, of the response variable (but every unit can come from a mean-different normal distribution) and that can occur only if the response variable is quantitative continuous. Very often the dependent variable is also a scale and in this case we cannot use linear regression at all. Let us sum up different types of regression in function of the nature of the dependent variable in case of indipendent observations[25]:

[23] Indeed there are even some management papers which use this index instead of Pearson's index, as for example [42].

[24] Probably the reader will find some aspects that even in the second analysis are not completely correct from a methodological point of view.

[25] I have done this classification on my personal statistical knowledge, in particular using slides of a statistics course ([46]) of the Faculty (Statistical Sciences) of my University (Università degli studi di Padova). Anyway you can find this information in most of all statistics books related with statistical models.

- linear regression: the dependent variable is quantitative continuous and three assumptions must be respected: normality, homoscedasticity of the random errors and linearity of the relation;
- gamma regression: the dependent variable is quantitative continuous positive: this model is very useful when the response variable is time or money;
- Poisson regression: the dependent variable is qualitative discrete, in particular an account;
- logistic regression: the dependent variable is dichotomic and with this model we can estimate the probability of the two events in function of the values of some indipendent variables;
- multi-logistic regression: the dependent variable is qualitative nominal;
- ordered-logistic regression: the dependent variable is qualitative ordinal: this model can be very useful for dependent variables that are Likert's scales[26].

[26] We have found also some papers where there is the same idea we had, that is comparing results of linear regression and ordered logit regression when the response variable is based on a Likert's scale: see [33] and [50]. Other papers dealing with ordered logit regression are [26] and [47].

Chapter 2

Multivariate ranking methods: theoretical backgrounds and critical comparison

2.1 Introduction and motivation

Firm's applied problems are often related to datasets observed over more units (subjects, samples of product unit, etc.), with reference to several variables (evaluations, product performances, etc.), with the aim of studying the relationship between these variables and a factor of interest under investigation (a given firm's feature, product, etc.). In this framework the main goal is to compare the factor levels (features, products, etc.), with respect to all variables, in order to find out the "best" one.

From a statistical point of view, when the response variable is multivariate in nature, the problem may become quite difficult to cope with, because the dimension of the parametric space may be very large. This situation can arise very often the context of the overall quality assessment of products, where evaluations are provided by taking into account for several aspects and points of view (for example performances of new products: strength, pleasantness, appropriateness of a new fragrance, punctuality, assistance, distribution network of a new service).

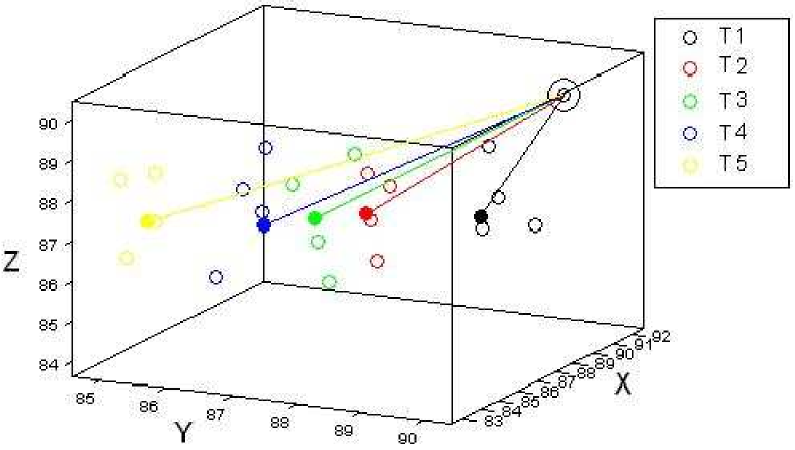


Figure 2.1: Example of a typical situation we are interested to investigate.

In fig. 2.1 we have illustrated a situation we could be interested to investigate: in this case we have 20 statistical units, that are part of 5 different groups (different treatments or products), on which three variables (X, Y and Z) are relevated; the full-coloured points represent the real average values of the three variables for each group, the empty-coloured points represent the observations and the circled point on the top is the optimum point, that represents the whole of values that the average values should assume in order to have the best treatment of all. So, it should be clear that the less a medium point of a group is near the optimum point, the better that group is; as a result, in the example of the figure the true ranking of the 5 groups is: black $\succ$ red $\succ$ green $\succ$ blue $\succ$ yellow.

The topic of defining a treatment ranking from a multivariate point of view seems to be quite recent: it has been firstly addressed by Bonnini, Corain, Salmaso et al. in 2006[1] and the reference framework is experimental design and analysis of variance. The literature of multiple comparison methods addresses the problem of ranking the treatment groups from worst to best, however there is no clear indications on how dealing with the information

[1] See [9].

from pairwise multiple comparisons, especially in case of blocking (or stratification) or in case of multivariate response variable.
This problem is not only of theoretical interest but it has also a recognized practical relevance, especially for applied research. Moreover, in industrial research a global ranking in terms of performance of all investigated products/prototypes is very often a natural goal; as a proof, in 2008 an international industrial organization called AISE has formally incorporated such a method as official standard for industrial research on house cleaning products[2].

2.2 Theoretical background

2.2.1 The ANOVA model

Let Y be the multivariate numeric variable related to the response of any experiment of interest and let us assume, without loss of generality, that high values of each Y univariate element correspond to better performance and therefore to a higher degree of treatment preference. The experimental design of interest is defined by the comparison of C groups or treatments with respect to S different variables where n replications of a single experiment are performed by a random assignment of a statistical unit to a given group. The C-group multivariate statistical model (with fixed effects) can be represented as follows:

$$Y_{ijk} = \mu_{ij} + \varepsilon_{ijk}, \quad \varepsilon_{ijk} \sim \mathrm{IID}(0, \sigma^2_{ij}), \quad i = 1, ..., C; j = 1, ..., S; k = 1, ..., n; \tag{2.1}$$

where, in the case of a balanced design, n is equal to the number of replications and indexes i and j are related with the groups (treatments) and the univariate response variable respectively.
The resulting inferential problem of interest is concerned with a set of S hypothesis testing procedures $H_{0j} : \mu_{1j} = \mu_{2j} = \ldots = \mu_{Cj}$ vs. $H_{1j} : \exists \mu_{ij} \neq \mu_{hj}$,

[2] See [3] and subsubsection 2.2.2 at page 23.

$i, h = 1, \ldots, C$, $i \neq h$, $j = 1, \ldots, S$. If H_{0j} is rejected a further possible set of $C \times (C-1)/2$ all pairwise comparisons are performed:

$$\begin{cases} H_{0ih|j}: & \mu_{ij} = \mu_{hj} \\ H_{1ih|j}: \exists & \mu_{ij} \neq \mu_{hj} \end{cases}.$$

In the framework of parametric methods, when assuming the hypothesis of normality for random error components, the inferential problem can be solved by means of the ANOVA F-test and a further set of pairwise tests using Fisher's LSD or Tukey procedures, which are two of most popular multiple comparison procedures[3]. On the basis of inferential results achieved at the univariate C-group comparison stage, the next step consists in producing a ranking of the treatments from the less to the more preferred.

2.2.2 5 proposed ranking methods

Scheme of ranking methods we are proposing

To perform almost all methods we are about to propose in the following subsubsections, we have to execute these steps:

1. the starting point is the result of the multiple comparisons analysis (S $C \times C$ p-value matrices);
2. a suitable score matrix is then defined;
3. through a synthesis procedure (sum, mean or some combination function) the scores are synthesized into a C-dimensional score vector;
4. the set of S score vectors are finally synthesized to perform one global score vector[4];

[3] See [36].

[4] We could also choose to use in the combining function different weights for the S variables used to produce the score in order to give different levels of importance to them; in the 5 ranking methods we are presenting we will not use weights, or, if you want, we will use all weights equal to $\frac{1}{S}$.

5. the rank[5] of this final global scores provides the required multivariate global ranking of treatments.

We are about to present 5 ranking methods: three of them (AISE, NPC and GPS) are of particular scientific interest as we have already explained in the previous section and follow completely the scheme descripted above; the last two (Method “0” and Method “1”) are only used as terms of comparisons for the first three.

AISE score method

In 2007 Corain and Salmaso[6] proposed to sum some meaningful scores from inferential results at the univariate C-group comparison and then to apply the Non Parametric Combination (NPC) of partial rankings[7]. In this way they acquire a unique preference criterion which jointly takes into account all performances achieved for every response variable. To illustrate the method, let us suppose H_{0j} has been rejected for all $j = 1, ..., S$, so that for each univariate response there is some treatment that significantly differs for some others.

In order to suitably synthesize the pairwise comparison results for each response variable j, $j = 1, ..., S$, let us define a set of S score matrices of dimension $C \times C$, where each element $[x_{ih|j}]$ is related to the comparison between the treatments i and h for each response variable j, giving the value of $+1$ to the significantly better treatment and -1 to the other, while both

[5] As we use rank transformation it is better to clear up that, from this moment and for all this work, 1 is the worst treatment/product/group and C is the best one, therefore with this choice the more the rank is high, the better the treatment is; this choice can appear a little strange, but we have done it essentially for two reasons: first because almost all scores are higher if the performance is better and lower if the performance is worse and second because the software we have used, R (see [39]), for default behaves like this.

[6] See [13].

[7] See [29].

scores are 0 if the comparison is not significant. Formally,

$$\begin{cases} \text{if } H_{0ih|j} : \mu_{ij} = \mu_{hj} & \text{is not rejected then } x_{ih|j} = x_{hi|j} = 0; \\ \text{if } H_{0ih|j} : \mu_{ij} = \mu_{hj} & \text{is rejected then } \begin{cases} \text{if } \bar{y}_{ij} > \bar{y}_{hj} & \text{then } x_{ih|j} = +1 \text{ and } x_{hi|j} = -1; \\ \text{if } \bar{y}_{ij} < \bar{y}_{hj} & \text{then } x_{ih|j} = -1 \text{ and } x_{hi|j} = +1; \end{cases} \end{cases} \tag{2.2}$$

where $\bar{y}_{ij}$ and $\bar{y}_{hj}$, $i, h = 1, \ldots, C$, $i \neq h$, are the sample means of groups i and h for response variable Y_{ij}, $i = 1, \ldots, C$, $j = 1, \ldots, S$. Note that pairwise comparisons and the valid score assignments are performed only when the C-sample test has rejected the null hypothesis H_{0j}, $j = 1, ..., S$.
For each response variable $j = 1, ..., S$ once this assignment has been performed for each pairwise comparison, it is easily feasible to obtain a set of S $X_j = [x_{1|j}, x_{2|j}, \ldots, x_{C|j}]'$ score vectors, $j = 1, ..., S$, where the C elements $x_{i|j}$, $i = 1, \ldots, C$, $j = 1, ..., S$, of the score vector X_j, are calculated by summing all the obtained scores for each treatment, i.e.:

$$x_{i|j} = \sum_{h=1, h \neq i}^{C} x_{ih|j}, \qquad i = 1, \ldots, C, \quad j = 1, \ldots, S. \tag{2.3}$$

In order to obtain the final global AISE score we just apply a simple sum:

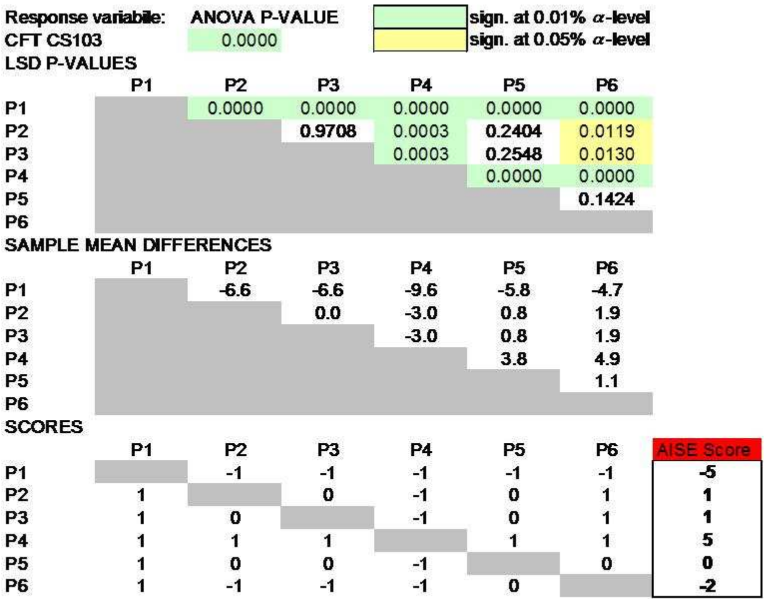
$$AISE_i = \sum_{j=1}^{S} x_{i|j}, \qquad i = 1, \ldots, C \tag{2.4}$$

In the end, we obtain the global combined ranking by applying the rank transformation:

$$G_i^{AISE} = R(AISE_i) = \#(AISE_i \geq AISE_h), \quad i, h = 1, \ldots, C. \tag{2.5}$$

NPC score method

Instead of using ±1 summation as proposed by Corain and Salmaso in 2007, we could also make directly use of p-values. For this goal let us consider the set of S one-sided p-value matrices of dimension $C \times C$, where each element



Response variabile: ANOVA P-VALUE 🟩 sign. at 0.01% α-level

CFT CS103 0.0000 🟨 sign. at 0.05% α-level

LSD P-VALUES

	P1	P2	P3	P4	P5	P6
P1		0.0000	0.0000	0.0000	0.0000	0.0000
P2			0.9708	0.0003	0.2404	0.0119
P3				0.0003	0.2548	0.0130
P4					0.0000	0.0000
P5						0.1424
P6						

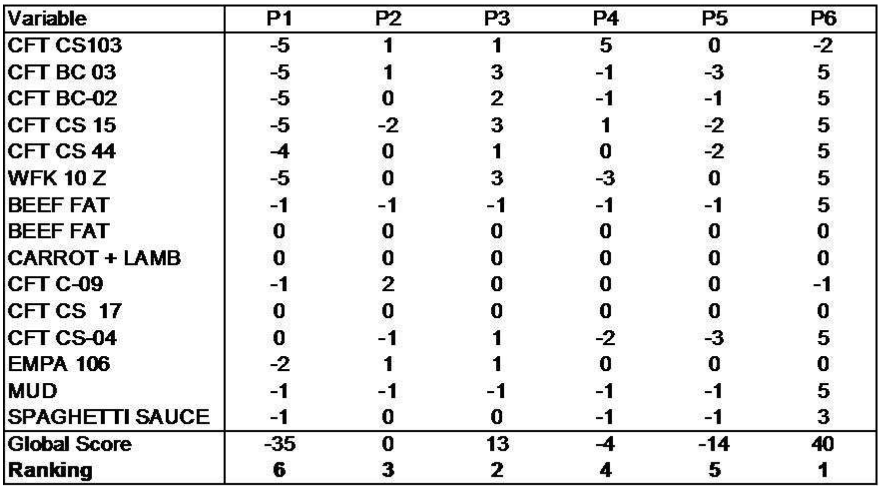
SAMPLE MEAN DIFFERENCES

	P1	P2	P3	P4	P5	P6
P1		-6.6	-6.6	-9.6	-5.8	-4.7
P2			0.0	-3.0	0.8	1.9
P3				-3.0	0.8	1.9
P4					3.8	4.9
P5						1.1
P6						

SCORES

	P1	P2	P3	P4	P5	P6	AISE Score
P1		-1	-1	-1	-1	-1	-5
P2	1		0	-1	0	1	1
P3	1	0		-1	0	1	1
P4	1	1	1		1	1	5
P5	1	0	0	-1		0	0
P6	1	-1	-1	-1	0		-2

Figure 2.2: Example of calculation of AISE score for one variable.



Variable	P1	P2	P3	P4	P5	P6
CFT CS103	-5	1	1	5	0	-2
CFT BC 03	-5	1	3	-1	-3	5
CFT BC-02	-5	0	2	-1	-1	5
CFT CS 15	-5	-2	3	1	-2	5
CFT CS 44	-4	0	1	0	-2	5
WFK 10 Z	-5	0	3	-3	0	5
BEEF FAT	-1	-1	-1	-1	-1	5
BEEF FAT	0	0	0	0	0	0
CARROT + LAMB	0	0	0	0	0	0
CFT C-09	-1	2	0	0	0	-1
CFT CS 17	0	0	0	0	0	0
CFT CS-04	0	-1	1	-2	-3	5
EMPA 106	-2	1	1	0	0	0
MUD	-1	-1	-1	-1	-1	5
SPAGHETTI SAUCE	-1	0	0	-1	-1	3
Global Score	-35	0	13	-4	-14	40
Ranking	6	3	2	4	5	1

Figure 2.3: Example of calculation of global AISE score (more than one variable).

$[p_{ih|j}]$ is related to the comparison between treatments i and h for response variable j. For each response variable j, $j = 1, ..., S$, it is possible to obtain an alternative set of S score vectors X_j, $j = 1, ..., S$, where each element of

X_j is calculated as follows:

$$x_{i|j} = -2 \sum_{h=1, h\neq i}^{C} \log(p_{ih|j}), \quad i = 1, \ldots, C, \quad j = 1, ..., S. \tag{2.6}$$

Note that we use as p-value synthesis criterion the Fisher's combining function. It is worth noting that the Fisher's combining function is non-parametric with respect to the underlying dependence structure among p-values from different univariate response variables, in that all kinds of monotonic dependences are implicitly captured. Indeed, no explicit model for this dependence structure is needed and no dependent coefficient has to be estimated directly from the data.

Then, in order to suitably synthesize the scores for each response variable j, $j = 1, \ldots, S$, Corain and Salmaso suggested to use the Non-Parametric Combination (NPC) of partial rankings[8] to acquire a unique preference criterion which jointly takes into account all response variables. In order to obtain the final global NPC score we apply just a simple sum:

$$NPC_i = \sum_{j=1}^{S} x_{i|j}, \qquad i = 1, \ldots, C \tag{2.7}$$

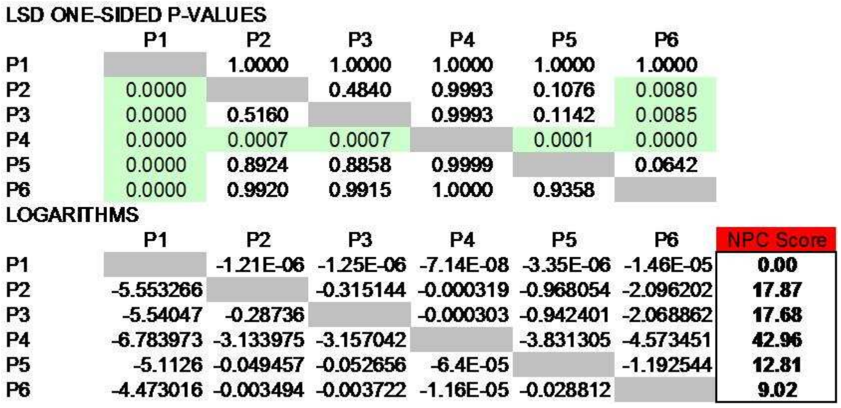
In the end, we obtain the global combined ranking by applying the rank transformation:

$$G_i^{NPC} = R(NPC_i) = \#(NPC_i \geq NPC_h), \quad i, h = 1, \ldots, C. \tag{2.8}$$

GPS score method

With respect to each response variable an ANOVA test is performed and from the usual $C \times (C-1)/2$ pairwise comparisons it is possible to test the statistical significance of the differences between the mean performances for each couple of treatments (u, v). Let us indicate with $\overline{y}_{(1)j} \geq \overline{y}_{(2)j} \geq \cdots \geq \overline{y}_{(C)j}$ the *ordered* observed sample means for Y_j and assume that high values correspond to better performance. The algorithm to calculate GPS score is the following:

[8] See [29].



LSD ONE-SIDED P-VALUES

	P1	P2	P3	P4	P5	P6
P1		1.0000	1.0000	1.0000	1.0000	1.0000
P2	0.0000		0.4840	0.9993	0.1076	0.0080
P3	0.0000	0.5160		0.9993	0.1142	0.0085
P4	0.0000	0.0007	0.0007		0.0001	0.0000
P5	0.0000	0.8924	0.8858	0.9999		0.0642
P6	0.0000	0.9920	0.9915	1.0000	0.9358	

LOGARITHMS

	P1	P2	P3	P4	P5	P6	NPC Score
P1		-1.21E-06	-1.25E-06	-7.14E-08	-3.35E-06	-1.46E-05	0.00
P2	-5.553266		-0.315144	-0.000319	-0.968054	-2.096202	17.87
P3	-5.54047	-0.28736		-0.000303	-0.942401	-2.068862	17.68
P4	-6.783973	-3.133975	-3.157042		-3.831305	-4.573451	42.96
P5	-5.1126	-0.049457	-0.052656	-6.4E-05		-1.192544	12.81
P6	-4.473016	-0.003494	-0.003722	-1.16E-05	-0.028812		9.02

Figure 2.4: Example of calculation of NPC score (for one variable).

1. for each of the S variables a $C \times C$ matrix X is created (see the example in table 2.1 at page 29) where the elements under the main diagonal are null and those over the main diagonal take value 0 or 1 according to the following rule:

$$X[u,v] = h\left[\overline{y}_{(u)j}, \overline{y}_{(v)j}\right] \begin{cases} 1 \quad \text{if} \quad \overline{y}_{(u)j} & \text{is significantly not equal to} \quad \overline{y}_{(v)j} \\ 0 & \text{otherwise;} \end{cases} \tag{2.9}$$

2. a rank table, as shown in the example of table 2.1, is created according to the following steps:

    (a) in row 1, rank C is assigned to the treatment with the higher mean (first column), indicated with (1), and to all the other products which mean performances are not significantly different from that of (1);

    (b) in row 2, rank $C-1$ is assigned to the treatment with the higher mean, among those excluded from rank C assignation, and to all the other products which mean performances are not significantly different from that of (2);

    (c) in row r, rank $C-r+1$ is assigned to the treatment with the higher mean, among those excluded from rank $C-r$ assignation, and to all the other products which mean performances are not significantly different from that of (r);

(d) the iterated procedure stops when a rank is assigned to the product (C);

3. for each treatment, the arithmetic mean of the values from the rank table (mean by columns) gives a partial performance score: $x_{(i)j}$;

4. in order to obtain the final global GPS score we apply, as usually, just a simple sum:

$$GPS_i = \sum_{j=1}^{S} x_{(i)j}, \qquad i = 1, \ldots, C; \tag{2.10}$$

5. in the end, we obtain the global combined ranking by applying, as usually, the rank transformation:

$$G_i^{GPS} = R(GPS_i) = \#(GPS_i \geq GPS_h), \quad i, h = 1, \ldots, C. \tag{2.11}$$

"0" method (mean of the means)

This is the first of the two methods we have used as terms of comparison for the previous three and it is very simple: it consists only in calculating the sample mean table (the mean of the variables in the different groups, so a $C \times S$ table) and then in aggregating with respect to the variables, simply by averaging and in this way obtaining a unique value for every group; the ranking is then obtained by applying the rank transformation to these values. In symbols operations made on observations y_{ijk} are:

$$\overline{y}_{ij} = \frac{1}{n} \sum_{k=1}^{n} y_{ijk}, \quad i = 1, \ldots, C; j = 1, \ldots, S; k = 1, \ldots, n. \tag{2.12}$$

$$0_i = \overline{y}_i = \frac{1}{S} \sum_{j=1}^{S} \overline{y}_{ij} \tag{2.13}$$

Ord. tr.	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
(1)		0	0	1	1	1	1	1
(2)			0	0	0	0	1	1
(3)				0	0	0	0	1
(4)					0	0	0	0
(5)						0	0	0
(6)							0	0
(7)								0
Rank								
4	8	8	8					
3		7	7	7	7	7		
2			6	6	6	6	6	
1				5	5	5	5	5
$Z_{(c)tsk}$	8	7.5	7	6	6	6	5.5	5

Table 2.1: Example of X matrix for the multiple comparisons between pairs of products ($C = 8$).

$$G_i^0 = R(0_i) = \#(0_i \geq 0_h), \quad i, h = 1, \ldots, C. \tag{2.14}$$

This method is indeed based on no real score, but on two simple arithmetical means; anyway it can be seen as an application of Multicriteria Methods (see subsections 1.4.6 at page 16 and above all 3.2.2 at page 58).

"1" method (based on confidence intervals of distances)

This is the other method used as term of comparison, but this one is a little bit complicated, based on the distance between the observation and the optimum point, it consists in the following steps:

1. euclidean distances between units (y_{ijk}) and optimum point (y_{ij}^o) are

calculated for each unit[9];

$$d_{ik} = \sqrt{\sum_{j=1}^{S}(y_{ijk} - y_{ij}^{o})^2} \quad i = 1, \dots, C; j = 1, \dots, S; k = 1, \dots, n. \tag{2.15}$$

2. arithmetic means of the distances are calculated, in order to obtain a unique value for every group (after this step we have a C-dimensional vector of means of distances);

$$\overline{d}_i = \frac{1}{n}\sum_{k=1}^{n} d_{ik} \tag{2.16}$$

3. from the vector of *ordered* means of distances confidence intervals[10] are calculated and then a score $C \times C$ matrix X of comparisons among groups is defined: 0 means that the two confidence intervals of that cell do intersect and therefore that the comparison is not significant, 1 means the contrary, that is the comparison is significant because the two intervals do not intersect; clearly, because of the ordering, the structure of matrix X is very similar to the matrix used by GPS score (1s will be jointed on the top right and over the diagonal of the matrix, 0s will be elsewhere); the two limits of the confidence interval are:

$$d^1_{(i)} = \overline{d}_{(i)} - t_{1-\frac{\alpha_1}{2};nC-C} \qquad d^2_{(i)} = \overline{d}_{(i)} + t_{1-\frac{\alpha_1}{2};nC-C} \tag{2.17}$$

where $d_{(i)}$ indicate the distances obtained at the previous step after we have ordered them.

4. to X GPS score for one variable is applied, in order to obtain a C-dimensional vector of scores: $1_{(i)}$;

[9] Notice that if we keep assuming that to large values correspond better performances as we are doing (see the beginning of this section), to small (and not to large) distances correspond better performances.

[10] We suggest to use Bonferroni's correction, in order to obtain *simultaneous* confidence intervals, that simply consists in using a significance level that is the significance level declared (very often 0.05) divided for the number of comparisons we are going to do, that is $\binom{C}{2}$; the degrees of freedom are $nC - C = n(C - 1)$.

5. afterwards the rank transformation is applied, but pay attention because this is the only case where to lower scores correspond better performances and to higher scores correspond worse performances, therefore the rank transformation is applied indeed to the opposite values of the score (we mean for example -4 instead of 4):

$$G_i^1 = R(-1_{(i)}) = \#(-1_{(i)} \geq -1_{(h)}), \quad i, h = 1, \ldots, C. \tag{2.18}$$

Mean and variance of the proposed scores

For our future research on these topics and to inhance the evaluation of the results of simulations performed in this work, it is very useful to calculate mean and variance of the three scores of scientific interest, in order to obtain a way to better calculate[11] confidence intervals of the scores. In fact the scores defined by different procedures (both parametric and non-parametric) can be viewed as realizations of appropriate random variables and depending on hypothesis of random errors, the distribution of these random variables can be derived (parametrically or non-parametrically) in an exact or asymptotically way.

AISE We have decided not to calculate mean and variance of AISE score because it would be not useful: as you will be able to see from the results of the simulation studies, it is clear that AISE score performs similar, but quite worse than GPS score and much worse than NPC score.

NPC In order to find a way to calculate mean and variance of NPC score, we have begun from this point[12]: under the null hypothesis H_0 (that is: "the groups are all equivalent") we have that

$$\psi_{ijk0} = -2 \sum_{h=1, h \neq i}^{C} \log(p_{ih|j}) \sim a\chi_g^2, \quad k = 1, \ldots, B \tag{2.19}$$

[11] Better than those that we will use in our simulation studies, which are more pseudo-confidence intervals of the scores than real true confidence interval of the scores.

[12] This starting point was suggested by Hinkley in [24].

where a is a parameter that takes into account the possible (probable indeed) dependence between the components of the sum and if $a = 1$ we have independence; g, instead, which represents the degrees of freedom of the chi-squared, is simply twice the number of components of the sum[13]; i is the index of the group ($i = 1, \dots, C$), j the one relative to the variable ($j = 1, \dots, S$) and, finally, k is the index of the simulations (B is the number of them and normally we choose $B = 1000$ or $B = 10000$). Substantially we consider that sum, which synthesizes every simulation, the realization of a chi-squared variable modified with an appropriate multiplicative constant.

So, we could think[14] that, by adding a certain parameter δ, we could report the distribution under alternative hypothesis H_1 to the one, simpler, under H_0:

$$\psi^{\delta}_{ijk1} = -2 \sum_{h=1, h \neq i}^{C} \log(p_{ih|j}) \sim a\chi^2_g, \quad k = 1, \dots, B. \tag{2.20}$$

Therefore, remembering properties of expectation and variance and that the mean of a chi-squared variable is its degrees of freedom and that the variance of a chi-squared variable is twice its degrees of freedom, we can state that:

$$E(\psi^{\delta}_1) = ag \tag{2.21}$$

$$Var(\psi^{\delta}_1) = 2a^2g \tag{2.22}$$

And therefore a possible good estimation of parameter a could be:

$$\hat{a} = \frac{1}{2}\frac{\hat{Var}(\psi^{\delta}_1)}{\hat{E}(\psi^{\delta}_1)}, \tag{2.23}$$

because:

$$\frac{Var(\psi^{\delta}_1)}{E(\psi^{\delta}_1)} = \frac{2a^2g}{ag} = 2a. \tag{2.24}$$

[13] Therefore $g = 2(C - 1)$.

[14] This was an idea of Lehmann who applied this to the uniform variable: under null hypothesis p-value is distributed as a $Unif(0, 1)$ variable, so we could think that $p^{\delta} \sim Unif(0, 1)$); nobody and nothing forbids us to use the same trick with a statistic distributed as a chi-squared variable.

With a simulation study it is obviously very simple to calculate $\hat{E}(\psi_1^\delta)$ and $V\hat{a}r(\psi_1^\delta)$, good estimations of the true $E(\psi_1^\delta)$ and $Var(\psi_1^\delta)$.
In order to estimate δ, probably we have to do it in a numeric way, for example by minimizing a loss function; maybe it is also possible to do it in another way but still more methodological research on this topic is needed.
We could say that all the reasoning about the calculation of expectation and variance of NPC score is based on a semi-parametric estimate of a distribution.

GPS It is important to begin this paragraph saying that our research was not enough to find a real method to calculate expectation and variance of GPS score, anyway until now we have reached a good point and we are working on it.
The first idea we had and we tried to follow, was to find a $C \times Q$[15] matrix A that pre-multiplicated to the Q-dimensional vector Y (that is the matrix of 0s and 1s written as a vector) becomes the C-dimensional vector of the scores (in the end we have always one score for every different group/treatment), in symbols:

$$AY = S; \tag{2.25}$$

this idea seemed to be good because elements of Y are almost all realizations of bernoullian variables and, for the central limit theorem, we could approximate the whole vector to a multivariate normal distribution, so that it would be very simple to calculate approximated estimations of $E(S) = E(AY)$ and $Var(S) = Var(AY)$ by finding matrix A.
But there are lots of problems following this idea, so we have tried another way: let us take as example the score matrix of table 2.1 at page 29 (see fig. 2.2). As we have already said, we can consider the scores $x_{ij}, \quad i,j = 1,\dots,C$ in the matrix as realizations of known causal variables:

$$X_{ij} \sim \text{Ber}(p_{ij}), \text{ or analogously } X_{ij} \sim \text{Bin}(1, p_{ij}), \tag{2.26}$$

[15] $Q = SC(C-1)$.

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ & 0 & 0 & 0 & 0 & 1 & 1 \\ & & 0 & 0 & 0 & 0 & 1 \\ & & & 0 & 0 & 0 & 0 \\ & & & & 0 & 0 & 0 \\ & & & & & 0 & 0 \\ & & & & & & 0 \\ & & & & & & \end{bmatrix}$$

Table 2.2: Beginning score matrix useful to get E and Var of the final GPS score.

where p_{ij} is the probability that the comparison between group i and group j is significant, therefore:

- we can estimate every p_{ij}, $i = 1, \ldots, C-1; j = i+1, \ldots, C$, (so only probabilities of the comparisons *written* in the beginning score matrix) with the p-value of that comparison;
- where scores are not written, we can consider them to be *always* 0, so $p_{ij} = 0$, $i = 1, \ldots, C; j = 1, \ldots, i$; in this way on and below the diagonal of the matrix we can consider to have realzations of a degenerate variable.

Notice that now we can estimate expectation and variance of these $C \times C$ scores, remembering that for a bernoullian variable X with parameter θ, $E(X) = \theta$ and $Var(X) = \theta(1-\theta)$.
At this point we had the idea of transforming the beginning matrix in another one where written scores *and* scores on the diagonal are inverted (1 appears instead of 0 and viceversa) and the *other* not written scores are considered to stay equal to 0; therefore in our guiding example the matrix would become the one you can see in table 2.3:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ & & 1 & 1 & 1 & 1 & 1 & 0 \\ & & & 1 & 1 & 1 & 1 & 1 \\ & & & & 1 & 1 & 1 & 1 \\ & & & & & 1 & 1 & 1 \\ & & & & & & 1 & 1 \\ & & & & & & & 1 \end{bmatrix}$$

Table 2.3: First passage matrix useful to get E and Var of the final GPS score.

Notice that even now we can estimate expectation and variance of the single scores: over the diagonal the probabilities of success and failure are now inverted and are respectively $1 - p$ and p.
Now in the first line we change 1 with C, in the second with $C - 1$ and so on until the first line in which only 1s *compare written*[16] and after that line we write 0 in all last lines of the matrix; furthermore all 0s of the table are cancelled. So, in our example we have now the matrix you can see in fig. 2.4. Even now the moments of interest (E and Var) of the elements of this matrix can be simply estimated (we have to deal with bernoullian variables multiplicated with some constants).
At this point we have some problems and we need some more methodological research to go on; we had thought two ways to obtain the vector of the C scores, which in our example is indicated in fig. 2.5. The first way we had thought would be to find a certain C-dimensional vector, that post-multiplicated to the last calculated $C \times C$ score matrix (see table 2.4) should give the C-dimensional vector of scores, but this way seems not practicable at all. The second way would be to obtain the final C scores by averaging

[16] That is we are not taking into account lines with 0s which are not written.

$$\begin{bmatrix} 8 & 8 & 8 & & & & & \\ & 7 & 7 & 7 & 7 & 7 & & \\ & & 6 & 6 & 6 & 6 & 6 & \\ & & & 5 & 5 & 5 & 5 & 5 \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \end{bmatrix}$$

Table 2.4: Second passage matrix useful to get E and Var of the final GPS score.

$$\begin{bmatrix} 8 & 7.5 & 7 & 6 & 6 & 6 & 5.5 & 5 \end{bmatrix}'.$$

Table 2.5: Final scores.

by column the scores of the last $C \times C$ matrix, taking into account only the scores which are different from 0 (so, only the written ones). For example the fourth score of the example above is 6, because 6 is the mean of the written values of the fourth column of the last score matrix, that are 5, 6 and 7 and the last one is 5, because in the last column only one 5 is written.
But where is there the problem in this second choice? Until now after all passages we have always obtained a score matrix which components are descriptable in terms of expectation and variance of the causal variables of which they are realizations, but on the final score vector it is very difficult to calculate expectation and variance, because we should calculate expectation and variance of the ratio of two causal variables: in fact the final scores

$x_i, \quad i = 1, \dots, C$ are calculated in this way:

$$x_i = \frac{\sum_{j=1}^{K_i} \tilde{x}_{ij}}{K_i},$$

where $\tilde{x}_{ij}$ are the scores of the last $C \times C$ matrix and K_i is the causal variable that describes the number of written values for the column i of the matrix. Therefore we are now not able any more to go on and to calculate expectation and variance of the final scores, but probably we are very close to our goal. We must also say that we have not already taken in account that if in the beginning score matrix we have two equal lines, we have to jump one of them and this can complicate the calculation of the moments during all passages we have illustrated.

Anyway the situation is not so bad, because the way of calculating the moments of the most important score, NPC, is almost completely sketched out and with the calculation of the moments of GPS score, as we have just said, we are not so far to complete it with some more research.

2.3 Simulation study

In order to evaluate the degrees of accuracy of these methods in detecting the true “unknown” treatment ranking, in this section we will perform a first simulation study where parameters are set and errors are randomly generated. In sections 2.5 and 3.3, respectively at pages 44 and 61, you will find other two simulation studies: the second uses true experimental data, the third is similar to the first, but contains some issues more than the first one. The main goals of these three simulation studies are to find out the best performing and the more robust multivariate ranking methods and to investigate the role played by the number of treatment to be ranked, the dimension of the response variable and the number of replications.

In these simulation studies we have applied the ranking methods (NPC, AISE, GPS) only to quantitative variables, but these methods are completely applicable also to qualitative variables, and in particular also ordinal vari-

ables (therefore also Likert's scales): to adapt the three methods to ordinal variables it is enough to obtain p-values of comparisons with an appropriate test for ordinal variables, instead of t-test, because as we can obtain some p-values that describe the pairwise comparisons, than we are completely capable to perform the three methods. It is very interesting that we can apply these methods also to ordinal variables and in particular to Likert's scales, because it could be possible to use them in the analyses of questionnaires with Likert's scales, analyses which we descripted in the first chapter.
But, as we have already said, now we are going to deal only with quantitative variables, even if, results are extendable to other types of variables (the main point is using a correct test for the comparisons), and in this section we are dealing only with the first simulation study; let us consider the following simulation settings:

- 3 and 6 positive numeric response variables ($p = 3, 6$), with "hypothetical" maximum value at 100 (the optimum point);
- 3, 5 and 7 treatments ($C = 3, 5, 7$);
- 4 and 8 experimental replications ($n = 4, 8$);
- three types of random errors ($\varepsilon_{ijk}, \quad i = 1, \ldots, C; j = 1, \ldots, p; k = 1, \ldots, n$): normal, skew-normal[17] (with shape parameter equal to 5) and Student's t (with 2 degrees of freedom). We have chosen these three distributions in order to have good examples of: a symmetric distribution with light tails (low curtosis) (normal distribution), an asymmetric distribution (we wanted to use exponential distribution, but there were lots of difficulties in using it at a multivariate level, so we chose skew-normal distribution), a symmetric distribution with heavy tails (high curtosis) (Student's t with 2 degrees of freedom);

[17] For further information on this finding of prof. Adelchi Azzalini of Department of Statistical Sciences of Padua, see [6].

- three variance-covariance settings (different variance-covariance matrices $(p \times p)$ of the vectors of random errors):
    1. setting 1: variables are independent ($\sigma^2_{ih} = 0,\ i, h = 1, \ldots, p,\ i \neq h$) and their variances are homoscedastic ($\sigma^2_{ii} = 1,\ i = 1, \ldots, p$);
    2. setting 2: variables are independent ($\sigma^2_{ih} = 0,\ i, h = 1, \ldots, p,\ i \neq h$) and their variances are heteroscedastic ($\sigma^2_{11} = 1,\ \sigma^2_{22} = 4,\ \sigma^2_{33} = 2.25$);
    3. setting 3: variables are not independent (covariances are all different with values in the interval $[0, 1]$) and their variances are homoscedastic ($\sigma^2_{ii} = 1,\ i = 1, \ldots, p$);
- the p $C \times C$ matrices of p-values are obtained with t-test: in NPC and AISE they are unilateral and no correction is applied, whereas in GPS they are bilateral and correction for multiple comparisons is applied[18].
- 1000 independent simulations are performed for each of the 108 settings $(2 \times 3 \times 2 \times 3 \times 3)$;
- fixed structures of true treatment mean values (see the two tables in the next page).

Note that the true global ranking follows the label treatment ordering and can be obtained by calculating the euclidean distance of each treatment from the perfect ideal treatment, that is from the 3-dimensional point $(100, 100, 100)$ with 3 variables or from the 6-dimensional point $(100, 100, 100, 100, 100, 100)$ with 6 variables.

[18] See R codes in appendix A at page 67.

True global ranking			μ_1	μ_2	μ_3	Distance
C=3	C=5	C=7				from optimum
		1	90	90	90	17.3
	1	2	89	89	87	20.3
1	2	3	88	87.5	87	21.7
2	3	4	87	87	87	22.5
3	4	5	86	86.5	87	23.4
	5	6	85	85	87	24.9
		7	84	84	84	27.7

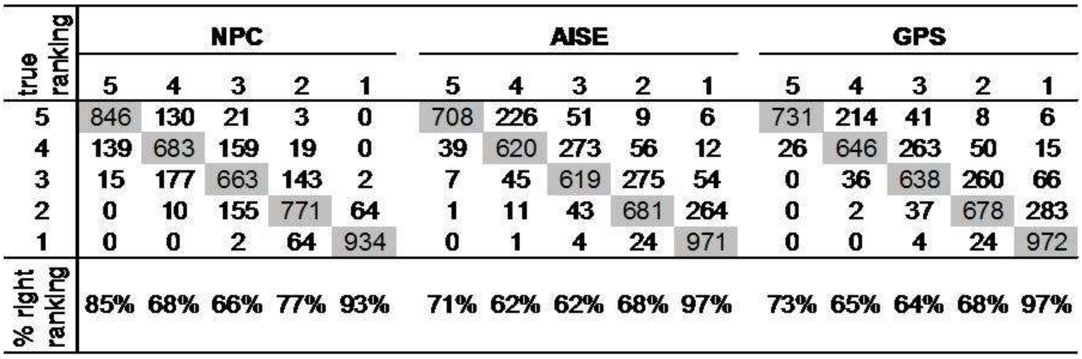
Table 2.6: Setting of treatment mean value for simulation study ($p = 3$).

True global ranking			μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	Distance
C=3	C=5	C=7							from optimum
		1	90	90	90	90	90	90	24.5
	1	2	89	89	87	90	89	89	27.4
1	2	3	88	87.5	87	90	87	87	30.1
2	3	4	87	87	87	87	87	87	31.8
3	4	5	86	86.5	87	84	87	86	34.2
	5	6	85	85	87	84	85	84	36.8
		7	84	84	84	84	84	84	39.2

Table 2.7: Setting of treatment mean value for simulation study with ($p = 6$).

Analyzing results of this first simulation study (you can find an example[19] of how we can present results in fig. 2.5) we can eventually state some considerations:

[19] In this first study we have performed only two indexes/calculations to analyze results for every method for every setting: classification matrix and the percentage of simulations when Spearman's ρ calculated between the true ranking and the calculated ranking is equal to 1 (for further details on these indexes see the second simulation study); in the example of fig. 2.5 this percentage is 52.5 for NPC method, 19.7 for AISE method and 20.2 for GPS method.



true ranking	NPC					AISE					GPS				
	5	4	3	2	1	5	4	3	2	1	5	4	3	2	1
5	846	130	21	3	0	708	226	51	9	6	731	214	41	8	6
4	139	683	159	19	0	39	620	273	56	12	26	646	263	50	15
3	15	177	663	143	2	7	45	619	275	54	0	36	638	260	66
2	0	10	155	771	64	1	11	43	681	264	0	2	37	678	283
1	0	0	2	64	934	0	1	4	24	971	0	0	4	24	972
% right ranking	85%	68%	66%	77%	93%	71%	62%	62%	68%	97%	73%	65%	64%	68%	97%

Figure 2.5: First simulation study: example of results (classification matrices) in the setting with p=3, C=5, n=4, normal errors, indep.-heter. var.-cov. matrix.

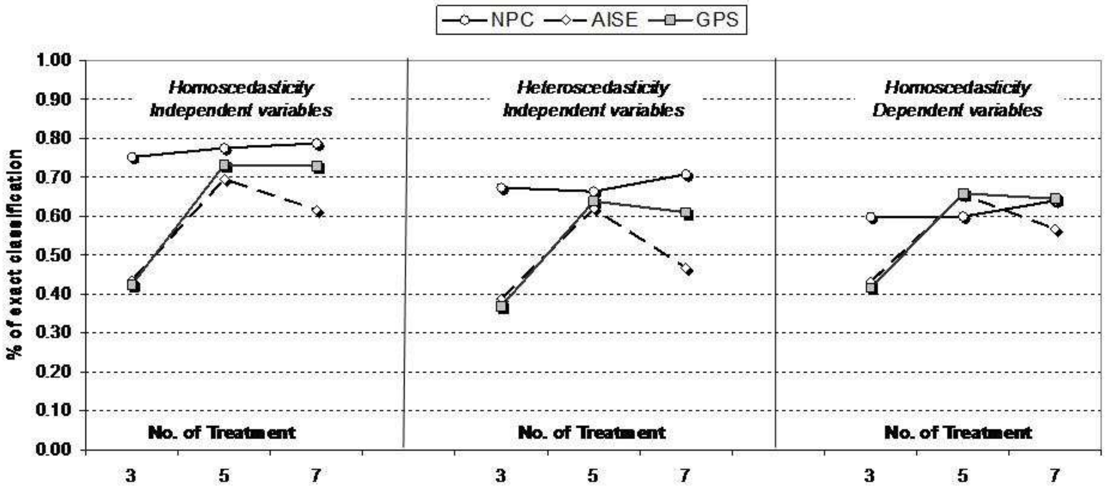


Figure 2.6: First simulation study: example of results (rate of right classification for the median treatment) in the settings with p=3, n=4, normal errors.

- the most performing method between those of interest is NPC, that is much better than AISE; GPS is almost in the middle of the two methods, but much less performing than NPC and not very distant from AISE; method “1” is completely wrong to search the true ranking (we could expect it because it is very difficult that confidence intervals

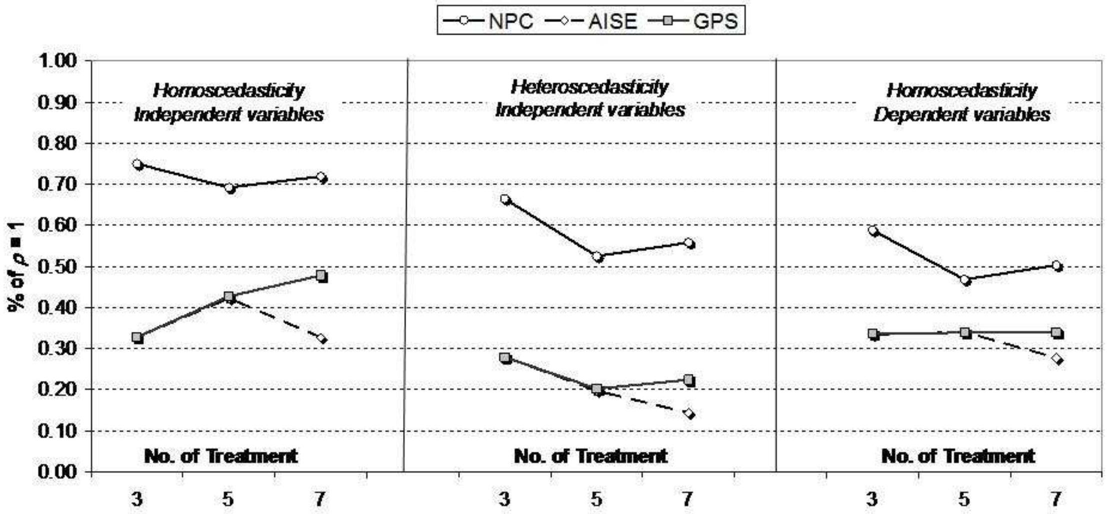


Figure 2.7: First simulation study: example of results (rate of of exact matching with the true ranking) in the settings with p=3, n=4, normal errors.

of near treatments do not intersect even if they are in reality different); method "0", the simplest because it is only the mean value of the mean values, seems to be always a little bit better than AISE. Therefore, the final ranking of the methods seems to be: 0 $\succ$ NPC $\succ\succ$ GPS $\succ$ AISE $\succ\succ$ 1[20];

- all methods get important benefits from an increased number of (informative) variables and a greater number of replications;

- asymmetry does not affect results almost at all, conversely heavy-tailed distributions reduce reliability; heteroscedasticity mainly reduces the performance of GPS and AISE, while dependency negatively affects NPC; in general, the most robust method is NPC.

[20] A $\succ\succ$ B indicates that A is much better than B.

2.4 Extension to multivariate RCB design

What we are introducing in this part was all implemented in a R code[21], but it was not possible to complete it, because simulations took really too much time[22]. Anyway we report briefly here the lying below idea, because maybe it is possible in the future to try to simplify the R code in some way and to obtain results in a useful time.

Let us consider the experimental design where there are n blocks and, within each block, experimental units are randomly assigned to the C ($C > 2$: $C = 2, 3, 4, 5$) treatments and exactly one experimental unit is assigned to each of the C treatments. Let Y be the multivariate variable related to a p-dimensional vector of responses (in our example p had to be equal to 3). The experimental design is developed with the aim of comparing the C treatments with respect to p different response variables. The statistical model (with fixed effects) for the multivariate Randomised Complete Block (RCB) design can be represented as follows:

$$Y_{ij} = \mu + \beta_i + \tau_j + \varepsilon_{ij}, \quad \varepsilon_{ij} \sim IID(0, \Sigma), \quad i = 1, \ldots, n; j = 1, \ldots, C. \tag{2.27}$$

where β_i, τ_j and Y_{ij}, are respectively the effect of the i-th block, the effect of the j-th treatment and the p-dimensional multivariate response variable for the i-th block and the j-th treatment. The random term ε_{ij} represents a p-vector of experimental errors with zero mean, variance-covariance matrix Σ and unknown continuous distribution P and the usual side-conditions for effects are given by the constrains $\sum_i \beta_i = \sum_j \tau_j = 0$. The model described in (2.27) is called "effect model"[23]. If we define $\mu_j = \mu + \tau_j, \quad j = 1, \ldots, C$, an alternative representation of the model is the so called "mean model", i.e.

$$Y_{ij} = \mu_j + \beta_i + \varepsilon_{ij}. \tag{2.28}$$

[21] See appendix A at page 67.

[22] We had to perform 16 settings and to obtain the results of only one of them we saw that it would take us more than one day!

[23] See [36].

The resulting inferential problem of interest is concerned with the following hypotheses:

1. $H_0: \ \tau_j = 0, \forall j$, against $H_1 : \exists j : \tau_j \neq 0$; note that this hypothesis is referred to a global test; if H_0 is rejected, it is of interest to perform inference on each pairwise comparison between couples of treatments, i.e.

2. $H_{0(jh)}: \ \tau_j = \tau_h, \ j, h = 1, \ldots, C, \ j \neq h$, against $H_{1(jh)}: \ \tau_j \neq \tau_h$; with reference to the second model, an equivalent representation of $H_{0(jh)}$ is the following: $H_{0(jh)}: \ \mu_j - \mu_h = 0, \ j, h = 1, \ldots, C; \ j \neq h$, against $H_{1(jh)}: \ \mu_j - \mu_h \neq 0$.

2.5 Application to industrial experiments

In this section we are dealing with the second simulation study we have eventually performed.
Let us consider the following real case study:

- the treatments of interest are 4 dosages (P1: 100%, P2: 95%, P3: 90%, P4: 85%) of a given detergent ($C = 4$); note that, a priori, we know the true ranking: $P1 \succ P2 \succ P3 \succ P4$;

- detergent performances are assessed by measuring the percentages of removed soil (so called reflectance) from a piece of fabric, previously soiled with 25 different soils ($p = 25$);

- the washing experiment has been replicated 24 times ($n = 24$), so we can extract random samples of different smaller sizes: 4, 8, 12, 16, 20;

- soils can be classified by their degree of importance (discrimination capability: 1, 2, 3) and by their main chemical properties (Bleachable, Enzymatic, General detergency): take a look at table 2.5 at page 46;

- since we grouped soils into 6 categories (1-5: all Bleachable soils of group 1, 6-9: all Enzymatic soils of group 1, 10-14: all General detergency soils of group 1, 1-14: all soils of group 1, 1-19: all soils of groups 1-2, 1-25: all soils (all soils of groups 1, 2 and 3)) the simulation settings were 42, because for each of the 6 group of variables we have performed 7 settings: 5 settings ($n = 4, 8, 12, 16, 20$) under $H_0 : C = 4$ (that we know it is true) and 2 settings under $H_1 : C = 8$ (that we know it is false).

The main idea of this second simulation study, is to apply the three methods of scientific interest (NPC, AISE and GPS, whereas "0" and "1" methods were used only in the first simulation study to use them as terms of comparisons for the other three methods and not for a real interest), beginning with real data and not only with invented ones, as in our first simulation study (see tables 2.6 and 2.7 at page 40). Furthermore, another very innotavive aspect of this second study, is that it is very interesting to try the methods also under a false hypothesis ($C = 8$).
In order to evaluate the goodness of the classifications of the ranking methods in the different settings, we have performed various indexes and calculations:

1. first of all for each setting we have three classification matrices (one for each method);

2. confidence intervals of the scores with percentiles method: after simulation are performed, then we put together all the obtained scores and then we take the 0.025 and the 0.975 quantiles (actually they are pseudo-confidence intervals);

3. confidence intervals of the rankings of the scores with percentiles method;

4. average Spearman's ρ: for every setting, for each of the 1000 simulations, Spearman's ρ correlation coefficient is calculated in order to compare the true classification and the classification (we measure their cograduation) performed by the simulation; the index of this third point is simply the mean of the Spearman's ρ's.;

Soil no.	Soil ID	Import.	Class
1	CFT CS-15	1	Bleachable
2	Empa 164	1	Bleachable
3	WFK 10J	1	Bleachable
4	WFK 10K	1	Bleachable
5	WFK 10LI	1	Bleachable
6	CFT CS-01	1	Enzymatic
7	Empa 111	1	Enzymatic
8	Empa 112	1	Enzymatic
9	WFK 10Z	1	Enzymatic
10	Empa 141	1	General detergency
11	Empa 143	1	General detergency
12	WFK 10C	1	General detergency
13	WFK 10D	1	General detergency
14	WFK 20D	1	General detergency
15	CFT CS-120	2	Bleachable
16	CFT CS-28	2	Enzymatic
17	CFT CH-021	2	General detergency
18	CFT CH-022	2	General detergency
19	WFK 10GM	2	General detergency
20	CFT RB-001	3	General detergency
21	CFT RB-005	3	General detergency
22	CFT RB-006	3	General detergency
23	CFT RB-007	3	General detergency
24	CFT RB-008	3	General detergency
25	CFT RB-009	3	General detergency

Table 2.8: Types of soils, classified by their importance and their main chemical properties.

5. percentage of simulations in which Spearman's ρ is equal to 1.

The first and the last ones are the only ones we used also in the first simulation study; here we decided to use also other methods, as you were able to

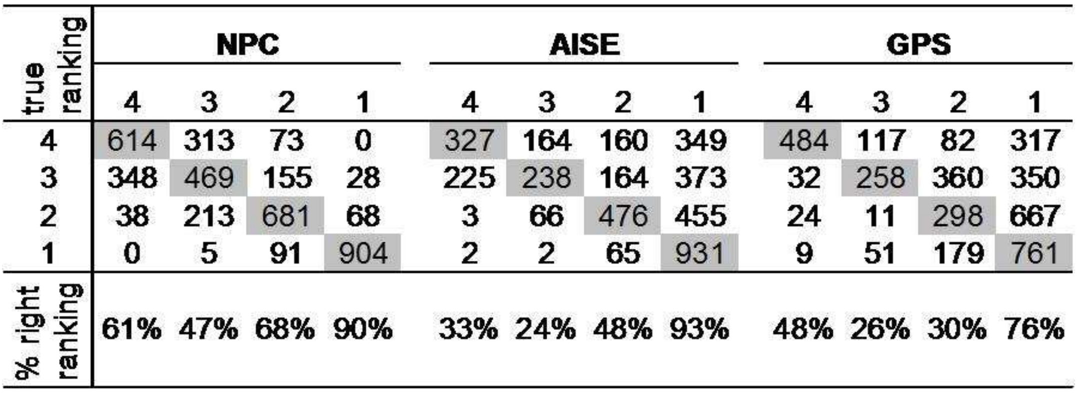
see from this list, in order to try to find better methods and ways to assess goodness of a classification. The fourth method (average Spearman's ρ) has a little drawback, because it is not always possible to calculate Spearman's ρ, for example when a certain method in a certain setting provides in one simulation a result in which all treatments have the same ranking[24], so when analyzing this index we must always remember this aspect and realize that this statistic is not so reliable in order to evaluate the classification; in particular we have seen that the number of cases in which it is not possible to calculate Spearman's ρ increases with the number of considered replications. Furthermore we have not this problem with NPC method, but only with AISE and GPS methods and this happens because these last two methods are constructed in a way that needs a larger sample difference in order to construct a score that can really reflect the reality. A main point to develop is just this: we will need in the future to improve these methods and to use an evaluation method based on the calculation of Expectation and Variance of the scores of the three methods (see subsubsection 2.2.2 and chapter 4 for some further details on this topic).
Analyzing results of simulations we have more or less the same indications of those got in the first simulation study (NPC is the best method, all methods work better with more treatments, with more replications and so on) and almost always all evaluating methods give the same results and gave us the same impressions, therefore in a certain sense we have no great surprises with this second simulation study. There are anyway some new aspects emerged from this second study and the main ones are synthesized in the following two points:

- the three ranking methods answer in a good way when we work under $H_1: \ C = 8$, knowing that it is false, because the number of simulations with Spearman's ρ equal to 1 is always 0, that is the three ranking methods work very good succeeding in recognizing that H_1 is false

[24] Software returns a message error when we ask it to calculate for example Spearman's ρ between these two vectors: $(1, 2, 3, 4, 5)$ and $(5, 5, 5, 5, 5)$.

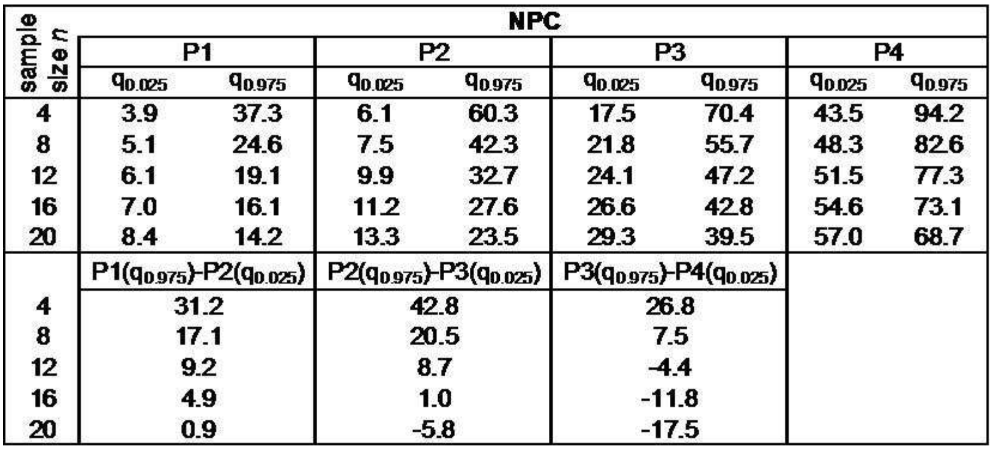
when it is so;

- the second point is a little more thorny: it happens, only with AISE and GPS, that these two methods get worse in terms of numbers of simulations with a perfect classification (percentage of cases on the 1000 simulations with $\rho = 1$) and this was very surprising for us.



true ranking	NPC				AISE				GPS			
	4	3	2	1	4	3	2	1	4	3	2	1
4	614	313	73	0	327	164	160	349	484	117	82	317
3	348	469	155	28	225	238	164	373	32	258	360	350
2	38	213	681	68	3	66	476	455	24	11	298	667
1	0	5	91	904	2	2	65	931	9	51	179	761
% right ranking	61%	47%	68%	90%	33%	24%	48%	93%	48%	26%	30%	76%

Figure 2.8: Second simulation study: example of results (classification matrices) in the setting with p=5, soils 10-14, n=4.



sample size n	NPC							
	P1		P2		P3		P4	
	$q_{0.025}$	$q_{0.975}$	$q_{0.025}$	$q_{0.975}$	$q_{0.025}$	$q_{0.975}$	$q_{0.025}$	$q_{0.975}$
4	3.9	37.3	6.1	60.3	17.5	70.4	43.5	94.2
8	5.1	24.6	7.5	42.3	21.8	55.7	48.3	82.6
12	6.1	19.1	9.9	32.7	24.1	47.2	51.5	77.3
16	7.0	16.1	11.2	27.6	26.6	42.8	54.6	73.1
20	8.4	14.2	13.3	23.5	29.3	39.5	57.0	68.7
	P1($q_{0.975}$)-P2($q_{0.025}$)		P2($q_{0.975}$)-P3($q_{0.025}$)		P3($q_{0.975}$)-P4($q_{0.025}$)			
4	31.2		42.8		26.8			
8	17.1		20.5		7.5			
12	9.2		8.7		-4.4			
16	4.9		1.0		-11.8			
20	0.9		-5.8		-17.5			

Figure 2.9: Second simulation study: example of results (quantiles of the simulated distribution of the scores) in the setting with p=3, n=4, normal errors for NPC method.

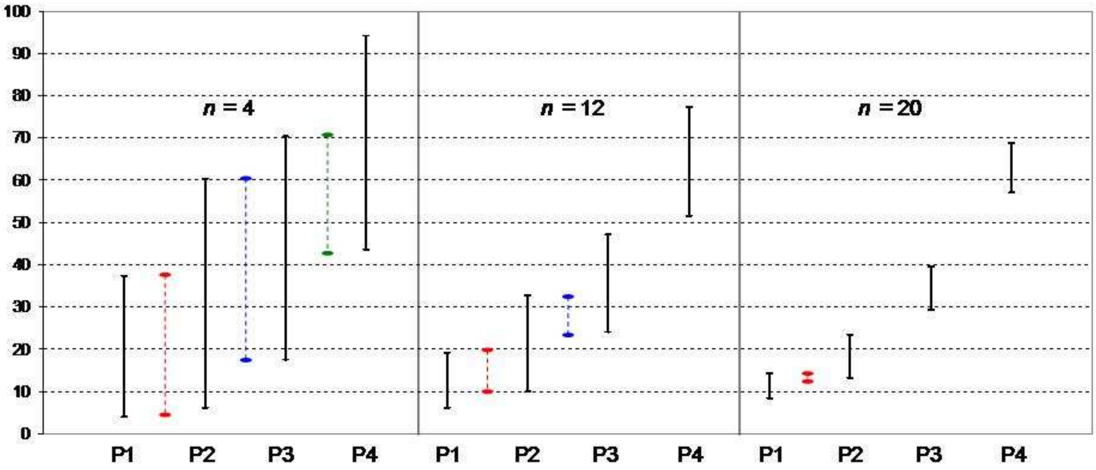


Figure 2.10: Second simulation study: example of results (graphic representation of pseudo-confidence intervals) in the settings with p=3, n=4, normal errors for NPC method.

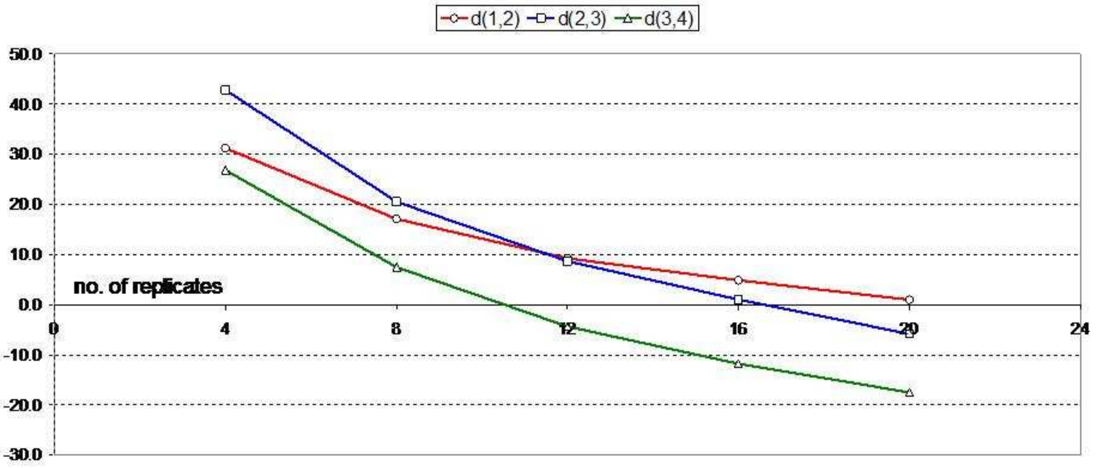


Figure 2.11: Second simulation study: example of results (a sort of way to estimate power) in the settings with p=3, n=4, normal errors for NPC method.

My explaination to the phenomenon illustrated in the second point of the previous list is this: if we take a better look at the results of simulation, we can see that the percentage of classifications with Spearman's ρ equal to 1 is the only evaluation index/statistic that gets worse by increasing the number of replications. We spent a whole day in order to understand this strange phenomenon and we had the idea of trying to perform a particular setting with all 24 replications (in this case one simulation is enough, because we sample (always without replacing!) all replications we have in hand) and we saw that by sampling all units, AISE and GPS cannot recognize some of the differences between treatments, while NPC can do it very well[25]. Therefore, if AISE and GPS cannot succeed in good working on the whole dataset, it happens that in samples of it, by increasing the number of replications and by coming nearer to the whole population this index of the two methods comes nearer and nearer to 0!

We have different proofs of this fact: first of all, all other statistics and indexes improve by increasing the number of replications (the average Spearman's ρ increases, confidence intervals become smaller and smaller) but this cannot happen on percentage of correct classifications, because this is only the number of the cases in which the classification is equal to the true one and not more and it is inadequate to describe the real trend of performances of methods; furthermore, we tried lots of times to increase the first type error (in all simulations α value is always 0.05) used for the multiple comparison in AISE and GPS and indeed we could see that by reaching high level of α, even AISE and GPS could recognize some differences that first they could not recognize (sometimes 0.20 was enough, sometimes even 0.60 or 0.70!).

We realize this is very difficult to understand, also because it was very difficult even for us to find and to explain, but we are really sure of what we are saying and we were also a little bit proud when we discovered it, also because first we thought that it was an error of the R code or of something else.

We can conclude that we have a further consistent proof that NPC is really

[25] This is because NPC is much more informative than the other two ranking methods and by using directly p-values and not using an intermediate significance score, as AISE and GPS do, NPC can point out also very small differences and it is almost impossible that two treatments with NPC are in ex-aequo, differently from the other two methods where this happens very usually.

the best ranking method, and that in particular it is very robust and can really recognize the difference between two treatments which are very close to each other; however, probably (we had just taken a look at it by making a brief test in R, while we were working on these things) it is possible that NPC risks to have the opposite drawback of AISE and GPS: probably when, in the truth classification, two treatments are completely equal, the inevitable sample differences are sufficient for NPC to recognize differences between treatments that in reality are not true.

Chapter 3

Multivariate performance indicators: theory, methods and application to new product development

3.1 Introduction and motivation

In the research and development of new products often the aim is focused on evaluating the product performances in connection with more than one aspect (dimension) and/or under several conditions (strata). In this framework the main goal of statistical data analysis consists in the calculation of an index to obtain a global performance evaluation of the products under investigation which is a synthesis of the information given by whole performance data.

The considered experimental design presents a multivariate response variable where the univariate components have different degrees of importance. In general, each dimension of the global performance should be evaluated under different conditions which can be represented by two or more strata, jointly considered. The methodological solution to cope with this problem is

described and applied, considering different possible data transformation.
Let us suppose that the global performance is represented by a variable η, that indicates a complex and underlying concept, often named *construct*, which is not directly measurable, hence it is broken into a set of measurable components, dimensions or items. In order to build up a global performance index, two main critical steps have to be taken into account: *standardization* and *aggregation*.
Standardization methods should take into account both the data properties and objectives of the analysis. Let $Y_1, Y_2, \ldots, Y_K$ be the informative variables representing the measurable components of η. Standardization of $Y_1, Y_2, \ldots, Y_K$ is a transformation that replaces each Y_k by a new variable $T_k(Y_k)$. The main reason for standardization is to allow for the comparability among variables; a review of the most commonly used transformations and an exploration of the main mathematical and statistical consequences of their application is proposed in [1].
After the transformation of non homogeneous data, it is necessary to put together the variables $T_k(Y_k)$ through an *aggregating function* $g(\cdot)$. Hence the *aggregation* allows us to obtain a global final variable which gives a measure of the *construct* or *latent variable* η:

$$Y = g\left[T_1(Y_1), T_2(Y_2), \ldots, T_K(Y_K); \omega_1, \omega_2, \ldots, \omega_K\right], \tag{3.1}$$

where $\omega_1, \ldots, \omega_K$ are the weights (degrees of importance) assigned to $Y_1, \ldots, Y_K$, respectively, which usually have an important impact on the aggregated values of the performance index. Although some weights could be negative, in general $\omega_k \geq 0$, $k = 1, 2, \ldots, K$ and $\sum_k \omega_k = 1$: from now on we will assume this condition unless a different assumption is explicitly done.
The most frequent aggregation functions proposed in literature[1] are based on additive methods and require assumptions about indicators and weights which are often not desirable and difficult to meet and to test[2]. For this

[1] See [18] for an extensive review.

[2] See [37].

reason other aggregation methods heve been proposed and among these, we mention multiplicative methods such as geometric aggregation and multi-criteria analysis.
In this chapter we are facing the problem of determining a comparative global performance evaluation of C products, summing up partial performance measures coming from multivariate experimental data in presence of multistratification. The complexity of the experimental design is due to the following aspects:

- the response variable is multivariate in nature and the univariate component variables present different degrees of importance;
- often one or more component variables represent primary performances, while the other ones represent secondary performances and two partial aggregated evaluations are at least needed along with the global evaluation;
- some experiments (in general those related to primary performances) allow replications hence, for some responses, comparative evaluations can be based on multiple comparisons of one-way ANOVA but, for economic or practical reasons, other responses are characterized by unreplicated designs, hence, for these variables, inferential procedures are not possible;
- each dimension of the global performance should be evaluated under different conditions which can be represented by two or more strata, jointly considered.

3.2 Composite indexes

A composite index has to measure a complex and underlying phenomenon Y which is not measurable but can be broken into K measurable components, dimensions or items. Data transformation procedure for the calculation of a composite indicator consists in a sequence of steps aimed to achieve comparability among component variables $Y_1, Y_2, \ldots, Y_K$ and to make a synthesis of the available information: the former purpose is obtained through *standardization*, the latter can be achieved through the application of an *aggregation* technique.

3.2.1 Standardization: data transformations to obtain homogeneous variables

Let us suppose that Y_{ck} represents the value of k-th variable for c-th unit (product): a possible standardization approach to have comparability is to rank each variable across units. For example, in the case of decreasing *rank transformation* we have:

$$T_k(Y_{ck}) = R(Y_{ck}) = \sum_{u=1}^{C} I_{uk}(Y_{ck}) + 1, \tag{3.2}$$

where

$$I_{uk}(x) = \begin{cases} 1 & \text{if } Y_{uk} > x \\ 0 & \text{otherwise} \end{cases} . \tag{3.3}$$

This typical non-linear transformation requires just simple calculations and is robust in presence of outliers; the main drawback is the loss of information related to the original metric. The evaluation of a unit based on a given variable consists in the position of the unit in the ranking based on that variable; relative rank $R(Y_{ck})/C$ can be preferable to $R(Y_{ck})$ because it takes values in the interval $[0, 1]$.

To avoid computational problems in the aggregation phase (i.e. null denominator, null argument of logarithm, etc.), relative rank $[R(Y_{ck}) + c_1]/(C + c_2)$ can be calculated, where c_1 and c_2 are constants such that the relative rank takes values in the open interval $(0, 1)$.

The traditional *standardization* method converts all original variables to variables with zero mean and standard deviation equal to one, applying the well-known transformation:

$$T_k(Y_{ck}) = \frac{Y_{ck} - \overline{Y}_k}{S_k}, \tag{3.4}$$

where $\overline{Y}_k$ is the sample mean and S_k the sample standard deviation of Y_k. The standardized variables present different ranges and this transformation is not robust with respect to outliers. Sometimes a similar linear transformation, with median instead of mean as a location measure and median absolute deviation instead of standard deviation as a variability measure, is used.

The *re-scaling* technique produces standardized variables with identical range $[0, 1]$:

$$T_k(Y_{ck}) = \frac{Y_{ck} - \min_u(Y_{uk})}{\max_u(Y_{uk}) - \min_u(Y_{uk})}. \tag{3.5}$$

Since this method uses range instead of standard deviation as denominator, outliers have a great effect on standardization.

Standardization can also be applied just comparing the original data with the maximum value, according to the following formula:

$$T_k(Y_{ck}) = \frac{Y_{ck}}{\max_u(Y_{uk})}. \tag{3.6}$$

In this case the standardized variables assume values in $[\frac{\min_u(Y_{uk})}{\max_u(Y_{uk})}, 1]$.

A similar transformation can be obtained just considering the minimum value, for instance:

$$T_k(Y_{ck}) = 1 - \frac{\min_u(Y_{uk})}{Y_{ck}}, \tag{3.7}$$

and the range of standardized variable is $[0, 1 - \frac{\min_u(Y_{uk})}{\max_u(Y_{uk})}]$. It is worth noting that the latter transformation is non-linear unlike standardization, based on maximum value and (3.5), which are linear.

When standardization is aimed at the *comparison with a reference unit* (or *with a target*), value one is given to the reference unit, i.e. $T_k(Y_k^*) = 1$, and transformed values are calculated through the following ratio:

$$T_k(Y_{ck}) = \frac{Y_{ck}}{Y_k^*}, \tag{3.8}$$

where Y_k^* indicates the value of Y_k corresponding to the reference unit or the target value for Y_k. With this method, typical of economic applications where all Y_{ck} are non-negative (e.g. index numbers), transformed data take value in $[0, \infty)$. Alternatively, when we are interested in gaps, the relative variations $T_k(Y_{ck}) = \frac{Y_{ck}-Y_k^*}{Y_k^*}$, taking values in $(-\infty, +\infty)$, can be calculated. A similar, but more robust, method of standardization distinguishes among values above, close to, or below a certain percentage threshold around the mean or a reference value:

$$T_k(Y_{ck}) = \begin{cases} -1 & \text{if} \quad Y_{ck} - m_k < -\delta_k|m_k| \\ a_k Y_{ck} & \text{if} \quad -\delta_k|m_k| \leq Y_{ck} - m_k \leq +\delta_k|m_k| \\ +1 & \text{if} \quad Y_{ck} - m_k > +\delta_k|m_k| \end{cases}, \tag{3.9}$$

where m_ks are the means or the reference values, δ_ks are the percentage thresholds and a_ks are non negative constants; as a special case some a_ks could be null. The drawbacks of this non-linear transformation are the arbitrariness of δ_k and the loss of the information about the original metric.

3.2.2 Aggregation: synthesis of information

The application of an aggregation technique consists in the choice of an appropriate function $g : \mathbb{R}^K \to \mathbb{R}$ to apply (3.1). The most used are additive techniques but they require assumptions which are often not desirable and sometimes difficult to meet and to verify[3], hence some authors propose alternative aggregation methods such as multiplicative (geometric) aggregations or non-compensatory aggregations (e.g. multi-criteria methods).

Additive aggregation is based on the weighted sum of standardized variables:

[3] See [37].

$$Y = \sum_{k=1}^{K} \omega_k T_k(Y_k). \tag{3.10}$$

There should be no conflict or synergy among standardized variables; in case of conflict, standardization should be used also to change direction of the original variables decreasingly related to the latent variable Y.

Additive aggregation is a fully compensatory approach because low values in some variables can be completely compensated by sufficiently high values in other variables. Assuming all T_k are positive, *geometric aggregation* presents less compensability because it is based on a multiplicative approach:

$$Y = \prod_{k=1}^{K} T_k(Y_k)^{\omega_k}. \tag{3.11}$$

If a geometric aggregation is applied to calculate a composite performance indicator, a unit under evaluation should prefer to increase partial indicators (variables) with low score than those with high score to improve its position in the global ranking.

The *multi-criteria approach* is based on a non-compensatory rationale; the main assumption is the comparability between units for each variable Y_k and the method consists in ordering the units after pairwise comparisons across the whole set of variables. Multi-criteria analysis allows us for considering jointly qualitative and numeric variables and in general it does not necessarily require standardization to assure comparability among variables. For each variable Y_k a *preference function* is defined, such that for each couple of units (u, v), it indicates if u is worse, equivalent or better than v $(u, v = 1, \dots, C)$. The preference function can be written in this way:

$$h_k(Y_{uk}, Y_{vk}) = \begin{cases} -1 & \text{if} \quad u \text{ is worse than } v \text{ according to } Y_k \\ 0 & \text{if} \quad u \text{ is equivalent to } v \text{ according to } Y_k \\ +1 & \text{if} \quad u \text{ is better than } v \text{ according to } Y_k \end{cases}. \tag{3.12}$$

The above general definition of the preference function can be applied to a wide range of functions and the choice about which function should be used depends on the decision-making problem and from the nature of Y_k, hence for each aspect (criterion) a specific preference function must be defined. The most common preference functions are the following:

- *subjective*: values +1, 0 and -1 are assigned according to judgements of experts;
- *dichotomic*: -1 is assigned if a requested characteristic or property is satisfied by v but not by u, 0 is assigned if both units or neither of them satisfy the characteristic/property and +1 is assigned otherwise;
- *ordinal*: the k-th preference function takes the value +1 if $Y_{uk} > Y_{vk}$, -1 if $Y_{uk} < Y_{vk}$ and 0 otherwise;
- *ϵ-ordinal*: the k-th preference function takes the value +1 if $Y_{uk} > Y_{vk} + \epsilon$, -1 if $Y_{uk} < Y_{vk} + \epsilon$ and 0 otherwise;
- *α-stochastic*: value +1 (-1) is given if the observed value of Y_{uk} is greater (less) than the observed value of Y_{vk} and if they are stochastically not equal (at significance level α); value 0 is given otherwise.

Hence, considering Y_k, for each unit a flow is computed according to:

$$T_k(Y_{ck}) = \Phi_c^{(k)} = \sum_{v=1}^{C} h_k(Y_{ck}, Y_{vk}). \tag{3.13}$$

The flow measures the degree of preference associated to each unit: a positive flow expresses how much the unit dominates the other ones and a negative flow indicates how much it is dominated by the other ones; based on these flows, K partial rankings of the C units are obtained. The global synthesis respect to the K aspects can be obtained through a weighted mean of flows (or of flow transformations which do not modify partial rankings). For a review of the main multi-criteria methods related to the construction of a composite index see [22].

3.3 Simulation study

The settings considered in this third simulation study do not take into account all we have said in this third chapter, but anyway also this simulation study offers some interesting cues; they are almost the same of the first study, except from the following points:

1. the number of treatments is 5 or 10 ($C = 5, 10$), no longer 3, 5 and 7;
2. two (and no longer three) types of random errors are considered: we have excluded the skew-normal distribution, because we had seen in the first simulation study that skewness was not an influence factor on the results;
3. we have not changed all other features of the simulations ($p = 3, 6$; $n = 4, 8$; always the same three variance-covariance random error settings);
4. notice that in this case we have 48 simulation settings ($2 \times 2 \times 2 \times 2 \times 3$).

Practically there is only a new aspect in this study with respect to the first study, that is we have added two new phases:

- aggregation of scores;
- standardization of scores.

Actually in the first study we had already used aggregation (it is not possible using no form of aggregation): in all three methods of interest (NPC, AISE and GPS) we had used always the sum of scores and for some of them we also had to use a sort of pre-aggregation (for example an intermediate sum for AISE and an intermediate Fisher's combining function for NPC); here we introduce three new types of little more sophisticated aggregations (always without weighting variables) for each of the three methods:

1. arithmetic mean;
2. geometric mean;

3. Fisher's combining function.

Furthermore, for each method, to each type of score aggregation we applied 11 types of standardization:

1. std. with the minimum value: $\frac{I}{\min(I)}$;
2. std. with the sample minimum value: $\frac{I}{I_{(1)}}$;
3. std. with the maximum value: $\frac{I}{\max(I)}$;
4. std. with the sample maximum value: $\frac{I}{I_{(n)}}$;
5. no standardization: I;
6. std. with mean and variance: $\frac{I-\bar{I}}{\text{sd}(I)}$;
7. inverse std.: $\frac{1}{I}$;
8. norm. with respect to the minimum value: $\frac{I-\min(I)}{\max(I)-\min(I)}$;
9. norm. with respect to the sample minimum value: $\frac{I-I_{(1)}}{I_{(n)}-I_{(1)}}$;
10. norm. with respect to the maximum value: $\frac{\max(I)-I}{\max(I)-\min(I)}$;
11. norm. with respect to the sample maximum value: $\frac{I_{(n)}-I}{I_{(n)}-I_{(1)}}$.

So, in the end, we obtain 3 × 11 = 33 new sub-methods, for each of the three ranking methods, for each of the 48 settings (actually: 32 sub-methods for NPC, 33 each for AISE and GPS, for each setting, because for NPC the combination between aggregation with Fisher's combining function and standardization with inverse standardization is not practicable). To each of the, therefore, 98 combinations are then applied some tricks[4] and simple

[4] For "tricks" we mean that in lots of these combinations we have some logistic problems: for example if the maximum theoretical value of an index is 0 we must take into account of this in the third standardization by dividing for a very very little number close to 0; it can also occur that in a simulation the sample maximum value is 0, so we have to prevent this potential problem by taking into account of it also in the fourth standardization (some

calculations, so that all these 98 combination are included in the interval $[0,1]$[5].

The main idea and our goal in this study is to find out the best combination of aggregation and standardization in function of the features of the setting and of the ranking method used. The evaluation methods are the same used also for the second simulation study (see section 2.5 at page 44).

The main new considerations that we can do analyzing results of third study, besides all indications we had from the previous two studies, that keep being true, and the main new indications that we can have are:

- as we could expect it, the type of standardization does not influence at all the goodness of the classification obtained with a certain method (NPC, AISE or GPS); in fact standardization was introduced above all to have comparable results (in particular all 98 indexes are included in the interval $[0,1]$);
- what is very important to differentiate the goodness of the classification, instead, is the type of aggregation: in particular we have noticed that in a great part of the results arithmetic mean behaves in a certain way and the other two aggregation types (geometric mean and Fisher's combining function) behaves similarly in a different way; in particular arithmetic mean seems to perform and to fit better if we use it with NPC method, whereas geometric mean and Fisher's combining function seem to perform and to fit better if we use them with AISE and GPS methods and these statements seem to be true for almost every

simulations could have no problem, some other ones not and often you do not realize it at once, but often you realize it when you see that the software in the simulations returns some errors). This is only a single example of what may happen, but we had to correct lots and lots of the 98 combinations and it took us a big part of our time to found all the problems (both evident and not evident ones) of these combinations.

[5] We are not explaining in detail what we have exactly done for each of the 98 combinations, because it would be really too long! If you want to try to understand what we have done in detail, try to take a look at the code of this simulation: see appendix A at page 67.

setting (so, type of variance-covariance matrix, type of distribution and so on seem to have not a great influence on these considerations).

After these three simulation studies it is clear that in the future we will have to better explore the results of simulations (see also discussion in chapter 4 about this topic), but we can anyway already for sure conclude some things:

- multivariate ranking methods are reliable tools to rank treatments within the experimental design framework;
- NPC method allows us to better rank the treatments than AISE and GPS methods;
- NPC method is able to "include" more useful information from experimental data than other methods;
- NPC method is also the most robust method, especially in case of heavy-tailed random distribution and heteroscedasticity;
- the proposed simulation-based approach allows us to properly design the suitable size of replications.

Chapter 4

Discussion and conclusions

Work and commitment lying below this thesis to carry it out were really a lot, more than the reader can imagine (it is sufficient for example that you take a look at the length of R codes used for simulation studies or at the huge number of bibliographical entries). But unfortunately even working so hard, it was not enough to perform all we wanted to do; we are listing these topics so that in the future it will possible to develop them[1].

- It would be good to perform the two parallel analyses of chapter 1 in order to be able to compare them and to better understand if the analysis of management literature is really not correct at all, or if it is possible to use it even if it is not statistically correct; it would be even more beautiful to approach this situation with simulations, in order to be a little more certain of the results. Anyway it is very long to plan and sure it will take us really a lot of time.
- The second important thing that we should better develop in the future is to find a better way to evaluate results of simulations, that are performances of ranking methods of interest; therefore, we started to try this method and the main point is to try to find a way to calculate

[1] Anyway we will surely have to develop almost all of them after my graduation.

Expectation (E) and Variance (Var) of the scores of the methods of interest, in order to calculate some confidence intervals. In reality the way to begin was already found and illustrated in subsubsection 2.2.2, but, as we have already said in that section, we need some more time to better develop these ideas and unfortunately there was no time enough to complete totally these calculations in this work, therefore additional further methodological research is needed to develop formal confidence intervals.

- We are in possession of lots of results from the three simulation studies (results that can be even improved by developing the previous point) and we could exploit them better by performing other analyses, graphics, considerations and so on, that could allow us to discover other interesting aspects of the methods, because probably (surely, indeed) we have not found out everything about them yet; furthermore simulations have to be extended by comparing different methods to perform inference (parametric vs. non-parametric).

- Furthermore we should better develop the aspect of the extension of the work to multivariate RCB design, as we have already said in the relative section, because this one is probably very interesting and there are still lots of things and pieces of information to discover about this topic.

Appendix A

R codes used for the analyses

We wanted to report here in this work the R codes, so that if the reader knows the language R, he can take the code here and try to improve it; we decided to report the codes here, also because they were a great part of our work and they took us a great part of the time used to perform this work, also because we had to update the codes lots and lots of times and also simulations were runned more than one time, because there has been always something to adjust or update.

A.1 R code used for the first simulation study

```
library(mnormt)
library(mvtnorm)
library(sn)
source("pairwise_beta.r")
source("pairwise_gamma.txt")
source("score.r")

mu7<-read.csv("mu.csv",header=FALSE)
mu7<-matrix(unlist(mu7),ncol=6) # medie delle 6 variabili con 7 trattamenti
S2<-read.csv("sigma.csv",header=FALSE)
```

```
S2<-matrix(unlist(S2),ncol=6) # sigma etero-indip (6 variabili)
S1<-matrix(c(rep(c(1,rep(0,6)),5),1),ncol=6) # sigma omo-indip (6 variabili)
S3<-read.table("sigma3.txt",header=FALSE)
S3<-matrix(unlist(S3),ncol=6) # sigma omo-dip (6 variabili)
mu7b<-mu7[,1:3] # medie delle 3 variabili con 7 trattamenti
S1b<-S1[1:3,1:3] # sigma omo-indip (3 variabili)
S2b<-S2[1:3,1:3] # sigma etero-indip (3 variabili)
S3b<-S3[1:3,1:3] # sigma omo-dip (3 variabili)
mu5<-mu7[2:6,] # medie delle 6 variabili con 5 trattamenti
mu3<-mu7[3:5,] # medie delle 6 variabili con 3 trattamenti
mu5b<-mu7b[2:6,] # medie delle 3 variabili con 5 trattamenti
mu3b<-mu7b[3:5,] # medie delle 3 variabili con 3 trattamenti

simula.settings<-function(mu=mu7,sigma=S1,distr=c("norm","t2","nasimm")
,n=8,alpha=0.05,B=1000){

C=dim(mu)[1] #n° trattamenti
p=dim(mu)[2] #n,° variabili
Sc<-array(0,dim=c(B,C,p))
Sc_mean<-array(0,dim=c(B,C,p))
Grad_0<-array(0,dim=c(C,B)) # matriciona di risultati per media medie
Grad_1<-array(0,dim=c(C,B)) # matriciona di risultati per int. conf. dist.
Grad_mean<-array(0,dim=c(C,B))  # matriciona di risultati per la media
Grad<-array(0,dim=c(C,B))  # matriciona di risultati per NPC rank
Grad_gps<-array(0,dim=c(C,B)) # matriciona di risultati per GPS
SP_0<-array(0,dim=c(B,1))  # vettore di indici di Spearman per media medie
SP_1<-array(0,dim=c(B,1))  # vettore di indici di Spearman per int. conf. dist.
SP<-array(0,dim=c(B,1))   # vettore di indici di Spearman per NCP rank
SP_mean<-array(0,dim=c(B,1))    # vettore di indici di Spearman per media
SP_gps<-array(0,dim=c(B,1))   # vettore di indici di Spearman per GPS
```

```
for(cc in 1:B){

X<-array(0,dim=c(C,p,n))  # inizializza la matrice di dati
for(i in 1:C){
if (distr=="norm") X[i,,]<-rmvnorm(n,mean=mu[i,],sigma=sigma)
if (distr=="t2") X[i,,]<-rmvt(n, sigma = sigma, df = 2)+mu[i,]
if (distr=="nasimm") X[i,,]<-rmsn(n, xi=mu[i,], Omega=sigma, alpha=rep(5,p))
}
X<-matrix(X,ncol=p,byrow=FALSE)
label<-rep(seq(1,C),n)
Y<- data.frame(label,X)
Z<-data.frame(Y,double(n*C))
colnames(Z)[dim(Z)[2]]<-"dist"
Z[,dim(Z)[2]]<-apply(Y[,-1],1,function(x){sqrt(sum((100-x)^2))})
m<-array(0,dim=c(C,p))

for(j in 1:p){
for(i in 1:C){
m[i,j]<-mean(X[label==i,j])
}
}

P<-array(1,dim=c(C,C,p))
D<-array(0,dim=c(C,C,p))
MSD<-array(0,dim=c(p,1))
P_mean<-array(1,dim=c(C,C,p))
P_gps<-array(1,dim=c(C+1,C,p))
D_mean<-array(0,dim=c(C,C,p))
D_1<-array(0,dim=c(C+1,C))
for(j in 1:p){
MSD[j]<-unlist(summary(aov(Y[,(j+1)]~as.factor(Y$label),data=Y)))[6]
```

```
}
MSD<-sqrt(MSD)
MSDdist<-unlist(summary(aov(Z[,dim(Z)[2]]~as.factor(Z$label),data=Z)))[6]
MSDdist<-sqrt(MSDdist)
mdist<-tapply(Z$dist,Z$label,mean)
mm<-t(m)
colnames(mm)<-seq(1:C)

for(j in 1:p){
P[,,j]=pairwise(m,MSD,k=j,n,alt="greater",correct=FALSE)$P
D[,,j]=pairwise(m,MSD,k=j,n,alt="greater",correct=FALSE)$D
P_mean[,,j]=pairwise(m,MSD,k=j,n,alt="two.sided",correct=TRUE)$P
a=ord.pairwise(mm[j,],MSD[j],n,alt="two.sided",correct=TRUE)$P
b<-as.numeric(colnames(a))
P_gps[,,j]<-rbind(a,b)
}
pp1<-P_gps
for (i in 1:C){
for (j in 1:C){
for (k in 1:p){
if (pp1[i,j,k]>=alpha) pp1[i,j,k]=0
else pp1[i,j,k]=1
}}}
pp<-pp1
for (k in 1:p){
for (j in 1:C){
for (i in j:C){
pp[i,j,k]<-0
}}}
d<-matrix(0,p*2,C)
for (k in 1:p){
```

```
d[2*k-1,]<-score.inv(pp[-(C+1),,k])
d[2*k,]<-pp[C+1,,k]
}
g<-matrix(0,p,C)
for(k in 1:p){
ii<-order(d[2*k,],d[2*k-1,])
g[k,]<-(d[2*k-1,])[ii]
}
e<-apply(g,2,sum)

alpha1<-alpha/choose(C,2)
ordmdist<-sort(mdist,decreasing=T) # si ordina rispetto alle distanze
l1<-ordmdist-qt(1-alpha1/2,n*C-C)*MSDdist
l2<-ordmdist+qt(1-alpha1/2,n*C-C)*MSDdist

for(i in 1:(C-1)){
for(j in (i+1):C){
if ((l2[j]<l2[i])&(l2[j]>l1[i])) {D_1[i,j]=0}
else {D_1[i,j]=1}
}
}
D_1[C+1,]<-as.numeric(names(ordmdist))
d_int<-matrix(0,2,C)
d_int[1,]<-score.inv(D_1[-(C+1),])
d_int[2,]<-D_1[C+1,]
g_int<-double(C)
ii<-order(d_int[2,],d_int[1,])
g_int<-(d_int[1,])[ii]

D_mean<-apply(ifelse(P_mean<alpha,sign(D)*1,0),c(1,3),sum)
Sc_mean[cc,,]<-D_mean # da utilizzare se si vuole il punteggio originale
```

```
Grad_mean[,cc]<-rank(apply(Sc_mean[cc,,],1,sum),ties.method = "max")
Grad_gps[,cc]<-rank(e,ties.method = "max")
Sc[cc,,]<-apply(P,c(1,3),function(x){-2*sum(log(x))})
lambda<-apply(Sc[cc,,],1,sum)
Grad[,cc]<-rank(lambda,ties.method = "max") #graduatoria finale secondo Fisher
mmm<-apply(m,1,mean)
Grad_0[,cc]<-rank(mmm,ties.method = "max")
Grad_1[,cc]<-rank(-g_int,ties.method = "max")
SP_0[cc,]<-cor.test(Grad_0[,cc],seq(C,1),method = "spearman")$estimate
SP_1[cc,]<-cor.test(Grad_1[,cc],seq(C,1),method = "spearman")$estimate
SP[cc,]<-cor.test(Grad[,cc],seq(C,1),method = "spearman")$estimate
SP_mean[cc,]<-cor.test(Grad_mean[,cc],seq(C,1),method = "spearman")$estimate
SP_gps[cc,]<-cor.test(Grad_gps[,cc],seq(C,1),method = "spearman")$estimate
} #fine generazione dei dati

truth<-matrix(rep(seq(C,1),B),ncol=B,byrow=FALSE)
t0<-table(truth,Grad_0)
t00<-table(truth,Grad_1)
t1<-table(truth,Grad)
t2<-table(truth,Grad_mean)
t3<-table(truth,Grad_gps)
SP0<-table(SP_0[SP_0>0.9999])/B
SP00<-table(SP_1[SP_1>0.9999])/B
SP1<-table(SP[SP>0.9999])/B
SP2<-table(SP_mean[SP_mean>0.9999])/B
SP3<-table(SP_gps[SP_gps>0.9999])/B
t<-list(t0,t00,t1,t2,t3,SP0,SP00,SP1,SP2,SP3)
names(t)<-c("0","1","NPC","Media","GPS","Sp0","Sp1","SpNPC","SpMedia","SpGPS")
t
}
```

```
# Simulazioni:
set.seed(123)
a<-simula.settings(mu=mu3b,sigma=S1b,n=4)
write.table(a$"0","a5.txt",sep="\t",row.names=F,col.names=F)
write.table(a$"1","a6.txt",sep="\t",row.names=F,col.names=F)
write.table(a$NPC,"a1.txt",sep="\t",row.names=F,col.names=F)
write.table(a$Media,"a2.txt",sep="\t",row.names=F,col.names=F)
write.table(a$GPS,"a3.txt",sep="\t",row.names=F,col.names=F)
write.table(c(a$Sp0,a$Sp1,a$SpNPC,a$SpMedia,a$SpGPS),"a4.txt",sep="\t",
row.names=F,col.names=F)
set.seed(123)
b<-simula.settings(mu=mu3b,sigma=S1b,n=8)
write.table(b$"0","b5.txt",sep="\t",row.names=F,col.names=F)
write.table(b$"1","b6.txt",sep="\t",row.names=F,col.names=F)
write.table(b$NPC,"b1.txt",sep="\t",row.names=F,col.names=F)
write.table(b$Media,"b2.txt",sep="\t",row.names=F,col.names=F)
write.table(b$GPS,"b3.txt",sep="\t",row.names=F,col.names=F)
write.table(c(b$Sp0,b$Sp1,b$SpNPC,b$SpMedia,b$SpGPS),"b4.txt",sep="\t",
row.names=F,col.names=F)
set.seed(123)
c<-simula.settings(mu=mu5b,sigma=S1b,n=4)
write.table(c$"0","c5.txt",sep="\t",row.names=F,col.names=F)
write.table(c$"1","c6.txt",sep="\t",row.names=F,col.names=F)
write.table(c$NPC,"c1.txt",sep="\t",row.names=F,col.names=F)
write.table(c$Media,"c2.txt",sep="\t",row.names=F,col.names=F)
write.table(c$GPS,"c3.txt",sep="\t",row.names=F,col.names=F)
write.table(c(c$Sp0,c$Sp1,c$SpNPC,c$SpMedia,c$SpGPS),"c4.txt",sep="\t",
row.names=F,col.names=F)
...
(108 different simulations, therefore other 105)
```

A.2 R code used for the second simulation study

```
library(mnormt)
library(sn)
source("pairwise_beta.r")
source("pairwise_gamma.txt")
source("score.r")

dati<-read.table("Dataset.txt",header=T,row.names=1)
dati1<-matrix(0,96,27)
dati1<-as.data.frame(dati1)
dati1[,1:2]<-dati[1:96,1:2]
for (i in 1:25){
dati1[,i+2]<-dati[(96*i-95):(96*i),5]
}
colnames(dati1)<-c(c("Product","Type"),
c("CFT-RB-001","CFT-RB-009","CFT-RB-005","CFT-RB-006","CFT-RB-007",
  "CFT-RB-008","WFK-10J","CFT-CH022","CFT-CH021","WFK-10K","CFT-CS-15",
  "WFK-10D","WFK-10Z","CFT-CS-01","Empa-143","WFK-20D","WFK-10LI",
  "Empa-141","WFK-10GM","WFK-10C","empa-164","CFT-CS-28","Empa-111",
  "Empa-112","CFT-CS-120"))
dati<-dati1
set25_123<-seq(1:25)
set19_12<-set25_123[-c(1:6)]
set14_1<-set25_123[-c(1:6,8,9,19,22,25)]
set5_1gd<-c(12,15,16,18,20)
set5_1b<-c(7,10,11,17,21)
set4_1e<-c(13,14,23,24)
dat1=dati[,c(1:2,set25_123+2)]
dat2=dati[,c(1:2,set19_12+2)]
```

```
dat3=dati[,c(1:2,set14_1+2)]
dat4=dati[,c(1:2,set5_1gd+2)]
dat5=dati[,c(1:2,set5_1b+2)]
dat6=dati[,c(1:2,set4_1e+2)]

simula2.settings<-function(C=4,n=4,setting=dat1,alpha=0.05,B=1000){

p=dim(setting)[2]-2 # n° variabili
if (C==4) kk<-24
if (C==8) kk<-12
Sc<-array(0,dim=c(B,C,p))
Sc_mean<-array(0,dim=c(B,C,p))
Grad_mean<-array(0,dim=c(C,B))  # matriciona di risultati per la media
Grad<-array(0,dim=c(C,B))  # matriciona di risultati per NPC rank
Grad_gps<-array(0,dim=c(C,B)) # matriciona di risultati per GPS
Score_mean<-array(0,dim=c(C,B))
Score<-array(0,dim=c(C,B))
Score_gps<-array(0,dim=c(C,B))
SP<-array(0,dim=c(B,1))   #vettore di indici di Spearman per NCP rank
SP_mean<-array(0,dim=c(B,1))    #vettore di indici di Spearman per media
SP_gps<-array(0,dim=c(B,1))   # vettore di indici di Spearman per GPS

for(cc in 1:B){

X<-array(0,dim=c(n,p,C))
X<-matrix(X,ncol=p,byrow=FALSE)
X<-as.data.frame(X)
s<-sample(1:kk,n)
for (i in 1:n){
for (k in 1:C){
X[n*(k-1)+i,]<-setting[(k-1)*kk+s[i],3:(p+2)]
}
}
colnames(X)<-colnames(setting)[3:(p+2)]
label<-double(n*C)
for (i in 1:C){
label[((i-1)*n+1):(n*i)]<-rep(i,n)
}
X<- data.frame(label,X)
m<-array(0,dim=c(C,p))

for(j in 1:p){
for(i in 1:C){
m[i,j]<-mean(X[label==i,(j+1)])
}
}

P<-array(1,dim=c(C,C,p))
D<-array(0,dim=c(C,C,p))
MSD<-array(0,dim=c(p,1))
P_mean<-array(1,dim=c(C,C,p))
P_gps<-array(1,dim=c(C+1,C,p))
D_mean<-array(0,dim=c(C,C,p))

for(j in 1:p){
MSD[j]<-unlist(summary(aov(X[,(j+1)]~as.factor(X$label),data=X)))[6]
}
MSD<-sqrt(MSD)
mm<-t(m)
colnames(mm)<-seq(1:C)

for(j in 1:p){
```

```
P[,,j]=pairwise(m,MSD,k=j,n,alt="greater",correct=FALSE)$P
D[,,j]=pairwise(m,MSD,k=j,n,alt="greater",correct=FALSE)$D
P_mean[,,j]=pairwise(m,MSD,k=j,n,alt="two.sided",correct=TRUE)$P
a=ord.pairwise(mm[j,],MSD[j],n,alt="two.sided",correct=TRUE)$P
b<-as.numeric(colnames(a))
P_gps[,,j]<-rbind(a,b)
}
pp1<-P_gps
for (i in 1:C){
for (j in 1:C){
for (k in 1:p){
if (pp1[i,j,k]>=alpha) pp1[i,j,k]=0
else pp1[i,j,k]=1
}}}
pp<-pp1
for (k in 1:p){
for (j in 1:C){
for (i in j:C){
pp[i,j,k]<-0
}}}
d<-matrix(0,p*2,C)
for (k in 1:p){
d[2*k-1,]<-score.inv(pp[-(C+1),,k])
d[2*k,]<-pp[C+1,,k]
}
g<-matrix(0,p,C)
for(k in 1:p){
ii<-order(d[2*k,],d[2*k-1,])
g[k,]<-(d[2*k-1,])[ii]
}
e<-apply(g,2,sum)
```

```
D_mean<-apply(ifelse(P_mean<alpha,sign(D)*1,0),c(1,3),sum)
Sc_mean[cc,,]<-D_mean
lam<-apply(Sc_mean[cc,,],1,sum)
Score_mean[,cc]<-lam
Grad_mean[,cc]<-rank(lam,ties.method = "max")
Score_gps[,cc]<-e
Grad_gps[,cc]<-rank(e,ties.method = "max")
Sc[cc,,]<-apply(P,c(1,3),function(x){-2*sum(log(x))})
lambda<-apply(Sc[cc,,],1,sum)
Score[,cc]<-lambda
Grad[,cc]<-rank(lambda,ties.method = "max") #graduatoria finale secondo Fisher

if (C==4) SP[cc,]<-cor.test(Grad[,cc],seq(C,1),
method = "spearman")$estimate
if (C==8) SP[cc,]<-cor.test(Grad[,cc],c(8,8,6,6,4,4,2,2),
method = "spearman")$estimate
if (C==4) SP_mean[cc,]<-cor.test(Grad_mean[,cc],seq(C,1),
method = "spearman")$estimate
if (C==8) SP_mean[cc,]<-cor.test(Grad_mean[,cc],c(8,8,6,6,4,4,2,2),
method = "spearman")$estimate
if (C==4) SP_gps[cc,]<-cor.test(Grad_gps[,cc],seq(C,1),
method = "spearman")$estimate
if (C==8) SP_gps[cc,]<-cor.test(Grad_gps[,cc],c(8,8,6,6,4,4,2,2),
method = "spearman")$estimate
} #fine generazione dei dati

if (C==4) truth<-matrix(rep(seq(C,1),B),ncol=B,byrow=FALSE)
if (C==8) truth<-matrix(rep(c("8A","8B","6A","6B","4A","4B","2A","2B"),B),
ncol=B,byrow=FALSE)
t1<-table(truth,Grad)
```

```
t2<-table(truth,Grad_mean)
t3<-table(truth,Grad_gps)
SP1<-table(SP[SP>0.9999])/B
SP2<-table(SP_mean[SP_mean>0.9999])/B
SP3<-table(SP_gps[SP_gps>0.9999])/B
SP11<-c(mean(SP,na.rm=T),B-sum(table(SP)))
SP22<-c(mean(SP_mean,na.rm=T),B-sum(table(SP_mean)))
SP33<-c(mean(SP_gps,na.rm=T),B-sum(table(SP_gps)))
Int1<-matrix(0,C,3)
Int1<-as.data.frame(Int1)
colnames(Int1)<-c("q 0.025","q 0.975","scarto")
rownames(Int1)<-1:C
for (i in 1:C){
Int1[i,1]<-quantile(Grad[i,],0.025)
Int1[i,2]<-quantile(Grad[i,],0.975)
Int1[i,3]<-Int1[i,2]-Int1[i,1]
}
Int2<-matrix(0,C,3)
Int2<-as.data.frame(Int2)
colnames(Int2)<-c("q 0.025","q 0.975","scarto")
rownames(Int2)<-1:C
for (i in 1:C){
Int2[i,1]<-quantile(Grad_mean[i,],0.025)
Int2[i,2]<-quantile(Grad_mean[i,],0.975)
Int2[i,3]<-Int2[i,2]-Int2[i,1]
}
Int3<-matrix(0,C,3)
Int3<-as.data.frame(Int3)
colnames(Int3)<-c("q 0.025","q 0.975","scarto")
rownames(Int3)<-1:C
for (i in 1:C){
```

```
Int3[i,1]<-quantile(Grad_gps[i,],0.025)
Int3[i,2]<-quantile(Grad_gps[i,],0.975)
Int3[i,3]<-Int3[i,2]-Int3[i,1]
}
Int11<-matrix(0,C,3)
Int11<-as.data.frame(Int11)
colnames(Int11)<-c("q 0.025","q 0.975","scarto")
rownames(Int11)<-1:C
for (i in 1:C){
Int11[i,1]<-quantile(Score[i,],0.025)
Int11[i,2]<-quantile(Score[i,],0.975)
Int11[i,3]<-Int11[i,2]-Int11[i,1]
}
Int22<-matrix(0,C,3)
Int22<-as.data.frame(Int22)
colnames(Int22)<-c("q 0.025","q 0.975","scarto")
rownames(Int22)<-1:C
for (i in 1:C){
Int22[i,1]<-quantile(Score_mean[i,],0.025)
Int22[i,2]<-quantile(Score_mean[i,],0.975)
Int22[i,3]<-Int22[i,2]-Int22[i,1]
}
Int33<-matrix(0,C,3)
Int33<-as.data.frame(Int33)
colnames(Int33)<-c("q 0.025","q 0.975","scarto")
rownames(Int33)<-1:C
for (i in 1:C){
Int33[i,1]<-quantile(Score_gps[i,],0.025)
Int33[i,2]<-quantile(Score_gps[i,],0.975)
Int33[i,3]<-Int33[i,2]-Int33[i,1]
}
```

```
t<-list(t1,t2,t3,SP1,SP2,SP3,SP11,SP22,SP33,Int1,Int2,Int3,Int11,Int22,Int33)
names(t)<-c("NPC","Media","GPS","SpNPC","SpMedia","SpGPS","SpmNPC","SpmMedia",
"SpmGPS","IntNPC","IntMedia","IntGPS","IntsNPC","IntsMedia","IntsGPS")
t
}

# Simulazioni:
set.seed(123)
a<-simula2.settings(C=4,n=4,setting=dat1)
write.table(a$NPC,"a1.txt",sep="\t",row.names=F,col.names=F)
write.table(a$Media,"a2.txt",sep="\t",row.names=F,col.names=F)
write.table(a$GPS,"a3.txt",sep="\t",row.names=F,col.names=F)
write.table(c(a$SpNPC,a$SpMedia,a$SpGPS),"a4.txt",sep="\t",row.names=F,
col.names=F)
write.table(a$IntNPC,"a5.txt",sep="\t",row.names=F,col.names=F)
write.table(a$IntMedia,"a6.txt",sep="\t",row.names=F,col.names=F)
write.table(a$IntGPS,"a7.txt",sep="\t",row.names=F,col.names=F)
write.table(c(a$SpmNPC,a$SpmMedia,a$SpmGPS),"a8.txt",sep="\t",row.names=F,
col.names=F)
write.table(a$IntsNPC,"a9.txt",sep="\t",row.names=F,col.names=F)
write.table(a$IntsMedia,"a10.txt",sep="\t",row.names=F,col.names=F)
write.table(a$IntsGPS,"a11.txt",sep="\t",row.names=F,col.names=F)
set.seed(123)
b<-simula2.settings(C=4,n=8,setting=dat1)
write.table(b$NPC,"b1.txt",sep="\t",row.names=F,col.names=F)
write.table(b$Media,"b2.txt",sep="\t",row.names=F,col.names=F)
write.table(b$GPS,"b3.txt",sep="\t",row.names=F,col.names=F)
write.table(c(b$SpNPC,b$SpMedia,b$SpGPS),"b4.txt",sep="\t",row.names=F,
col.names=F)
write.table(b$IntNPC,"b5.txt",sep="\t",row.names=F,col.names=F)
write.table(b$IntMedia,"b6.txt",sep="\t",row.names=F,col.names=F)
write.table(b$IntGPS,"b7.txt",sep="\t",row.names=F,col.names=F)
write.table(c(b$SpmNPC,b$SpmMedia,b$SpmGPS),"b8.txt",sep="\t",row.names=F,
col.names=F)
write.table(b$IntsNPC,"b9.txt",sep="\t",row.names=F,col.names=F)
write.table(b$IntsMedia,"b10.txt",sep="\t",row.names=F,col.names=F)
write.table(b$IntsGPS,"b11.txt",sep="\t",row.names=F,col.names=F)
set.seed(123)
c<-simula2.settings(C=4,n=12,setting=dat1)
write.table(c$NPC,"c1.txt",sep="\t",row.names=F,col.names=F)
write.table(c$Media,"c2.txt",sep="\t",row.names=F,col.names=F)
write.table(c$GPS,"c3.txt",sep="\t",row.names=F,col.names=F)
write.table(c(c$SpNPC,c$SpMedia,c$SpGPS),"c4.txt",sep="\t",row.names=F,
col.names=F)
write.table(c$IntNPC,"c5.txt",sep="\t",row.names=F,col.names=F)
write.table(c$IntMedia,"c6.txt",sep="\t",row.names=F,col.names=F)
write.table(c$IntGPS,"c7.txt",sep="\t",row.names=F,col.names=F)
write.table(c(c$SpmNPC,c$SpmMedia,c$SpmGPS),"c8.txt",sep="\t",row.names=F,
col.names=F)
write.table(c$IntsNPC,"c9.txt",sep="\t",row.names=F,col.names=F)
write.table(c$IntsMedia,"c10.txt",sep="\t",row.names=F,col.names=F)
write.table(c$IntsGPS,"c11.txt",sep="\t",row.names=F,col.names=F)
...
(42 different simulations, therefore other 39)
```

A.3 R code used for the third simulation study

```
library(mnormt)
library(mvtnorm)
source("pairwise_beta.r")
source("pairwise_gamma.txt")
source("score.r")

mu10<-read.csv("mu10.csv",header=FALSE)
mu10<-matrix(unlist(mu10),ncol=6) # medie delle 6 variabili con 10 trattamenti
S2<-read.csv("sigma.csv",header=FALSE)
S2<-matrix(unlist(S2),ncol=6) # sigma etero-indip (6 variabili)
S1<-matrix(c(rep(c(1,rep(0,6)),5),1),ncol=6) # sigma omo-indip (6 variabili)
S3<-read.table("sigma3.txt",header=FALSE)
S3<-matrix(unlist(S3),ncol=6) # sigma omo-dip (6 variabili)
mu10b<-mu10[,1:3] # medie delle 3 variabili con 10 trattamenti
S1b<-S1[1:3,1:3] # sigma omo-indip (3 variabili)
S2b<-S2[1:3,1:3] # sigma etero-indip (3 variabili)
S3b<-S3[1:3,1:3] # sigma omo-dip (3 variabili)
mu5<-mu10[3:7,] # medie delle 6 variabili con 5 trattamenti
mu5b<-mu10b[3:7,] # medie delle 3 variabili con 5 trattamenti

simula3.settings<-function(mu=mu10,sigma=S1,distr=c("norm","t2"),n=8,
alpha=0.05,B=1000){

C=dim(mu)[1] #n° trattamenti
p=dim(mu)[2] #n,° variabili
Sc<-array(0,dim=c(B,C,p))
Sc_mean<-array(0,dim=c(B,C,p))
Score_mean<-array(0,dim=c(C,B,33))  # matriciona di risultati per la media
Grad_mean<-array(0,dim=c(C,B))
Score<-array(0,dim=c(C,B,33))  # matriciona di risultati per NPC rank
Grad<-array(0,dim=c(C,B))
Score_gps<-array(0,dim=c(C,B,33)) # matriciona di risultati per GPS
Grad_gps<-array(0,dim=c(C,B))
SP<-array(0,dim=c(B,1,33))   #vettore di indici di Spearman per NCP rank
SP_mean<-array(0,dim=c(B,1,33))    #vettore di indici di Spearman per media
```

```
SP_gps<-array(0,dim=c(B,1,33))   # vettore di indici di Spearman per GPS

for(cc in 1:B){

X<-array(0,dim=c(C,p,n))  # inizializza la matrice di dati
for(i in 1:C){
if (distr=="norm") X[i,,]<-rmvnorm(n,mean=mu[i,],sigma=sigma)
if (distr=="t2") X[i,,]<-rmvt(n, sigma = sigma, df = 2)+mu[i,]
}
X<-matrix(X,ncol=p,byrow=FALSE)
label<-rep(seq(1,C),n)
X<- data.frame(label,X)
m<-array(0,dim=c(C,p))
for(j in 1:p){
for(i in 1:C){
m[i,j]<-mean(X[label==i,(j+1)])
}
}
P<-array(1,dim=c(C,C,p))
D<-array(0,dim=c(C,C,p))
MSD<-array(0,dim=c(p,1))
P_mean<-array(1,dim=c(C,C,p))
P_gps<-array(1,dim=c(C+1,C,p))
D_mean<-array(0,dim=c(C,C,p))

for(j in 1:p){
MSD[j]<-unlist(summary(aov(X[,(j+1)]~as.factor(X$label),data=X)))[6]
}
MSD<-sqrt(MSD)
mm<-t(m)
colnames(mm)<-seq(1:C)
```

```
for(j in 1:p){
P[,,j]=pairwise(m,MSD,k=j,n,alt="greater",correct=FALSE)$P
D[,,j]=pairwise(m,MSD,k=j,n,alt="greater",correct=FALSE)$D
P_mean[,,j]=pairwise(m,MSD,k=j,n,alt="two.sided",correct=TRUE)$P
a=ord.pairwise(mm[j,],MSD[j],n,alt="two.sided",correct=TRUE)$P
b<-as.numeric(colnames(a))
P_gps[,,j]<-rbind(a,b)
}
pp1<-P_gps
for (i in 1:C){
for (j in 1:C){
for (k in 1:p){
if (pp1[i,j,k]>=alpha) pp1[i,j,k]=0
else pp1[i,j,k]=1
}}}
pp<-pp1
for (k in 1:p){
for (j in 1:C){
for (i in j:C){
pp[i,j,k]<-0
}}}
d<-matrix(0,p*2,C) # Fase di pre-aggregazione per GPS
for (k in 1:p){
d[2*k-1,]<-score.inv(pp[-(C+1),,k])
d[2*k,]<-pp[C+1,,k]
}
g<-matrix(0,p,C)
for(k in 1:p){
ii<-order(d[2*k,],d[2*k-1,])
g[k,]<-(d[2*k-1,])[ii]
}

# Fase dell'aggregazione per GPS
e1<-apply(g,2,mean) # Media aritmetica
e2<-apply(g,2,function(x){prod(x)^(1/p)}) # Media geometrica
e3<-apply(g,2,function(x){-2*sum(log(x))}) # Fisher

# Fase della normalizzazione per GPS
e1a<-e1/1 # std della media aritmetica col minimo teorico
e1a<-(e1a-1)/(C-1)
e1b<-e1/min(e1) # std della media aritmetica col minimo campionario
if ((max(e1b)-min(e1b))!=0) e1b<-(e1b-min(e1b))/(max(e1b)-min(e1b))
if ((max(e1b)-min(e1b))==0) e1b<-rep(0,C)
e1c<-e1/C # std della media aritmetica col massimo teorico
e1c<-(e1c-1/C)/(1-1/C)
e1d<-e1/max(e1) # std della media aritmetica col massimo campionario
if ((max(e1d)-min(e1d))!=0) e1d<-(e1d-min(e1d))/(max(e1d)-min(e1d))
if ((max(e1d)-min(e1d))==0) e1d<-rep(0,C)
e1_<-e1 # media aritmetica non standardizzata
e1_<-e1a
if (sd(e1)!=0) {e1e<-(e1-mean(e1))/sd(e1)
e1e<-(e1e-min(e1e))/(max(e1e)-min(e1e))}
if (sd(e1)==0) e1e<-e1_ # std della media aritmetica con media 0 e var 1
e1f<-1/e1 # std della media aritmetica con il suo reciproco (INVERSIONE!)
e1f<-(e1f-1/C)/(1-1/C)
e1g<-(e1-1)/(C-1)
if ((max(e1)-min(e1))!=0) e1h<-(e1-min(e1))/(max(e1)-min(e1))
if ((max(e1)-min(e1))==0) e1h<-rep(0,C)
e1i<-(C-e1)/(C-1) # (INVERSIONE!)
e1i<-e1i
if ((max(e1)-min(e1))!=0) {e1j<-(max(e1)-e1)/(max(e1)-min(e1));e1j<-e1j}
if ((max(e1)-min(e1))==0) e1j<-rep(0,C)
e2a<-e2/1
e2a<-(e2a-1)/(C-1)
e2b<-e2/min(e2)
if ((max(e2b)-min(e2b))!=0) e2b<-(e2b-min(e2b))/(max(e2b)-min(e2b))
if ((max(e2b)-min(e2b))==0) e2b<-rep(0,C)
e2c<-e2/C
e2c<-(e2c-1/C)/(1-1/C)
e2d<-e2/max(e2)
if ((max(e2d)-min(e2d))!=0) e2d<-(e2d-min(e2d))/(max(e2d)-min(e2d))
if ((max(e2d)-min(e2d))==0) e2d<-rep(0,C)
e2_<-e2
e2_<-e2a
if (sd(e2)!=0) {e2e<-(e2-mean(e2))/sd(e2)
e2e<-(e2e-min(e2e))/(max(e2e)-min(e2e))}
if (sd(e2)==0) e2e<-e2_ # std della media aritmetica con media 0 e var 1
e2f<-1/e2 # (INVERSIONE!)
e2f<-(e2f-1/C)/(1-1/C)
e2g<-(e2-1)/(C-1)
if ((max(e2)-min(e2))!=0) e2h<-(e2-min(e2))/(max(e2)-min(e2))
if ((max(e2)-min(e2))==0) e2h<-rep(0,C)
e2i<-(C-e2)/(C-1) # (INVERSIONE!)
e2i<-e2i
if ((max(e2)-min(e2))!=0) {e2j<-(max(e2)-e2)/(max(e2)-min(e2));e2j<-e2j}
if ((max(e2)-min(e2))==0) e2j<-rep(0,C)
e3a<-e3/(-2*p*log(C))
e3b<-e3/min(e3-0.01)
e3c<-e3/-0.01 # Accorgimento per l'applicabilità
e3c<-e3c/(2*p*log(C)/0.01)
e3d<-e3/(max(e3)-0.01) # Accorgimento per l'applicabilità
e3d<-e3d/(-2*p*log(C)/(max(e3)-0.01))
```

```
e3_<-e3 # (INVERSIONE!)
e3_<-e3a
if (sd(e3)!=0) {e3e<-(e3-mean(e3))/sd(e3)
e3e<-(e3e-min(e3e))/(max(e3e)-min(e3e))}
if (sd(e3)==0) e3e<-e3_ # std della media aritmetica con media 0 e var 1
e3f<-1/(e3-0.01) # Accorgimento per l'applicabilità
e3f<-(e3f+1/0.01)/((2*p*log(C))/(2*p*log(C)+0.01)+1/0.01)
e3g<-(e3-(-2*p*log(C)))/(0-(-2*p*log(C))) # (INVERSIONE!)
e3g<-e3g
if ((max(e3)-min(e3))!=0) {e3h<-(e3-min(e3))/(max(e3)-min(e3));e3h<-e3h}
if ((max(e3)-min(e3))==0) e3h<-rep(0,C)
e3i<-(0-e3)/(0-(-2*p*log(C)))
if ((max(e3)-min(e3))!=0) e3j<-(max(e3)-e3)/(max(e3)-min(e3))
if ((max(e3)-min(e3))==0) e3j<-rep(0,C)

# Classifica punteggio GPS
M3<-matrix(0,33,3+C*3)
M3<-as.data.frame(M3)
if (C==5) colnames(M3)<-c("#(rho=1)","rhom","NA","q0.025 1","q0.975 1",
"diff 1","q0.025 2","q0.975 2","diff 2","q0.025 3","q0.975 3","diff 3",
"q0.025 4","q0.975 4","diff 4","q0.025 5","q0.975 5","diff 5")
if (C==10) colnames(M3)<-c("#(rho=1)","rhom","NA","q0.025 1","q0.975 1",
"diff 1","q0.025 2","q0.975 2","diff 2","q0.025 3","q0.975 3","diff 3",
"q0.025 4","q0.975 4","diff 4","q0.025 5","q0.975 5","diff 5","q0.025 6",
"q0.975 6","diff 6","q0.025 7","q0.975 7","diff 7","q0.025 8","q0.975 8",
"diff 8","q0.025 9","q0.975 9","diff 9","q0.025 10","q0.975 10","diff 10")
M33<-matrix(0,33,C)
cn1<-c("rank 1","rank 2","rank 3","rank 4","rank 5")
cn2<-c("rank 1","rank 2","rank 3","rank 4","rank 5","rank 6","rank 7",
"rank 8","rank 9","rank 10")
if (C==5) colnames(M33)<-cn1
if (C==10) colnames(M33)<-cn2
M333<-matrix(0,33,C)
cn11<-c("mean 1","mean 2","mean 3","mean 4","mean 5")
cn22<-c("mean 1","mean 2","mean 3","mean 4","mean 5","mean 6","mean 7",
"mean 8","mean 9","mean 10")
if (C==5) colnames(M333)<-cn11
if (C==10) colnames(M333)<-cn22
eee<-matrix(0,33,C)
eee[1,]<-e1a
eee[2,]<-e2a
eee[3,]<-e3a
eee[4,]<-e1b
eee[5,]<-e2b
eee[6,]<-e3b
eee[7,]<-e1c
eee[8,]<-e2c
eee[9,]<-e3c
eee[10,]<-e1d
eee[11,]<-e2d
eee[12,]<-e3d
eee[13,]<-e1_
eee[14,]<-e2_
eee[15,]<-e3_ # INVERSIONE!
eee[16,]<-e1e
eee[17,]<-e2e
eee[18,]<-1-e3e # INVERSIONE!
eee[19,]<-1-e1f # INVERSIONE!
eee[20,]<-1-e2f # INVERSIONE!
eee[21,]<-e3f
eee[22,]<-e1g
eee[23,]<-e2g
```

```
eee[24,]<-1-e3g # INVERSIONE!
eee[25,]<-e1h
eee[26,]<-e2h
eee[27,]<-1-e3h # INVERSIONE!
eee[28,]<-1-e1i # INVERSIONE!
eee[29,]<-1-e2i # INVERSIONE!
eee[30,]<-e3i
eee[31,]<-1-e1j # INVERSIONE!
eee[32,]<-1-e2j # INVERSIONE!
eee[33,]<-e3j
for (i in c(1:33)){
Score_gps[,cc,i]<-rank(eee[i,],ties.method = "max")
SP_gps[cc,,i]<-cor.test(Score_gps[,cc,i],seq(C,1),method = "spearman")$estimate
if (length(table(SP_gps[,,i][SP_gps[,,i]>0.9999]))>0)
{M3[i,1]<-table(SP_gps[,,i][SP_gps[,,i]>0.9999])/B}
else {M3[i,1]<-0}
M3[i,2]<-mean(SP_gps[,,i],na.rm=T)
M3[i,3]<-B-sum(table(SP_gps[,,i]))
Score_gps[,cc,i]<-eee[i,]
for (j in 1:C){
M3[i,j*3+1]<-quantile(Score_gps[j,,i],0.025)
M3[i,j*3+2]<-quantile(Score_gps[j,,i],0.975)
M3[i,j*3+3]<-M3[i,j*3+2]-M3[i,j*3+1]
}
}
for (i in c(1:33)){
for (j in 1:C){
M333[i,j]<-mean(Score_gps[j,,i])
}
M33[i,]<-rank(M333[i,],ties.method="max")
}
```

```
# Fase di pre-aggregazione e pre-normalizzazione
D_mean<-apply(ifelse(P_mean<alpha,sign(D)*1,0),c(1,3),sum)
Sc_mean[cc,,]<-D_mean
Sc_mean[cc,,]<-D_mean+(C-1)
Sc_mean[cc,,]<-Sc_mean[cc,,]/(2*(C-1))
for (i in 1:C) {
for (j in 1:p) {
if (Sc_mean[cc,i,j]==0) Sc_mean[cc,i,j]<-Sc_mean[cc,i,j]+0.0000000001
}
}

# Fase dell'aggregazione per Media
e1<-apply(Sc_mean[cc,,],1,mean) # Media aritmetica
e2<-apply(Sc_mean[cc,,],1,function(x){prod(x)^(1/p)}) # Media geometrica
e3<-apply(Sc_mean[cc,,],1,function(x){-2*sum(log(x))}) # Fisher

# Fase della normalizzazione per Media
e1a<-e1/0.0000000001 # std della media aritmetica col minimo teorico
e1a<-(e1a-1)/(1/0.0000000001-1)
e1b<-e1/min(e1) # std della media aritmetica col minimo campionario
if ((max(e1b)-min(e1b))!=0) e1b<-(e1b-min(e1b))/(max(e1b)-min(e1b))
if ((max(e1b)-min(e1b))==0) e1b<-rep(0,C)
e1c<-e1/1 # std della media aritmetica col massimo teorico
e1c<-(e1c-0.0000000001)/0.9999999999
e1d<-e1/max(e1) # std della media aritmetica col massimo campionario
if ((max(e1d)-min(e1d))!=0) e1d<-(e1d-min(e1d))/(max(e1d)-min(e1d))
if ((max(e1d)-min(e1d))==0) e1d<-rep(0,C)
e1_<-e1 # media aritmetica non standardizzata
e1_<-e1c
if (sd(e1)!=0) {e1e<-(e1-mean(e1))/sd(e1)
e1e<-(e1e-min(e1e))/(max(e1e)-min(e1e))}
if (sd(e1)==0) e1e<-e1 # std della media aritmetica con media 0 e var 1
e1f<-1/e1 # std della media aritmetica con il suo reciproco (INVERSIONE!)
e1f<-(e1f-1)/(1/0.0000000001-1)
e1g<-(e1-0.0000000001)/0.9999999999
if ((max(e1)-min(e1))!=0) e1h<-(e1-min(e1))/(max(e1)-min(e1))
if ((max(e1)-min(e1))==0) e1h<-rep(0,C)
e1i<-(1-e1)/0.9999999999 # (INVERSIONE!)
if ((max(e1)-min(e1))!=0) e1j<-(max(e1)-e1)/(max(e1)-min(e1)) # (INVERSIONE!)
if ((max(e1)-min(e1))==0) e1j<-rep(0,C)
e2a<-e2/0.0000000001 # std della media aritmetica col minimo teorico
e2a<-(e2a-1)/(1/0.0000000001-1)
e2b<-e2/min(e2) # std della media aritmetica col minimo campionario
if ((max(e2b)-min(e2b))!=0) e2b<-(e2b-min(e2b))/(max(e2b)-min(e2b))
if ((max(e2b)-min(e2b))==0) e2b<-rep(0,C)
e2c<-e2/1 # std della media aritmetica col massimo teorico
e2c<-(e2c-0.0000000001)/0.9999999999
e2d<-e2/max(e2) # std della media aritmetica col massimo campionario
if ((max(e2d)-min(e2d))!=0) e2d<-(e2d-min(e2d))/(max(e2d)-min(e2d))
if ((max(e2d)-min(e2d))==0) e2d<-rep(0,C)
e2_<-e2 # media aritmetica non standardizzata
e2_<-e2c
if (sd(e2)!=0) {e2e<-(e2-mean(e2))/sd(e2)
e2e<-(e2e-min(e2e))/(max(e2e)-min(e2e))}
if (sd(e2)==0) e2e<-e2 # std della media aritmetica con media 0 e var 1
e2f<-1/e2 # std della media aritmetica con il suo reciproco (INVERSIONE!)
e2f<-(e2f-1)/(1/0.0000000001-1)
e2g<-(e2-0.0000000001)/0.9999999999
if ((max(e2)-min(e2))!=0) e2h<-(e2-min(e2))/(max(e2)-min(e2))
if ((max(e2)-min(e2))==0) e2h<-rep(0,C)
e2i<-(1-e2)/0.9999999999 # (INVERSIONE!)
```

```
if ((max(e2)-min(e2))!=0) e2j<-(max(e2)-e2)/(max(e2)-min(e2)) # (INVERSIONE!)
if ((max(e2)-min(e2))==0) e2j<-rep(0,C)
max=-2*p*log(0.0000000001)
min=0.0000000001
e3a<-e3/min # (INVERSIONE!)
e3a<-e3a/(max/min)
e3b<-e3/min(e3+min) # (INVERSIONE!)
if ((max(e3b)-min(e3b))!=0) e3b<-(e3b-min(e3b))/(max(e3b)-min(e3b))
if ((max(e3b)-min(e3b))==0) e3b<-rep(0,C)
e3c<-e3/max # (INVERSIONE!)
e3d<-e3/max(e3) # (INVERSIONE!)
if ((max(e3d)-min(e3d))!=0) e3d<-(e3d-min(e3d))/(max(e3d)-min(e3d))
if ((max(e3d)-min(e3d))==0) e3d<-rep(0,C)
e3_<-e3 # (INVERSIONE!)
e3_<-e3_/max
if (sd(e3)!=0) {e3e<-(e3-mean(e3))/sd(e3)
e3e<-(e3e-min(e3e))/(max(e3e)-min(e3e))}
if (sd(e3)==0) e3e<-e3_ # std della media aritmetica con media 0 e var 1
e3f<-1/(e3+0.0000000001)
e3f<-(e3f-1/max)/(1/min-1/max)
e3g<-e3c # (INVERSIONE!)
if ((max(e3)-min(e3))!=0) e3h<-(e3-min(e3))/(max(e3)-min(e3)) # (INVERSIONE!)
if ((max(e3)-min(e3))==0) e3h<-rep(0,C)
e3i<-(max-e3)/max
if ((max(e3)-min(e3))!=0) e3j<-(max(e3)-e3)/(max(e3)-min(e3))
if ((max(e3)-min(e3))==0) e3j<-rep(0,C)

# Classifica punteggio Media
M2<-matrix(0,33,3+C*3)
M2<-as.data.frame(M2)
if (C==5) colnames(M2)<-c("#(rho=1)","rhom","NA","q0.025 1","q0.975 1",
"diff 1","q0.025 2","q0.975 2","diff 2","q0.025 3","q0.975 3","diff 3",
"q0.025 4","q0.975 4","diff 4","q0.025 5","q0.975 5","diff 5")
if (C==10) colnames(M2)<-c("#(rho=1)","rhom","NA","q0.025 1","q0.975 1",
"diff 1","q0.025 2","q0.975 2","diff 2","q0.025 3","q0.975 3","diff 3",
"q0.025 4","q0.975 4","diff 4","q0.025 5","q0.975 5","diff 5","q0.025 6",
"q0.975 6","diff 6","q0.025 7","q0.975 7","diff 7","q0.025 8","q0.975 8",
"diff 8","q0.025 9","q0.975 9","diff 9","q0.025 10","q0.975 10","diff 10")
M22<-matrix(0,33,C)
if (C==5) colnames(M22)<-cn1
if (C==10) colnames(M22)<-cn2
M222<-matrix(0,33,C)
if (C==5) colnames(M222)<-cn11
if (C==10) colnames(M222)<-cn22
eee<-matrix(0,33,C)
eee[1,]<-e1a
eee[2,]<-e2a
eee[3,]<-1-e3a # INVERSIONE!
eee[4,]<-e1b
eee[5,]<-e2b
eee[6,]<-1-e3b # INVERSIONE!
eee[7,]<-e1c
eee[8,]<-e2c
eee[9,]<-1-e3c # INVERSIONE!
eee[10,]<-e1d
eee[11,]<-e2d
eee[12,]<-1-e3d # INVERSIONE!
eee[13,]<-e1_
eee[14,]<-e2_
eee[15,]<-1-e3_ # INVERSIONE!
eee[16,]<-e1e
eee[17,]<-e2e
```

```
eee[18,]<-1-e3e # INVERSIONE!
eee[19,]<-1-e1f # INVERSIONE!
eee[20,]<-1-e2f # INVERSIONE!
eee[21,]<-e3f
eee[22,]<-e1g
eee[23,]<-e2g
eee[24,]<-1-e3g # INVERSIONE!
eee[25,]<-e1h
eee[26,]<-e2h
eee[27,]<-1-e3h # INVERSIONE!
eee[28,]<-1-e1i # INVERSIONE!
eee[29,]<-1-e2i # INVERSIONE!
eee[30,]<-e3i
eee[31,]<-1-e1j # INVERSIONE!
eee[32,]<-1-e2j # INVERSIONE!
eee[33,]<-e3j
for (i in c(1:33)){
Score_mean[,cc,i]<-rank(eee[i,],ties.method = "max")
SP_mean[cc,,i]<-cor.test(Score_mean[,cc,i],seq(C,1),method = "spearman")
$estimate
if (length(table(SP_mean[,,i][SP_mean[,,i]>0.9999]))>0)
{M2[i,1]<-table(SP_mean[,,i][SP_mean[,,i]>0.9999])/B}
else {M2[i,1]<-0}
M2[i,2]<-mean(SP_mean[,,i],na.rm=T)
M2[i,3]<-B-sum(table(SP_mean[,,i]))
Score_mean[,cc,i]<-eee[i,]
for (j in 1:C){
M2[i,j*3+1]<-quantile(Score_mean[j,,i],0.025)
M2[i,j*3+2]<-quantile(Score_mean[j,,i],0.975)
M2[i,j*3+3]<-M2[i,j*3+2]-M2[i,j*3+1]
}
```

```
}
for (i in c(1:33)){
for (j in 1:C){
M222[i,j]<-mean(Score_mean[j,,i])
}
M22[i,]<-rank(M222[i,],ties.method="max")
}

# Fase di pre-aggregazione per NPC
Sc[cc,,]<-apply(P,c(1,3),function(x){-2*sum(log(x+0.00000000000000000001))})

# Fase dell'aggregazione per NPC
e1<-apply(Sc[cc,,],1,mean) # Media aritmetica
e2<-apply(Sc[cc,,],1,function(x){prod(x)^(1/p)}) # Media geometrica
e3<-apply(Sc[cc,,],1,function(x){-2*sum(log(x))}) # Fisher

# Fase della normalizzazione per NPC
mm<-(-2*C*log(1.0000000001))
max<-(-2*C*log(0.0000000001)) # Trucchetti per tentare comunque le std.
min<-(-2*C*log(max)) # Trucchetti per tentare comunque le std.
e1a<-e1/0.0000000001 # std della media aritmetica col minimo teorico
e1a<-(e1-mm/0.0000000001)/(max/0.0000000001-mm/0.0000000001)
e1b<-e1/min(e1) # std della media aritmetica col minimo campionario
e1b<-(e1b-min(e1b))/(max(e1b)-min(e1b))
e1c<-e1/max # std della media aritmetica col massimo teorico
e1c<-(e1c-mm/max)/(1-mm/max)
e1d<-e1/max(e1) # std della media aritmetica col massimo campionario
e1d<-(e1d-min(e1d))/(max(e1d)-min(e1d))
e1_<-e1 # media aritmetica non standardizzata
e1_<-(e1-mm)/(max-mm)
if (sd(e1)!=0) {e1e<-(e1-mean(e1))/sd(e1)
```

```
e1e<-(e1e-min(e1e))/(max(e1e)-min(e1e))}
if (sd(e1)==0) e1e<-e1 # std della media aritmetica con media 0 e var 1
e1f<-1/(e1+0.0000000001) # std della media aritmetica con il suo reciproco
e1f<-(e1f-1/max)/(1/max-1/0.0000000001)
e1g<-e1_ # UGUALE A e1c
if ((max(e1)-min(e1))!=0) e1h<-(e1-min(e1))/(max(e1)-min(e1))
if ((max(e1)-min(e1))==0) e1h<-rep(0,C)
e1i<-(max-e1)/(max-0) # (INVERSIONE!) UGUALE A 1-e1c
if ((max(e1)-min(e1))!=0) e1j<-(max(e1)-e1)/(max(e1)-min(e1)) # (INVERSIONE!)
if ((max(e1)-min(e1))==0) e1j<-rep(0,C)
e2a<-e2/0.0000000001
e2a<-(e2-mm/0.0000000001)/(max/0.0000000001-mm/0.0000000001)
e2b<-e2/min(e2)
e2b<-(e2b-min(e2b))/(max(e2b)-min(e2b))
e2c<-e2/max
e2c<-(e2c-mm/max)/(1-mm/max)
e2d<-e2/max(e2)
e2d<-(e2d-min(e2d))/(max(e2d)-min(e2d))
e2_<-e2
e2_<-(e2-mm)/(max-mm)
if (sd(e2)!=0) {e2e<-(e2-mean(e2))/sd(e2)
e2e<-(e2e-min(e2e))/(max(e2e)-min(e2e))}
if (sd(e2)==0) e2e<-e2 # std della media aritmetica con media 0 e var 1
e2f<-1/(e2+0.0000000001) # (INVERSIONE!)
e2f<-(e2f-1/max)/(1/max-1/0.0000000001)
e2g<-e2_ # UGUALE A e2c
if ((max(e2)-min(e2))!=0) e2h<-(e2-min(e2))/(max(e2)-min(e2))
if ((max(e2)-min(e2))==0) e2h<-rep(0,C)
e2i<-(max-e2)/(max-0) # (INVERSIONE!) UGUALE A 1-e2c
if ((max(e2)-min(e2))!=0) e2j<-(max(e2)-e2)/(max(e2)-min(e2)) # (INVERSIONE!)
if ((max(e2)-min(e2))==0) e2j<-rep(0,C)
```

```
e3a<-e3/min
e3a<-(e3a-max/min)/(1-max/min)
e3b<-e3/min(e3)
e3b<-(e3b-min(e3b))/(max(e3b)-min(e3b))
e3c<-e3/max # (INVERSIONE!)
e3c<-(e3c-1)/(1-min/max)
e3d<-e3/max(e3) # (INVERSIONE!)
e3d<-(e3d-min(e3d))/(max(e3d)-min(e3d))
e3_<-e3 # (INVERSIONE!)
e3_<-(e3-min)/(max-min)
if (sd(e3)!=0) {e3e<-(e3-mean(e3))/sd(e3)
e3e<-(e3e-min(e3e))/(max(e3e)-min(e3e))}
if (sd(e3)==0) e3e<-e3 # std della media aritmetica con media 0 e var 1
#e3f<-1/e3 # Senza senso perché abbiamo valori sia positivi che negativi
e3g<-e3_ # (INVERSIONE!)
if ((max(e3)-min(e3))!=0) e3h<-(e3-min(e3))/(max(e3)-min(e3))
if ((max(e3)-min(e3))==0) e3h<-rep(0,C)
#e3i<-((-2*p*log(1.5/(C+1)))-e3)/((-2*p*log(1.5/(C+1)))-
(-2*p*log((C+0.5)/(C+1))))
e3i<-(max-e3)/(max-min)
if ((max(e3)-min(e3))!=0) e3j<-(max(e3)-e3)/(max(e3)-min(e3)) # (INVERSIONE!)
if ((max(e3)-min(e3))==0) e3j<-rep(0,C)

# Classifica punteggio NPC
M1<-matrix(0,33,3+C*3)
M1<-as.data.frame(M1)
if (C==5) colnames(M1)<-c("#(rho=1)","rhom","NA","q0.025 1","q0.975 1",
"diff 1","q0.025 2","q0.975 2","diff 2","q0.025 3","q0.975 3","diff 3",
"q0.025 4","q0.975 4","diff 4","q0.025 5","q0.975 5","diff 5")
if (C==10) colnames(M1)<-c("#(rho=1)","rhom","NA","q0.025 1","q0.975 1",
"diff 1","q0.025 2","q0.975 2","diff 2","q0.025 3","q0.975 3","diff 3",
"q0.025 4","q0.975 4","diff 4","q0.025 5","q0.975 5","diff 5","q0.025 6",
"q0.975 6","diff 6","q0.025 7","q0.975 7","diff 7","q0.025 8","q0.975 8",
"diff 8","q0.025 9","q0.975 9","diff 9","q0.025 10","q0.975 10","diff 10")
M11<-matrix(0,33,C)
if (C==5) colnames(M11)<-cn1
if (C==10) colnames(M11)<-cn2
M111<-matrix(0,33,C)
if (C==5) colnames(M111)<-cn11
if (C==10) colnames(M111)<-cn22
eee<-matrix(0,33,C)
eee[1,]<-e1a
eee[2,]<-e2a
eee[3,]<-e3a
eee[4,]<-e1b
eee[5,]<-e2b
if (min(e3)<0) {eee[6,]<-e3b}
if (min(e3)>=0) {eee[6,]<-1-e3b} # Inversione condizionata
eee[7,]<-e1c
eee[8,]<-e2c
eee[9,]<--e3c # INVERSIONE!
eee[10,]<-e1d
eee[11,]<-e2d
if (max(e3)<=0) {eee[12,]<-e3d}
if (max(e3)>0) {eee[12,]<-1-e3d} # Inversione condizionata
eee[13,]<-e1_
eee[14,]<-e2_
eee[15,]<-1-e3_ # INVERSIONE!
eee[16,]<-e1e
eee[17,]<-e2e
eee[18,]<-1-e3e # INVERSIONE!
eee[19,]<-1+e1f # INVERSIONE!
```

```
eee[20,]<-1+e2f # INVERSIONE!
#eee[21,]<-e3f
eee[22,]<-e1g
eee[23,]<-e2g
eee[24,]<-1-e3g # INVERSIONE!
eee[25,]<-e1h
eee[26,]<-e2h
eee[27,]<-1-e3h # INVERSIONE!
eee[28,]<-1-e1i # INVERSIONE!
eee[29,]<-1-e2i # INVERSIONE!
eee[30,]<-e3i
eee[31,]<-1-e1j # INVERSIONE!
eee[32,]<-1-e2j # INVERSIONE!
eee[33,]<-e3j

for (i in c(1:20,22:33)){
Score[,cc,i]<-rank(eee[i,],ties.method = "max")
SP[cc,,i]<-cor.test(Score[,cc,i],seq(C,1),method = "spearman")$estimate
if (length(table(SP[,,i][SP[,,i]>0.9999]))>0)
{M1[i,1]<-table(SP[,,i][SP[,,i]>0.9999])/B}
else {M1[i,1]<-0}
M1[i,2]<-mean(SP[,,i],na.rm=T)
M1[i,3]<-B-sum(table(SP[,,i]))
Score[,cc,i]<-eee[i,]
for (j in 1:C){
M1[i,j*3+1]<-quantile(Score[j,,i],0.025)
M1[i,j*3+2]<-quantile(Score[j,,i],0.975)
M1[i,j*3+3]<-M1[i,j*3+2]-M1[i,j*3+1]
}
}
for (i in c(1:20,22:33)){
```

```
for (j in 1:C){
M111[i,j]<-mean(Score[j,,i])
}
M11[i,]<-rank(M111[i,],ties.method="max")
}
} #fine generazione dei dati

t<-list(M1,M111,M11,M2,M222,M22,M3,M333,M33)
names(t)<-c("NPC","NPCm","NPCgm","Media","Mediam","Mediagm",
"GPS","GPSm","GPSgm")
t
}

# Simulazioni:
set.seed(123)
a<-simula3.settings(distr="norm",mu=mu5b,sigma=S1b,n=4)
write.table(a$NPC,"a1.txt",sep="\t",row.names=F,col.names=F)
write.table(a$NPCm,"a2.txt",sep="\t",row.names=F,col.names=F)
write.table(a$NPCgm,"a3.txt",sep="\t",row.names=F,col.names=F)
write.table(a$Media,"a4.txt",sep="\t",row.names=F,col.names=F)
write.table(a$Mediam,"a5.txt",sep="\t",row.names=F,col.names=F)
write.table(a$Mediagm,"a6.txt",sep="\t",row.names=F,col.names=F)
write.table(a$GPS,"a7.txt",sep="\t",row.names=F,col.names=F)
write.table(a$GPSm,"a8.txt",sep="\t",row.names=F,col.names=F)
write.table(a$GPSgm,"a9.txt",sep="\t",row.names=F,col.names=F)
set.seed(123)
b<-simula3.settings(distr="norm",mu=mu5b,sigma=S1b,n=8)
write.table(b$NPC,"b1.txt",sep="\t",row.names=F,col.names=F)
write.table(b$NPCm,"b2.txt",sep="\t",row.names=F,col.names=F)
write.table(b$NPCgm,"b3.txt",sep="\t",row.names=F,col.names=F)
write.table(b$Media,"b4.txt",sep="\t",row.names=F,col.names=F)
write.table(b$Mediam,"b5.txt",sep="\t",row.names=F,col.names=F)
write.table(b$Mediagm,"b6.txt",sep="\t",row.names=F,col.names=F)
write.table(b$GPS,"b7.txt",sep="\t",row.names=F,col.names=F)
write.table(b$GPSm,"b8.txt",sep="\t",row.names=F,col.names=F)
write.table(b$GPSgm,"b9.txt",sep="\t",row.names=F,col.names=F)
set.seed(123)
c<-simula3.settings(distr="norm",mu=mu10b,sigma=S1b,n=4)
write.table(c$NPC,"c1.txt",sep="\t",row.names=F,col.names=F)
write.table(c$NPCm,"c2.txt",sep="\t",row.names=F,col.names=F)
write.table(c$NPCgm,"c3.txt",sep="\t",row.names=F,col.names=F)
write.table(c$Media,"c4.txt",sep="\t",row.names=F,col.names=F)
write.table(c$Mediam,"c5.txt",sep="\t",row.names=F,col.names=F)
write.table(c$Mediagm,"c6.txt",sep="\t",row.names=F,col.names=F)
write.table(c$GPS,"c7.txt",sep="\t",row.names=F,col.names=F)
write.table(c$GPSm,"c8.txt",sep="\t",row.names=F,col.names=F)
write.table(c$GPSgm,"c9.txt",sep="\t",row.names=F,col.names=F)
...
(48 different simulations, therefore other 45)
```

A.4 Functions we need for all three simulation studies

A.4.1 Pairwise

This function is useful to do multiple comparisons and it returns a matrix of p-values and a matrix of differences (it is useful to perform NPC and AISE):

```
pairwise<-function(m,MSD,k,n=4,alt=c("less","two.sided","greater"),
correct=c(TRUE,FALSE)){

C<-dim(m)[1]
T<-array(0,dim=c(C,C))
P<-array(1,dim=c(C,C))
D<-array(0,dim=c(C,C))
df<-(n*C)-C

for(i in 1:(C-1)){
for(j in (i+1):C){
D[i,j]<- m[i,k]-m[j,k]
t<-sqrt(2)*(D[i,j])/MSD[k]
if(alt=="less"){P[i,j]<-pt(t,df);P[j,i]<-1-pt(t,df)}
if(alt=="two.sided"){P[i,j]<-2*min(pt(t,df),(1-pt(t,df)));P[j,i]=P[i,j]}
if(alt=="greater"){P[i,j]<-1-pt(t,df);P[j,i]<-pt(t,df)}
D[j,i]<- -D[i,j]
}
}

n.test<-choose(C,2)
if(correct==TRUE){P<-apply(P,c(1,2),function(x){min(x*n.test,1)})}
return(list(P=P,D=D))
}
```

A.4.2 Ord.pairwise

This function is useful to do multiple comparisons and it returns a matrix of *p*-values and a matrix of differences, but it is different from the previous because it pre-orders sample means (it is useful to perform GPS):

```
ord.pairwise<-function(m,MSD,n=4,alt="two.sided",correct=c(TRUE,FALSE)){

ordm<-sort(m,decreasing=TRUE)
ordmp<-as.matrix(ordm,1,C)
o<-t(ordmp)
C<-length(m)
T<-array(0,dim=c(C,C))
P<-array(1,dim=c(C,C))
D<-array(0,dim=c(C,C))
df<-(n*C)-C

for(i in 1:(C-1)){
for(j in (i+1):C){
D[i,j]<- ordm[i]-ordm[j]
t<-sqrt(2)*(D[i,j])/MSD
if (alt=="two.sided"){P[i,j]<-2*min(pt(t,df),(1-pt(t,df)));P[j,i]=P[i,j]}
D[j,i]<- -D[i,j]
}
}

colnames(P)<-colnames(o)
n.test<-choose(C,2)
if(correct==TRUE){P<-apply(P,c(1,2),function(x){min(x*n.test,1)})}
return(list(P=P,D=D))
}
```

A.4.3 Score

This last function contains in turn other three functions that are needed to calculate GPS score; `score` and `score2` were two different ways to calculate GPS score as it was thought in the origin (ranks were inverted and you had to begin with 1 and not with C), `score.inv` is the correct one and the function used in the three codes:

```
score<-function(X){
rango<-function(x){
l=seq(1,length(x))
o=order(x,decreasing=F)
r=vector(length=length(x))
r[o]=rep(seq(1,length(table(x))),table(x))
return(r)
}
row=rango(apply(-X,1,sum))
col=rango(apply(X,2,sum))
scores=apply(cbind(row,col),1,mean)
return(scores)
}

score.inv<-function(xx){
x<-1-xx
C=dim(x)[1]
s<-matrix(0,C,C)
for (i in 1:C) {
for (j in 1:C) {
if (x[i,j]==1) s[i,j]=C+1-i
else s[i,j]=0
}
}
cfr_row<-matrix(0,C,C)
for (i in 1:(C-1)) {
for (j in (i+1):C) {
if ((sum(as.numeric(xx[i,]==xx[j,])))!=C) cfr_row[i,j]=1
else cfr_row[i,j]=0
}
}
r<-double(C)
for (i in 1:C) {
if (sum(cfr_row[i,])==C-i) r[i]=1
else r[i]=0
}
k<-c(0,seq(1:C)[r==1])
```

```
for (i in (1:(length(k)-1))) {
r[(k[i]+1):(k[i+1])]<-sort(r[(k[i]+1):(k[i+1])],decreasing=T)
}
for (i in 1:C) {
if (r[i]==1) s[i,]<-s[i,]
else s[i,]<-double(C)
}
for (i in 2:C) {
for (j in 1:(i-1)) {
if (sum(s[i,])==0) s[i,]=0
else {if (s[i,j]!=0) s[i,j]=0}
}
}
for (i in 1:C) {
if (sum(s[i,])==0) s<-rbind(s[-i,],double(C))
}
for (i in 1:C) {
if (sum(s[i,])==0) s<-rbind(s[-i,],double(C))
}
for (i in 1:C) {
if (sum(s[i,])==0) s<-rbind(s[-i,],double(C))
}
for (i in 1:C) {
if (sum(s[i,])==0) s<-rbind(s[-i,],double(C))
}
for (i in 1:C) {
if (sum(s[i,])==0) s<-rbind(s[-i,],double(C))
}
for (i in 1:C) {
if (sum(s[i,])==0) s<-rbind(s[-i,],double(C))
}
```

```
for (i in 1:C) {
if (sum(s[i,])==0) s<-rbind(s[-i,],double(C))
}
for (i in 1:C) {
if (sum(s[i,])==0) s<-rbind(s[-i,],double(C))
}
for (i in 1:C) {
if (sum(s[i,])==0) s<-rbind(s[-i,],double(C))
}
for (i in 1:C) {
if (sum(s[i,])==0) s<-rbind(s[-i,],double(C))
}
for (i in 1:C) {
if (sum(s[i,])==0) s<-rbind(s[-i,],double(C))
}
for (i in 1:C) {
if (sum(s[i,])==0) s<-rbind(s[-i,],double(C))
}
for (i in 1:C) {
for (j in 1:C) {
if (s[i,j]!=0) s[i,j]=C+1-i
}
}
t<-double(C)
for (j in 1:C) {
t[j]<-mean(s[,j][s[,j]!=0])
}
t
}

score2<-function(xx){
```

```
x<-1-xx
C=dim(x)[1]
s<-matrix(0,C,C)
for (i in 1:C) {
for (j in 1:C) {
if (x[i,j]==1) s[i,j]=i
else s[i,j]=0
}
}
cfr_row<-matrix(0,C,C)
for (i in 1:(C-1)) {
for (j in (i+1):C) {
if ((sum(as.numeric(xx[i,]==xx[j,])))!=C) cfr_row[i,j]=1
else cfr_row[i,j]=0
}
}
r<-double(C)
for (i in 1:C) {
if (sum(cfr_row[i,])==C-i) r[i]=1
else r[i]=0
}
k<-c(0,seq(1:C)[r==1])
for (i in (1:(length(k)-1))) {
r[(k[i]+1):(k[i+1])]<-sort(r[(k[i]+1):(k[i+1])],decreasing=T)
}
for (i in 1:C) {
if (r[i]==1) s[i,]<-s[i,]
else s[i,]<-double(C)
}
for (i in 2:C) {
for (j in 1:(i-1)) {
```

```
if (sum(s[i,])==0) s[i,]=0
else {if (s[i,j]!=0) s[i,j]=0}
}
}
for (i in 1:C) {
if (sum(s[i,])==0) s<-rbind(s[-i,],double(C))
}
for (i in 1:C) {
if (sum(s[i,])==0) s<-rbind(s[-i,],double(C))
}
for (i in 1:C) {
if (sum(s[i,])==0) s<-rbind(s[-i,],double(C))
}
for (i in 1:C) {
if (sum(s[i,])==0) s<-rbind(s[-i,],double(C))
}
for (i in 1:C) {
if (sum(s[i,])==0) s<-rbind(s[-i,],double(C))
}
for (i in 1:C) {
if (sum(s[i,])==0) s<-rbind(s[-i,],double(C))
}
for (i in 1:C) {
if (sum(s[i,])==0) s<-rbind(s[-i,],double(C))
}
for (i in 1:C) {
if (sum(s[i,])==0) s<-rbind(s[-i,],double(C))
}
for (i in 1:C) {
if (sum(s[i,])==0) s<-rbind(s[-i,],double(C))
}
```

```
for (i in 1:C) {
if (sum(s[i,])==0) s<-rbind(s[-i,],double(C))
}
for (i in 1:C) {
if (sum(s[i,])==0) s<-rbind(s[-i,],double(C))
}
for (i in 1:C) {
if (sum(s[i,])==0) s<-rbind(s[-i,],double(C))
}
for (i in 1:C) {
for (j in 1:C) {
if (s[i,j]!=0) s[i,j]=i
}
}
t<-double(C)
for (j in 1:C) {
t[j]<-mean(s[,j][s[,j]!=0])
}
t
}
```

A.5 R code for multivariate RCB design

```
library(mnormt)
library(mvtnorm)
library(sn)
library(vegan)

mu21<-matrix(rep(0,6),2,3)
mu22<-mu21;mu22[2,1]<-4
mu23<-mu22;mu23[2,2]<-4
```

```
mu24<-mu23;mu24[2,3]<-4
colnames(mu21)<-c("X","Y","Z");rownames(mu21)<-c(1,2)
colnames(mu22)<-c("X","Y","Z");rownames(mu22)<-c(1,2)
colnames(mu23)<-c("X","Y","Z");rownames(mu23)<-c(1,2)
colnames(mu24)<-c("X","Y","Z");rownames(mu24)<-c(1,2)
mu31<-matrix(rep(0,9),3,3)
mu32<-matrix(0,3,3);mu32[2,]<-c(2,0,0);mu32[3,]<-mu22[2,]
mu33<-mu32;mu33[,2]<-mu33[,1]
mu34<-mu33;mu34[,3]<-mu33[,1]
colnames(mu31)<-c("X","Y","Z");rownames(mu31)<-c(1,2,3)
colnames(mu32)<-c("X","Y","Z");rownames(mu32)<-c(1,2,3)
colnames(mu33)<-c("X","Y","Z");rownames(mu33)<-c(1,2,3)
colnames(mu34)<-c("X","Y","Z");rownames(mu34)<-c(1,2,3)
mu41<-matrix(rep(0,12),4,3)
mu42<-mu41;mu42[,1]<-c(0,2,3,4)
mu43<-mu42;mu43[,2]<-c(0,2,3,4)
mu44<-mu43;mu44[,3]<-c(0,2,3,4)
colnames(mu41)<-c("X","Y","Z");rownames(mu41)<-c(1,2,3,4)
colnames(mu42)<-c("X","Y","Z");rownames(mu42)<-c(1,2,3,4)
colnames(mu43)<-c("X","Y","Z");rownames(mu43)<-c(1,2,3,4)
colnames(mu44)<-c("X","Y","Z");rownames(mu44)<-c(1,2,3,4)
mu51<-matrix(rep(0,15),5,3)
mu52<-mu51;mu52[,1]<-c(0,1,2,3,4)
mu53<-mu52;mu53[,2]<-c(0,1,2,3,4)
mu54<-mu53;mu54[,3]<-c(0,1,2,3,4)
colnames(mu51)<-c("X","Y","Z");rownames(mu51)<-c(1,2,3,4,5)
colnames(mu52)<-c("X","Y","Z");rownames(mu52)<-c(1,2,3,4,5)
colnames(mu53)<-c("X","Y","Z");rownames(mu53)<-c(1,2,3,4,5)
colnames(mu54)<-c("X","Y","Z");rownames(mu54)<-c(1,2,3,4,5)
S1<-matrix(c(9,0,0,0,9,0,0,0,9),3,3)
S2<-matrix(c(9,-0.5,0.5,-0.5,9,-0.25,0.5,-0.25,9),3,3)
```

```
simula.adonis<-function(mu=mu21,B=1000,np=1000){
C=dim(mu)[1]
p=dim(mu)[2]
dataset<-matrix(0,B*36*C*6,10)
k<-dim(dataset)[1]
colnames(dataset)<-c("ID_sim","Num_blocchi","Num_tratt","ID_blocco",
"ID_tratt","Distr","Sigma","X","Y","Z")
for (i in 1:(36*B*C)){
dataset[i,2]<-6
}
for (i in ((36*B*C)+1):(96*B*C)){
dataset[i,2]<-10
}
for (i in ((96*B*C)+1):k){
dataset[i,2]<-20
}
dataset[,3]<-C
dataset[1:(36*B*C),4]<-rep(1:6,(6*B*C))
dataset[((36*B*C)+1):(96*B*C),4]<-rep(1:10,(6*B*C))
dataset[((96*B*C)+1):k,4]<-rep(1:20,(6*B*C))
if (C==2) {
dataset[1:(36*B*C),5]<-rep(c(rep(1,6),rep(2,6)),(3*B*C))
dataset[((36*B*C)+1):(96*B*C),5]<-rep(c(rep(1,10),rep(2,10)),(3*B*C))
dataset[((96*B*C)+1):k,5]<-rep(c(rep(1,20),rep(2,20)),(3*B*C))
dataset<-as.data.frame(dataset)
dataset[1:(36*B*C),6]<-rep(c(rep("norm",24),rep("t2",24),rep("nasimm",24)),B)
dataset[((36*B*C)+1):(96*B*C),6]<-rep(c(rep("norm",40),rep("t2",40),
rep("nasimm",40)),B)
dataset[((96*B*C)+1):k,6]<-rep(c(rep("norm",80),rep("t2",80),
rep("nasimm",80)),B)
```

```
dataset[1:(36*B*C),7]<-rep(c(rep(c(rep("S1",12),rep("S2",12)),3)),B)
dataset[((36*B*C)+1):(96*B*C),7]<-rep(c(rep(c(rep("S1",20),rep("S2",20)),3)),B)
dataset[((96*B*C)+1):k,7]<-rep(c(rep(c(rep("S1",40),rep("S2",40)),3)),B)
}
if (C==3) {
dataset[1:(36*B*C),5]<-rep(c(rep(1,6),rep(2,6),rep(3,6)),(2*B*C))
dataset[((36*B*C)+1):(96*B*C),5]<-rep(c(rep(1,10),rep(2,10),rep(3,10)),(2*B*C))
dataset[((96*B*C)+1):k,5]<-rep(c(rep(1,20),rep(2,20),rep(3,20)),(2*B*C))
dataset<-as.data.frame(dataset)
dataset[1:(36*B*C),6]<-rep(c(rep("norm",36),rep("t2",36),rep("nasimm",36)),B)
dataset[((36*B*C)+1):(96*B*C),6]<-rep(c(rep("norm",60),rep("t2",60),
rep("nasimm",60)),B)
dataset[((96*B*C)+1):k,6]<-rep(c(rep("norm",120),rep("t2",120),
rep("nasimm",120)),B)
dataset[1:(36*B*C),7]<-rep(c(rep(c(rep("S1",18),rep("S2",18)),3)),B)
dataset[((36*B*C)+1):(96*B*C),7]<-rep(c(rep(c(rep("S1",30),rep("S2",30)),3)),B)
dataset[((96*B*C)+1):k,7]<-rep(c(rep(c(rep("S1",60),rep("S2",60)),3)),B)
}
if (C==4) {
dataset[1:(36*B*C),5]<-rep(c(rep(1,6),rep(2,6),rep(3,6),rep(4,6)),(1.5*B*C))
dataset[((36*B*C)+1):(96*B*C),5]<-rep(c(rep(1,10),rep(2,10),rep(3,10),
rep(4,10)),(1.5*B*C))
dataset[((96*B*C)+1):k,5]<-rep(c(rep(1,20),rep(2,20),rep(3,20),rep(4,20)),
(1.5*B*C))
dataset<-as.data.frame(dataset)
dataset[1:(36*B*C),6]<-rep(c(rep("norm",48),rep("t2",48),rep("nasimm",48)),B)
dataset[((36*B*C)+1):(96*B*C),6]<-rep(c(rep("norm",80),rep("t2",80),
rep("nasimm",80)),B)
dataset[((96*B*C)+1):k,6]<-rep(c(rep("norm",160),rep("t2",160),
rep("nasimm",160)),B)
dataset[1:(36*B*C),7]<-rep(c(rep(c(rep("S1",24),rep("S2",24)),3)),B)
```

```
dataset[((36*B*C)+1):(96*B*C),7]<-rep(c(rep(c(rep("S1",40),rep("S2",40)),3)),B)
dataset[((96*B*C)+1):k,7]<-rep(c(rep(c(rep("S1",80),rep("S2",80)),3)),B)
}
if (C==5) {
dataset[1:(36*B*C),5]<-rep(c(rep(1,6),rep(2,6),rep(3,6),rep(4,6),rep(5,6)),
(1.2*B*C))
dataset[((36*B*C)+1):(96*B*C),5]<-rep(c(rep(1,10),rep(2,10),rep(3,10),
rep(4,10),rep(5,10)),(1.2*B*C))
dataset[((96*B*C)+1):k,5]<-rep(c(rep(1,20),rep(2,20),rep(3,20),rep(4,20),
rep(5,20)),(1.2*B*C))
dataset<-as.data.frame(dataset)
dataset[1:(36*B*C),6]<-rep(c(rep("norm",60),rep("t2",60),rep("nasimm",60)),B)
dataset[((36*B*C)+1):(96*B*C),6]<-rep(c(rep("norm",100),rep("t2",100),
rep("nasimm",100)),B)
dataset[((96*B*C)+1):k,6]<-rep(c(rep("norm",200),rep("t2",200),
rep("nasimm",200)),B)
dataset[1:(36*B*C),7]<-rep(c(rep(c(rep("S1",30),rep("S2",30)),3)),B)
dataset[((36*B*C)+1):(96*B*C),7]<-rep(c(rep(c(rep("S1",50),rep("S2",50)),3)),B)
dataset[((96*B*C)+1):k,7]<-rep(c(rep(c(rep("S1",100),rep("S2",100)),3)),B)
}
for (j in 1:B){
dataset[((j-1)*36*C+1):((j-1)*36*C+36*C),1]<-j
dataset[((j-1)*60*C+1+36*B*C):((j-1)*60*C+60*C+36*B*C),1]<-j
dataset[((j-1)*120*C+1+96*B*C):((j-1)*120*C+120*C+96*B*C),1]<-j
}

for (i in 1:B){
if (C==2){
dataset[(72*(i-1)+1):(72*(i-1)+6),8:10]<-rmvnorm(6,mean=mu[1,],sigma=S1)
dataset[(72*(i-1)+7):(72*(i-1)+12),8:10]<-rmvnorm(6,mean=mu[2,],sigma=S1)
dataset[(72*(i-1)+13):(72*(i-1)+18),8:10]<-rmvnorm(6,mean=mu[1,],sigma=S2)
```

```
dataset[(72*(i-1)+19):(72*(i-1)+24),8:10]<-rmvnorm(6,mean=mu[2,],sigma=S2)
dataset[(72*(i-1)+25):(72*(i-1)+30),8:10]<-rmvt(6,sigma=S1,df=2)+mu[1,]
dataset[(72*(i-1)+31):(72*(i-1)+36),8:10]<-rmvt(6,sigma=S1,df=2)+mu[2,]
dataset[(72*(i-1)+37):(72*(i-1)+42),8:10]<-rmvt(6,sigma=S2,df=2)+mu[1,]
dataset[(72*(i-1)+43):(72*(i-1)+48),8:10]<-rmvt(6,sigma=S2,df=2)+mu[2,]
dataset[(72*(i-1)+49):(72*(i-1)+54),8:10]<-rmsn(6,xi=mu[1,],Omega=S1,
alpha=rep(5,p))
dataset[(72*(i-1)+55):(72*(i-1)+60),8:10]<-rmsn(6,xi=mu[2,],Omega=S1,
alpha=rep(5,p))
dataset[(72*(i-1)+61):(72*(i-1)+66),8:10]<-rmsn(6,xi=mu[1,],Omega=S2,
alpha=rep(5,p))
dataset[(72*(i-1)+67):(72*(i-1)+72),8:10]<-rmsn(6,xi=mu[2,],Omega=S2,
alpha=rep(5,p))

dataset[(120*(i-1)+1+36*B*C):(120*(i-1)+10+36*B*C),8:10]<-rmvnorm(10,
mean=mu[1,],sigma=S1)
dataset[(120*(i-1)+11+36*B*C):(120*(i-1)+20+36*B*C),8:10]<-rmvnorm(10,
mean=mu[2,],sigma=S1)
dataset[(120*(i-1)+21+36*B*C):(120*(i-1)+30+36*B*C),8:10]<-rmvnorm(10,
mean=mu[1,],sigma=S2)
dataset[(120*(i-1)+31+36*B*C):(120*(i-1)+40+36*B*C),8:10]<-rmvnorm(10,
mean=mu[2,],sigma=S2)
dataset[(120*(i-1)+41+36*B*C):(120*(i-1)+50+36*B*C),8:10]<-rmvt(10,
sigma=S1,df=2)+mu[1,]
dataset[(120*(i-1)+51+36*B*C):(120*(i-1)+60+36*B*C),8:10]<-rmvt(10,
sigma=S1,df=2)+mu[2,]
dataset[(120*(i-1)+61+36*B*C):(120*(i-1)+70+36*B*C),8:10]<-rmvt(10,
sigma=S2,df=2)+mu[1,]
dataset[(120*(i-1)+71+36*B*C):(120*(i-1)+80+36*B*C),8:10]<-rmvt(10,
sigma=S2,df=2)+mu[2,]
dataset[(120*(i-1)+81+36*B*C):(120*(i-1)+90+36*B*C),8:10]<-rmsn(10,
xi=mu[1,],Omega=S1,alpha=rep(5,p))
dataset[(120*(i-1)+91+36*B*C):(120*(i-1)+100+36*B*C),8:10]<-rmsn(10,
xi=mu[2,],Omega=S1,alpha=rep(5,p))
dataset[(120*(i-1)+101+36*B*C):(120*(i-1)+110+36*B*C),8:10]<-rmsn(10,
xi=mu[1,],Omega=S2,alpha=rep(5,p))
dataset[(120*(i-1)+111+36*B*C):(120*(i-1)+120+36*B*C),8:10]<-rmsn(10,
xi=mu[2,],Omega=S2,alpha=rep(5,p))

dataset[(240*(i-1)+1+96*B*C):(240*(i-1)+20+96*B*C),8:10]<-rmvnorm(20,
mean=mu[1,],sigma=S1)
dataset[(240*(i-1)+21+96*B*C):(240*(i-1)+40+96*B*C),8:10]<-rmvnorm(20,
mean=mu[2,],sigma=S1)
dataset[(240*(i-1)+41+96*B*C):(240*(i-1)+60+96*B*C),8:10]<-rmvnorm(20,
mean=mu[1,],sigma=S2)
dataset[(240*(i-1)+61+96*B*C):(240*(i-1)+80+96*B*C),8:10]<-rmvnorm(20,
mean=mu[2,],sigma=S2)
dataset[(240*(i-1)+81+96*B*C):(240*(i-1)+100+96*B*C),8:10]<-rmvt(20,
sigma=S1,df=2)+mu[1,]
dataset[(240*(i-1)+101+96*B*C):(240*(i-1)+120+96*B*C),8:10]<-rmvt(20,
sigma=S1,df=2)+mu[2,]
dataset[(240*(i-1)+121+96*B*C):(240*(i-1)+140+96*B*C),8:10]<-rmvt(20,
sigma=S2,df=2)+mu[1,]
dataset[(240*(i-1)+141+96*B*C):(240*(i-1)+160+96*B*C),8:10]<-rmvt(20,
sigma=S2,df=2)+mu[2,]
dataset[(240*(i-1)+161+96*B*C):(240*(i-1)+180+96*B*C),8:10]<-rmsn(20,
xi=mu[1,],Omega=S1,alpha=rep(5,p))
dataset[(240*(i-1)+181+96*B*C):(240*(i-1)+200+96*B*C),8:10]<-rmsn(20,
xi=mu[2,],Omega=S1,alpha=rep(5,p))
dataset[(240*(i-1)+201+96*B*C):(240*(i-1)+220+96*B*C),8:10]<-rmsn(20,
xi=mu[1,],Omega=S2,alpha=rep(5,p))
dataset[(240*(i-1)+221+96*B*C):(240*(i-1)+240+96*B*C),8:10]<-rmsn(20,
xi=mu[2,],Omega=S2,alpha=rep(5,p))
}

if (C==3){
dataset[(108*(i-1)+1):(108*(i-1)+6),8:10]<-rmvnorm(6,mean=mu[1,],sigma=S1)
dataset[(108*(i-1)+7):(108*(i-1)+12),8:10]<-rmvnorm(6,mean=mu[2,],sigma=S1)
dataset[(108*(i-1)+13):(108*(i-1)+18),8:10]<-rmvnorm(6,mean=mu[3,],sigma=S1)
dataset[(108*(i-1)+19):(108*(i-1)+24),8:10]<-rmvnorm(6,mean=mu[1,],sigma=S2)
dataset[(108*(i-1)+25):(108*(i-1)+30),8:10]<-rmvnorm(6,mean=mu[2,],sigma=S2)
dataset[(108*(i-1)+31):(108*(i-1)+36),8:10]<-rmvnorm(6,mean=mu[3,],sigma=S2)
dataset[(108*(i-1)+37):(108*(i-1)+42),8:10]<-rmvt(6,sigma=S1,df=2)+mu[1,]
dataset[(108*(i-1)+43):(108*(i-1)+48),8:10]<-rmvt(6,sigma=S1,df=2)+mu[2,]
dataset[(108*(i-1)+49):(108*(i-1)+54),8:10]<-rmvt(6,sigma=S1,df=2)+mu[3,]
dataset[(108*(i-1)+55):(108*(i-1)+60),8:10]<-rmvt(6,sigma=S2,df=2)+mu[1,]
dataset[(108*(i-1)+61):(108*(i-1)+66),8:10]<-rmvt(6,sigma=S2,df=2)+mu[2,]
dataset[(108*(i-1)+67):(108*(i-1)+72),8:10]<-rmvt(6,sigma=S2,df=2)+mu[3,]
dataset[(108*(i-1)+73):(108*(i-1)+78),8:10]<-rmsn(6,xi=mu[1,],Omega=S1,
alpha=rep(5,p))
dataset[(108*(i-1)+79):(108*(i-1)+84),8:10]<-rmsn(6,xi=mu[2,],Omega=S1,
alpha=rep(5,p))
dataset[(108*(i-1)+85):(108*(i-1)+90),8:10]<-rmsn(6,xi=mu[3,],Omega=S1,
alpha=rep(5,p))
dataset[(108*(i-1)+91):(108*(i-1)+96),8:10]<-rmsn(6,xi=mu[1,],Omega=S2,
alpha=rep(5,p))
dataset[(108*(i-1)+97):(108*(i-1)+102),8:10]<-rmsn(6,xi=mu[2,],Omega=S2,
alpha=rep(5,p))
dataset[(108*(i-1)+103):(108*(i-1)+108),8:10]<-rmsn(6,xi=mu[3,],Omega=S2,
alpha=rep(5,p))

dataset[(180*(i-1)+1+36*B*C):(180*(i-1)+10+36*B*C),8:10]<-rmvnorm(10,
mean=mu[1,],sigma=S1)
```

```
dataset[(180*(i-1)+11+36*B*C):(180*(i-1)+20+36*B*C),8:10]<-rmvnorm(10,
mean=mu[2,],sigma=S1)
dataset[(180*(i-1)+21+36*B*C):(180*(i-1)+30+36*B*C),8:10]<-rmvnorm(10,
mean=mu[3,],sigma=S1)
dataset[(180*(i-1)+31+36*B*C):(180*(i-1)+40+36*B*C),8:10]<-rmvnorm(10,
mean=mu[1,],sigma=S2)
dataset[(180*(i-1)+41+36*B*C):(180*(i-1)+50+36*B*C),8:10]<-rmvnorm(10,
mean=mu[2,],sigma=S2)
dataset[(180*(i-1)+51+36*B*C):(180*(i-1)+60+36*B*C),8:10]<-rmvnorm(10,
mean=mu[3,],sigma=S2)
dataset[(180*(i-1)+61+36*B*C):(180*(i-1)+70+36*B*C),8:10]<-rmvt(10,
sigma=S1,df=2)+mu[1,]
dataset[(180*(i-1)+71+36*B*C):(180*(i-1)+80+36*B*C),8:10]<-rmvt(10,
sigma=S1,df=2)+mu[2,]
dataset[(180*(i-1)+81+36*B*C):(180*(i-1)+90+36*B*C),8:10]<-rmvt(10,
sigma=S1,df=2)+mu[3,]
dataset[(180*(i-1)+91+36*B*C):(180*(i-1)+100+36*B*C),8:10]<-rmvt(10,
sigma=S2,df=2)+mu[1,]
dataset[(180*(i-1)+101+36*B*C):(180*(i-1)+110+36*B*C),8:10]<-rmvt(10,
sigma=S2,df=2)+mu[2,]
dataset[(180*(i-1)+111+36*B*C):(180*(i-1)+120+36*B*C),8:10]<-rmvt(10,
sigma=S2,df=2)+mu[3,]
dataset[(180*(i-1)+121+36*B*C):(180*(i-1)+130+36*B*C),8:10]<-rmsn(10,
xi=mu[1,],Omega=S1,alpha=rep(5,p))
dataset[(180*(i-1)+131+36*B*C):(180*(i-1)+140+36*B*C),8:10]<-rmsn(10,
xi=mu[2,],Omega=S1,alpha=rep(5,p))
dataset[(180*(i-1)+141+36*B*C):(180*(i-1)+150+36*B*C),8:10]<-rmsn(10,
xi=mu[3,],Omega=S1,alpha=rep(5,p))
dataset[(180*(i-1)+151+36*B*C):(180*(i-1)+160+36*B*C),8:10]<-rmsn(10,
xi=mu[1,],Omega=S2,alpha=rep(5,p))
dataset[(180*(i-1)+161+36*B*C):(180*(i-1)+170+36*B*C),8:10]<-rmsn(10,
xi=mu[2,],Omega=S2,alpha=rep(5,p))
dataset[(180*(i-1)+171+36*B*C):(180*(i-1)+180+36*B*C),8:10]<-rmsn(10,
xi=mu[3,],Omega=S2,alpha=rep(5,p))

dataset[(360*(i-1)+1+96*B*C):(360*(i-1)+20+96*B*C),8:10]<-rmvnorm(20,
mean=mu[1,],sigma=S1)
dataset[(360*(i-1)+21+96*B*C):(360*(i-1)+40+96*B*C),8:10]<-rmvnorm(20,
mean=mu[2,],sigma=S1)
dataset[(360*(i-1)+41+96*B*C):(360*(i-1)+60+96*B*C),8:10]<-rmvnorm(20,
mean=mu[3,],sigma=S1)
dataset[(360*(i-1)+61+96*B*C):(360*(i-1)+80+96*B*C),8:10]<-rmvnorm(20,
mean=mu[1,],sigma=S2)
dataset[(360*(i-1)+81+96*B*C):(360*(i-1)+100+96*B*C),8:10]<-rmvnorm(20,
mean=mu[2,],sigma=S2)
dataset[(360*(i-1)+101+96*B*C):(360*(i-1)+120+96*B*C),8:10]<-rmvnorm(20,
mean=mu[3,],sigma=S2)
dataset[(360*(i-1)+121+96*B*C):(360*(i-1)+140+96*B*C),8:10]<-rmvt(20,
sigma=S1,df=2)+mu[1,]
dataset[(360*(i-1)+141+96*B*C):(360*(i-1)+160+96*B*C),8:10]<-rmvt(20,
sigma=S1,df=2)+mu[2,]
dataset[(360*(i-1)+161+96*B*C):(360*(i-1)+180+96*B*C),8:10]<-rmvt(20,
sigma=S1,df=2)+mu[3,]
dataset[(360*(i-1)+181+96*B*C):(360*(i-1)+200+96*B*C),8:10]<-rmvt(20,
sigma=S2,df=2)+mu[1,]
dataset[(360*(i-1)+201+96*B*C):(360*(i-1)+220+96*B*C),8:10]<-rmvt(20,
sigma=S2,df=2)+mu[2,]
dataset[(360*(i-1)+221+96*B*C):(360*(i-1)+240+96*B*C),8:10]<-rmvt(20,
sigma=S2,df=2)+mu[3,]
dataset[(360*(i-1)+241+96*B*C):(360*(i-1)+260+96*B*C),8:10]<-rmsn(20,
xi=mu[1,],Omega=S1,alpha=rep(5,p))
dataset[(360*(i-1)+261+96*B*C):(360*(i-1)+280+96*B*C),8:10]<-rmsn(20,
xi=mu[2,],Omega=S1,alpha=rep(5,p))
dataset[(360*(i-1)+281+96*B*C):(360*(i-1)+300+96*B*C),8:10]<-rmsn(20,
xi=mu[3,],Omega=S1,alpha=rep(5,p))
dataset[(360*(i-1)+301+96*B*C):(360*(i-1)+320+96*B*C),8:10]<-rmsn(20,
xi=mu[1,],Omega=S2,alpha=rep(5,p))
dataset[(360*(i-1)+321+96*B*C):(360*(i-1)+340+96*B*C),8:10]<-rmsn(20,
xi=mu[2,],Omega=S2,alpha=rep(5,p))
dataset[(360*(i-1)+341+96*B*C):(360*(i-1)+360+96*B*C),8:10]<-rmsn(20,
xi=mu[3,],Omega=S2,alpha=rep(5,p))
}

if (C==4){
dataset[(144*(i-1)+1):(144*(i-1)+6),8:10]<-rmvnorm(6,mean=mu[1,],sigma=S1)
dataset[(144*(i-1)+7):(144*(i-1)+12),8:10]<-rmvnorm(6,mean=mu[2,],sigma=S1)
dataset[(144*(i-1)+13):(144*(i-1)+18),8:10]<-rmvnorm(6,mean=mu[3,],sigma=S1)
dataset[(144*(i-1)+19):(144*(i-1)+24),8:10]<-rmvnorm(6,mean=mu[4,],sigma=S1)
dataset[(144*(i-1)+25):(144*(i-1)+30),8:10]<-rmvnorm(6,mean=mu[1,],sigma=S2)
dataset[(144*(i-1)+31):(144*(i-1)+36),8:10]<-rmvnorm(6,mean=mu[2,],sigma=S2)
dataset[(144*(i-1)+37):(144*(i-1)+42),8:10]<-rmvnorm(6,mean=mu[3,],sigma=S2)
dataset[(144*(i-1)+43):(144*(i-1)+48),8:10]<-rmvnorm(6,mean=mu[4,],sigma=S2)
dataset[(144*(i-1)+49):(144*(i-1)+54),8:10]<-rmvt(6,sigma=S1,df=2)+mu[1,]
dataset[(144*(i-1)+55):(144*(i-1)+60),8:10]<-rmvt(6,sigma=S1,df=2)+mu[2,]
dataset[(144*(i-1)+61):(144*(i-1)+66),8:10]<-rmvt(6,sigma=S1,df=2)+mu[3,]
dataset[(144*(i-1)+67):(144*(i-1)+72),8:10]<-rmvt(6,sigma=S1,df=2)+mu[4,]
dataset[(144*(i-1)+73):(144*(i-1)+78),8:10]<-rmvt(6,sigma=S2,df=2)+mu[1,]
dataset[(144*(i-1)+79):(144*(i-1)+84),8:10]<-rmvt(6,sigma=S2,df=2)+mu[2,]
dataset[(144*(i-1)+85):(144*(i-1)+90),8:10]<-rmvt(6,sigma=S2,df=2)+mu[3,]
dataset[(144*(i-1)+91):(144*(i-1)+96),8:10]<-rmvt(6,sigma=S2,df=2)+mu[4,]
dataset[(144*(i-1)+97):(144*(i-1)+102),8:10]<-rmsn(6,xi=mu[1,],Omega=S1,
alpha=rep(5,p))
dataset[(144*(i-1)+103):(144*(i-1)+108),8:10]<-rmsn(6,xi=mu[2,],Omega=S1,
alpha=rep(5,p))
dataset[(144*(i-1)+109):(144*(i-1)+114),8:10]<-rmsn(6,xi=mu[3,],Omega=S1,
alpha=rep(5,p))
dataset[(144*(i-1)+115):(144*(i-1)+120),8:10]<-rmsn(6,xi=mu[4,],Omega=S1,
alpha=rep(5,p))
dataset[(144*(i-1)+121):(144*(i-1)+126),8:10]<-rmsn(6,xi=mu[1,],Omega=S2,
alpha=rep(5,p))
dataset[(144*(i-1)+127):(144*(i-1)+132),8:10]<-rmsn(6,xi=mu[2,],Omega=S2,
alpha=rep(5,p))
dataset[(144*(i-1)+133):(144*(i-1)+138),8:10]<-rmsn(6,xi=mu[3,],Omega=S2,
alpha=rep(5,p))
dataset[(144*(i-1)+139):(144*(i-1)+144),8:10]<-rmsn(6,xi=mu[4,],Omega=S2,
alpha=rep(5,p))

dataset[(240*(i-1)+1+36*B*C):(240*(i-1)+10+36*B*C),8:10]<-rmvnorm(10,
mean=mu[1,],sigma=S1)
dataset[(240*(i-1)+11+36*B*C):(240*(i-1)+20+36*B*C),8:10]<-rmvnorm(10,
mean=mu[2,],sigma=S1)
dataset[(240*(i-1)+21+36*B*C):(240*(i-1)+30+36*B*C),8:10]<-rmvnorm(10,
mean=mu[3,],sigma=S1)
dataset[(240*(i-1)+31+36*B*C):(240*(i-1)+40+36*B*C),8:10]<-rmvnorm(10,
mean=mu[4,],sigma=S1)
dataset[(240*(i-1)+41+36*B*C):(240*(i-1)+50+36*B*C),8:10]<-rmvnorm(10,
mean=mu[1,],sigma=S2)
dataset[(240*(i-1)+51+36*B*C):(240*(i-1)+60+36*B*C),8:10]<-rmvnorm(10,
mean=mu[2,],sigma=S2)
dataset[(240*(i-1)+61+36*B*C):(240*(i-1)+70+36*B*C),8:10]<-rmvnorm(10,
mean=mu[3,],sigma=S2)
dataset[(240*(i-1)+71+36*B*C):(240*(i-1)+80+36*B*C),8:10]<-rmvnorm(10,
mean=mu[4,],sigma=S2)
dataset[(240*(i-1)+81+36*B*C):(240*(i-1)+90+36*B*C),8:10]<-rmvt(10,
sigma=S1,df=2)+mu[1,]
dataset[(240*(i-1)+91+36*B*C):(240*(i-1)+100+36*B*C),8:10]<-rmvt(10,
sigma=S1,df=2)+mu[2,]
dataset[(240*(i-1)+101+36*B*C):(240*(i-1)+110+36*B*C),8:10]<-rmvt(10,
sigma=S1,df=2)+mu[3,]
dataset[(240*(i-1)+111+36*B*C):(240*(i-1)+120+36*B*C),8:10]<-rmvt(10,
sigma=S1,df=2)+mu[4,]
dataset[(240*(i-1)+121+36*B*C):(240*(i-1)+130+36*B*C),8:10]<-rmvt(10,
sigma=S2,df=2)+mu[1,]
dataset[(240*(i-1)+131+36*B*C):(240*(i-1)+140+36*B*C),8:10]<-rmvt(10,
sigma=S2,df=2)+mu[2,]
dataset[(240*(i-1)+141+36*B*C):(240*(i-1)+150+36*B*C),8:10]<-rmvt(10,
sigma=S2,df=2)+mu[3,]
dataset[(240*(i-1)+151+36*B*C):(240*(i-1)+160+36*B*C),8:10]<-rmvt(10,
sigma=S2,df=2)+mu[4,]
dataset[(240*(i-1)+161+36*B*C):(240*(i-1)+170+36*B*C),8:10]<-rmsn(10,
xi=mu[1,],Omega=S1,alpha=rep(5,p))
dataset[(240*(i-1)+171+36*B*C):(240*(i-1)+180+36*B*C),8:10]<-rmsn(10,
xi=mu[2,],Omega=S1,alpha=rep(5,p))
dataset[(240*(i-1)+181+36*B*C):(240*(i-1)+190+36*B*C),8:10]<-rmsn(10,
xi=mu[3,],Omega=S1,alpha=rep(5,p))
dataset[(240*(i-1)+191+36*B*C):(240*(i-1)+200+36*B*C),8:10]<-rmsn(10,
xi=mu[4,],Omega=S1,alpha=rep(5,p))
dataset[(240*(i-1)+201+36*B*C):(240*(i-1)+210+36*B*C),8:10]<-rmsn(10,
xi=mu[1,],Omega=S2,alpha=rep(5,p))
dataset[(240*(i-1)+211+36*B*C):(240*(i-1)+220+36*B*C),8:10]<-rmsn(10,
xi=mu[2,],Omega=S2,alpha=rep(5,p))
dataset[(240*(i-1)+221+36*B*C):(240*(i-1)+230+36*B*C),8:10]<-rmsn(10,
xi=mu[3,],Omega=S2,alpha=rep(5,p))
dataset[(240*(i-1)+231+36*B*C):(240*(i-1)+240+36*B*C),8:10]<-rmsn(10,
xi=mu[4,],Omega=S2,alpha=rep(5,p))
```

```
dataset[(480*(i-1)+1+96*B*C):(480*(i-1)+20+96*B*C),8:10]<-rmvnorm(20,
mean=mu[1,],sigma=S1)
dataset[(480*(i-1)+21+96*B*C):(480*(i-1)+40+96*B*C),8:10]<-rmvnorm(20,
mean=mu[2,],sigma=S1)
dataset[(480*(i-1)+41+96*B*C):(480*(i-1)+60+96*B*C),8:10]<-rmvnorm(20,
mean=mu[3,],sigma=S1)
dataset[(480*(i-1)+61+96*B*C):(480*(i-1)+80+96*B*C),8:10]<-rmvnorm(20,
mean=mu[4,],sigma=S1)
dataset[(480*(i-1)+81+96*B*C):(480*(i-1)+100+96*B*C),8:10]<-rmvnorm(20,
mean=mu[1,],sigma=S2)
dataset[(480*(i-1)+101+96*B*C):(480*(i-1)+120+96*B*C),8:10]<-rmvnorm(20,
mean=mu[2,],sigma=S2)
dataset[(480*(i-1)+121+96*B*C):(480*(i-1)+140+96*B*C),8:10]<-rmvnorm(20,
mean=mu[3,],sigma=S2)
dataset[(480*(i-1)+141+96*B*C):(480*(i-1)+160+96*B*C),8:10]<-rmvnorm(20,
mean=mu[4,],sigma=S2)
dataset[(480*(i-1)+161+96*B*C):(480*(i-1)+180+96*B*C),8:10]<-rmvt(20,
sigma=S1,df=2)+mu[1,]
dataset[(480*(i-1)+181+96*B*C):(480*(i-1)+200+96*B*C),8:10]<-rmvt(20,
sigma=S1,df=2)+mu[2,]
dataset[(480*(i-1)+201+96*B*C):(480*(i-1)+220+96*B*C),8:10]<-rmvt(20,
sigma=S1,df=2)+mu[3,]
dataset[(480*(i-1)+221+96*B*C):(480*(i-1)+240+96*B*C),8:10]<-rmvt(20,
sigma=S1,df=2)+mu[4,]
dataset[(480*(i-1)+241+96*B*C):(480*(i-1)+260+96*B*C),8:10]<-rmvt(20,
sigma=S2,df=2)+mu[1,]
dataset[(480*(i-1)+261+96*B*C):(480*(i-1)+280+96*B*C),8:10]<-rmvt(20,
sigma=S2,df=2)+mu[2,]
dataset[(480*(i-1)+281+96*B*C):(480*(i-1)+300+96*B*C),8:10]<-rmvt(20,
sigma=S2,df=2)+mu[3,]
```

```
dataset[(480*(i-1)+301+96*B*C):(480*(i-1)+320+96*B*C),8:10]<-rmvt(20,
sigma=S2,df=2)+mu[4,]
dataset[(480*(i-1)+321+96*B*C):(480*(i-1)+340+96*B*C),8:10]<-rmsn(20,
xi=mu[1,],Omega=S1,alpha=rep(5,p))
dataset[(480*(i-1)+341+96*B*C):(480*(i-1)+360+96*B*C),8:10]<-rmsn(20,
xi=mu[2,],Omega=S1,alpha=rep(5,p))
dataset[(480*(i-1)+361+96*B*C):(480*(i-1)+380+96*B*C),8:10]<-rmsn(20,
xi=mu[3,],Omega=S1,alpha=rep(5,p))
dataset[(480*(i-1)+381+96*B*C):(480*(i-1)+400+96*B*C),8:10]<-rmsn(20,
xi=mu[4,],Omega=S1,alpha=rep(5,p))
dataset[(480*(i-1)+401+96*B*C):(480*(i-1)+420+96*B*C),8:10]<-rmsn(20,
xi=mu[1,],Omega=S2,alpha=rep(5,p))
dataset[(480*(i-1)+421+96*B*C):(480*(i-1)+440+96*B*C),8:10]<-rmsn(20,
xi=mu[2,],Omega=S2,alpha=rep(5,p))
dataset[(480*(i-1)+441+96*B*C):(480*(i-1)+460+96*B*C),8:10]<-rmsn(20,
xi=mu[3,],Omega=S2,alpha=rep(5,p))
dataset[(480*(i-1)+461+96*B*C):(480*(i-1)+480+96*B*C),8:10]<-rmsn(20,
xi=mu[4,],Omega=S2,alpha=rep(5,p))
}

if (C==5){
dataset[(180*(i-1)+1):(180*(i-1)+6),8:10]<-rmvnorm(6,mean=mu[1,],sigma=S1)
dataset[(180*(i-1)+7):(180*(i-1)+12),8:10]<-rmvnorm(6,mean=mu[2,],sigma=S1)
dataset[(180*(i-1)+13):(180*(i-1)+18),8:10]<-rmvnorm(6,mean=mu[3,],sigma=S1)
dataset[(180*(i-1)+19):(180*(i-1)+24),8:10]<-rmvnorm(6,mean=mu[4,],sigma=S1)
dataset[(180*(i-1)+25):(180*(i-1)+30),8:10]<-rmvnorm(6,mean=mu[5,],sigma=S1)
dataset[(180*(i-1)+31):(180*(i-1)+36),8:10]<-rmvnorm(6,mean=mu[1,],sigma=S2)
dataset[(180*(i-1)+37):(180*(i-1)+42),8:10]<-rmvnorm(6,mean=mu[2,],sigma=S2)
dataset[(180*(i-1)+43):(180*(i-1)+48),8:10]<-rmvnorm(6,mean=mu[3,],sigma=S2)
dataset[(180*(i-1)+49):(180*(i-1)+54),8:10]<-rmvnorm(6,mean=mu[4,],sigma=S2)
dataset[(180*(i-1)+55):(180*(i-1)+60),8:10]<-rmvnorm(6,mean=mu[5,],sigma=S2)
```

```
dataset[(180*(i-1)+61):(180*(i-1)+66),8:10]<-rmvt(6,sigma=S1,df=2)+mu[1,]
dataset[(180*(i-1)+67):(180*(i-1)+72),8:10]<-rmvt(6,sigma=S1,df=2)+mu[2,]
dataset[(180*(i-1)+73):(180*(i-1)+78),8:10]<-rmvt(6,sigma=S1,df=2)+mu[3,]
dataset[(180*(i-1)+79):(180*(i-1)+84),8:10]<-rmvt(6,sigma=S1,df=2)+mu[4,]
dataset[(180*(i-1)+85):(180*(i-1)+90),8:10]<-rmvt(6,sigma=S1,df=2)+mu[5,]
dataset[(180*(i-1)+91):(180*(i-1)+96),8:10]<-rmvt(6,sigma=S2,df=2)+mu[1,]
dataset[(180*(i-1)+97):(180*(i-1)+102),8:10]<-rmvt(6,sigma=S2,df=2)+mu[2,]
dataset[(180*(i-1)+103):(180*(i-1)+108),8:10]<-rmvt(6,sigma=S2,df=2)+mu[3,]
dataset[(180*(i-1)+109):(180*(i-1)+114),8:10]<-rmvt(6,sigma=S2,df=2)+mu[4,]
dataset[(180*(i-1)+115):(180*(i-1)+120),8:10]<-rmvt(6,sigma=S2,df=2)+mu[5,]
dataset[(180*(i-1)+121):(180*(i-1)+126),8:10]<-rmsn(6,xi=mu[1,],Omega=S1,
alpha=rep(5,p))
dataset[(180*(i-1)+127):(180*(i-1)+132),8:10]<-rmsn(6,xi=mu[2,],Omega=S1,
alpha=rep(5,p))
dataset[(180*(i-1)+133):(180*(i-1)+138),8:10]<-rmsn(6,xi=mu[3,],Omega=S1,
alpha=rep(5,p))
dataset[(180*(i-1)+139):(180*(i-1)+144),8:10]<-rmsn(6,xi=mu[4,],Omega=S1,
alpha=rep(5,p))
dataset[(180*(i-1)+145):(180*(i-1)+150),8:10]<-rmsn(6,xi=mu[5,],Omega=S1,
alpha=rep(5,p))
dataset[(180*(i-1)+151):(180*(i-1)+156),8:10]<-rmsn(6,xi=mu[1,],Omega=S2,
alpha=rep(5,p))
dataset[(180*(i-1)+157):(180*(i-1)+162),8:10]<-rmsn(6,xi=mu[2,],Omega=S2,
alpha=rep(5,p))
dataset[(180*(i-1)+163):(180*(i-1)+168),8:10]<-rmsn(6,xi=mu[3,],Omega=S2,
alpha=rep(5,p))
dataset[(180*(i-1)+169):(180*(i-1)+174),8:10]<-rmsn(6,xi=mu[4,],Omega=S2,
alpha=rep(5,p))
dataset[(180*(i-1)+175):(180*(i-1)+180),8:10]<-rmsn(6,xi=mu[5,],Omega=S2,
alpha=rep(5,p))
```

```
dataset[(300*(i-1)+1+36*B*C):(300*(i-1)+10+36*B*C),8:10]<-rmvnorm(10,
mean=mu[1,],sigma=S1)
dataset[(300*(i-1)+11+36*B*C):(300*(i-1)+20+36*B*C),8:10]<-rmvnorm(10,
mean=mu[2,],sigma=S1)
dataset[(300*(i-1)+21+36*B*C):(300*(i-1)+30+36*B*C),8:10]<-rmvnorm(10,
mean=mu[3,],sigma=S1)
dataset[(300*(i-1)+31+36*B*C):(300*(i-1)+40+36*B*C),8:10]<-rmvnorm(10,
mean=mu[4,],sigma=S1)
dataset[(300*(i-1)+41+36*B*C):(300*(i-1)+50+36*B*C),8:10]<-rmvnorm(10,
mean=mu[5,],sigma=S1)
dataset[(300*(i-1)+51+36*B*C):(300*(i-1)+60+36*B*C),8:10]<-rmvnorm(10,
mean=mu[1,],sigma=S2)
dataset[(300*(i-1)+61+36*B*C):(300*(i-1)+70+36*B*C),8:10]<-rmvnorm(10,
mean=mu[2,],sigma=S2)
dataset[(300*(i-1)+71+36*B*C):(300*(i-1)+80+36*B*C),8:10]<-rmvnorm(10,
mean=mu[3,],sigma=S2)
dataset[(300*(i-1)+81+36*B*C):(300*(i-1)+90+36*B*C),8:10]<-rmvnorm(10,
mean=mu[4,],sigma=S2)
dataset[(300*(i-1)+91+36*B*C):(300*(i-1)+100+36*B*C),8:10]<-rmvnorm(10,
mean=mu[5,],sigma=S2)
dataset[(300*(i-1)+101+36*B*C):(300*(i-1)+110+36*B*C),8:10]<-rmvt(10,
sigma=S1,df=2)+mu[1,]
dataset[(300*(i-1)+111+36*B*C):(300*(i-1)+120+36*B*C),8:10]<-rmvt(10,
sigma=S1,df=2)+mu[2,]
dataset[(300*(i-1)+121+36*B*C):(300*(i-1)+130+36*B*C),8:10]<-rmvt(10,
sigma=S1,df=2)+mu[3,]
dataset[(300*(i-1)+131+36*B*C):(300*(i-1)+140+36*B*C),8:10]<-rmvt(10,
sigma=S1,df=2)+mu[4,]
dataset[(300*(i-1)+141+36*B*C):(300*(i-1)+150+36*B*C),8:10]<-rmvt(10,
sigma=S1,df=2)+mu[5,]
dataset[(300*(i-1)+151+36*B*C):(300*(i-1)+160+36*B*C),8:10]<-rmvt(10,
sigma=S2,df=2)+mu[1,]
dataset[(300*(i-1)+161+36*B*C):(300*(i-1)+170+36*B*C),8:10]<-rmvt(10,
sigma=S2,df=2)+mu[2,]
dataset[(300*(i-1)+171+36*B*C):(300*(i-1)+180+36*B*C),8:10]<-rmvt(10,
sigma=S2,df=2)+mu[3,]
dataset[(300*(i-1)+181+36*B*C):(300*(i-1)+190+36*B*C),8:10]<-rmvt(10,
sigma=S2,df=2)+mu[4,]
dataset[(300*(i-1)+191+36*B*C):(300*(i-1)+200+36*B*C),8:10]<-rmvt(10,
sigma=S2,df=2)+mu[5,]
dataset[(300*(i-1)+201+36*B*C):(300*(i-1)+210+36*B*C),8:10]<-rmsn(10,
xi=mu[1,],Omega=S1,alpha=rep(5,p))
dataset[(300*(i-1)+211+36*B*C):(300*(i-1)+220+36*B*C),8:10]<-rmsn(10,
xi=mu[2,],Omega=S1,alpha=rep(5,p))
dataset[(300*(i-1)+221+36*B*C):(300*(i-1)+230+36*B*C),8:10]<-rmsn(10,
xi=mu[3,],Omega=S1,alpha=rep(5,p))
dataset[(300*(i-1)+231+36*B*C):(300*(i-1)+240+36*B*C),8:10]<-rmsn(10,
xi=mu[4,],Omega=S1,alpha=rep(5,p))
dataset[(300*(i-1)+241+36*B*C):(300*(i-1)+250+36*B*C),8:10]<-rmsn(10,
xi=mu[5,],Omega=S1,alpha=rep(5,p))
dataset[(300*(i-1)+251+36*B*C):(300*(i-1)+260+36*B*C),8:10]<-rmsn(10,
xi=mu[1,],Omega=S2,alpha=rep(5,p))
dataset[(300*(i-1)+261+36*B*C):(300*(i-1)+270+36*B*C),8:10]<-rmsn(10,
xi=mu[2,],Omega=S2,alpha=rep(5,p))
dataset[(300*(i-1)+271+36*B*C):(300*(i-1)+280+36*B*C),8:10]<-rmsn(10,
xi=mu[3,],Omega=S2,alpha=rep(5,p))
dataset[(300*(i-1)+281+36*B*C):(300*(i-1)+290+36*B*C),8:10]<-rmsn(10,
xi=mu[4,],Omega=S2,alpha=rep(5,p))
dataset[(300*(i-1)+291+36*B*C):(300*(i-1)+300+36*B*C),8:10]<-rmsn(10,
xi=mu[5,],Omega=S2,alpha=rep(5,p))

dataset[(600*(i-1)+1+96*B*C):(600*(i-1)+20+96*B*C),8:10]<-rmvnorm(20,
mean=mu[1,],sigma=S1)
dataset[(600*(i-1)+21+96*B*C):(600*(i-1)+40+96*B*C),8:10]<-rmvnorm(20,
mean=mu[2,],sigma=S1)
dataset[(600*(i-1)+41+96*B*C):(600*(i-1)+60+96*B*C),8:10]<-rmvnorm(20,
mean=mu[3,],sigma=S1)
dataset[(600*(i-1)+61+96*B*C):(600*(i-1)+80+96*B*C),8:10]<-rmvnorm(20,
mean=mu[4,],sigma=S1)
dataset[(600*(i-1)+81+96*B*C):(600*(i-1)+100+96*B*C),8:10]<-rmvnorm(20,
mean=mu[5,],sigma=S1)
dataset[(600*(i-1)+101+96*B*C):(600*(i-1)+120+96*B*C),8:10]<-rmvnorm(20,
mean=mu[1,],sigma=S2)
dataset[(600*(i-1)+121+96*B*C):(600*(i-1)+140+96*B*C),8:10]<-rmvnorm(20,
mean=mu[2,],sigma=S2)
dataset[(600*(i-1)+141+96*B*C):(600*(i-1)+160+96*B*C),8:10]<-rmvnorm(20,
mean=mu[3,],sigma=S2)
dataset[(600*(i-1)+161+96*B*C):(600*(i-1)+180+96*B*C),8:10]<-rmvnorm(20,
mean=mu[4,],sigma=S2)
dataset[(600*(i-1)+181+96*B*C):(600*(i-1)+200+96*B*C),8:10]<-rmvnorm(20,
mean=mu[5,],sigma=S2)
dataset[(600*(i-1)+201+96*B*C):(600*(i-1)+220+96*B*C),8:10]<-rmvt(20,
sigma=S1,df=2)+mu[1,]
dataset[(600*(i-1)+221+96*B*C):(600*(i-1)+240+96*B*C),8:10]<-rmvt(20,
sigma=S1,df=2)+mu[2,]
dataset[(600*(i-1)+241+96*B*C):(600*(i-1)+260+96*B*C),8:10]<-rmvt(20,
sigma=S1,df=2)+mu[3,]
dataset[(600*(i-1)+261+96*B*C):(600*(i-1)+280+96*B*C),8:10]<-rmvt(20,
sigma=S1,df=2)+mu[4,]
dataset[(600*(i-1)+281+96*B*C):(600*(i-1)+300+96*B*C),8:10]<-rmvt(20,
sigma=S1,df=2)+mu[5,]
dataset[(600*(i-1)+301+96*B*C):(600*(i-1)+320+96*B*C),8:10]<-rmvt(20,
sigma=S2,df=2)+mu[1,]
```

```
dataset[(600*(i-1)+321+96*B*C):(600*(i-1)+340+96*B*C),8:10]<-rmvt(20,
sigma=S2,df=2)+mu[2,]
dataset[(600*(i-1)+341+96*B*C):(600*(i-1)+360+96*B*C),8:10]<-rmvt(20,
sigma=S2,df=2)+mu[3,]
dataset[(600*(i-1)+361+96*B*C):(600*(i-1)+380+96*B*C),8:10]<-rmvt(20,
sigma=S2,df=2)+mu[4,]
dataset[(600*(i-1)+381+96*B*C):(600*(i-1)+400+96*B*C),8:10]<-rmvt(20,
sigma=S2,df=2)+mu[5,]
dataset[(600*(i-1)+401+96*B*C):(600*(i-1)+420+96*B*C),8:10]<-rmsn(20,
xi=mu[1,],Omega=S1,alpha=rep(5,p))
dataset[(600*(i-1)+421+96*B*C):(600*(i-1)+440+96*B*C),8:10]<-rmsn(20,
xi=mu[2,],Omega=S1,alpha=rep(5,p))
dataset[(600*(i-1)+441+96*B*C):(600*(i-1)+460+96*B*C),8:10]<-rmsn(20,
xi=mu[3,],Omega=S1,alpha=rep(5,p))
dataset[(600*(i-1)+461+96*B*C):(600*(i-1)+480+96*B*C),8:10]<-rmsn(20,
xi=mu[4,],Omega=S1,alpha=rep(5,p))
dataset[(600*(i-1)+481+96*B*C):(600*(i-1)+500+96*B*C),8:10]<-rmsn(20,
xi=mu[5,],Omega=S1,alpha=rep(5,p))
dataset[(600*(i-1)+501+96*B*C):(600*(i-1)+520+96*B*C),8:10]<-rmsn(20,
xi=mu[1,],Omega=S2,alpha=rep(5,p))
dataset[(600*(i-1)+521+96*B*C):(600*(i-1)+540+96*B*C),8:10]<-rmsn(20,
xi=mu[2,],Omega=S2,alpha=rep(5,p))
dataset[(600*(i-1)+541+96*B*C):(600*(i-1)+560+96*B*C),8:10]<-rmsn(20,
xi=mu[3,],Omega=S2,alpha=rep(5,p))
dataset[(600*(i-1)+561+96*B*C):(600*(i-1)+580+96*B*C),8:10]<-rmsn(20,
xi=mu[4,],Omega=S2,alpha=rep(5,p))
dataset[(600*(i-1)+581+96*B*C):(600*(i-1)+600+96*B*C),8:10]<-rmsn(20,
xi=mu[5,],Omega=S2,alpha=rep(5,p))
}
}
dataset[,8:10]<-round(dataset[,8:10])
```

```
for (i in 1:B){
dataset$X[(36*C*(i-1)+1):(36*C*(i-1)+36*C)]<-dataset$X[(36*C*(i-1)+1)
:(36*C*(i-1)+36*C)]+round(rep(runif(6,-1.5,1.5),6*C))
dataset$X[(60*C*(i-1)+1+36*C*B):(60*C*(i-1)+60*C+36*C*B)]<-
dataset$X[(60*C*(i-1)+1+36*C*B):(60*C*(i-1)+60*C+36*C*B)]+
round(rep(runif(10,-1.5,1.5),6*C))
dataset$X[(120*C*(i-1)+1+96*C*B):(120*C*(i-1)+120*C+96*C*B)]<-
dataset$X[(120*C*(i-1)+1+96*C*B):(120*C*(i-1)+120*C+96*C*B)]+
round(rep(runif(20,-1.5,1.5),6*C))
dataset$Y[(36*C*(i-1)+1):(36*C*(i-1)+36*C)]<-dataset$Y[(36*C*(i-1)+1)
:(36*C*(i-1)+36*C)]+round(rep(runif(6,-1.5,1.5),6*C))
dataset$Y[(60*C*(i-1)+1+36*C*B):(60*C*(i-1)+60*C+36*C*B)]
<-dataset$Y[(60*C*(i-1)+1+36*C*B):(60*C*(i-1)+60*C+36*C*B)]+
round(rep(runif(10,-1.5,1.5),6*C))
dataset$Y[(120*C*(i-1)+1+96*C*B):(120*C*(i-1)+120*C+96*C*B)]<-
dataset$Y[(120*C*(i-1)+1+96*C*B):(120*C*(i-1)+120*C+96*C*B)]+
round(rep(runif(20,-1.5,1.5),6*C))
dataset$Z[(36*C*(i-1)+1):(36*C*(i-1)+36*C)]<-dataset$Z[(36*C*(i-1)+1)
:(36*C*(i-1)+36*C)]+round(rep(runif(6,-1.5,1.5),6*C))
dataset$Z[(60*C*(i-1)+1+36*C*B):(60*C*(i-1)+60*C+36*C*B)]<-
dataset$Z[(60*C*(i-1)+1+36*C*B):(60*C*(i-1)+60*C+36*C*B)]+
round(rep(runif(10,-1.5,1.5),6*C))
dataset$Z[(120*C*(i-1)+1+96*C*B):(120*C*(i-1)+120*C+96*C*B)]<-
dataset$Z[(120*C*(i-1)+1+96*C*B):(120*C*(i-1)+120*C+96*C*B)]+
round(rep(runif(20,-1.5,1.5),6*C))
}

dataset[,1]<-factor(dataset[,1])
dataset[,4]<-factor(dataset[,4])
dataset[,5]<-factor(dataset[,5])
```

```
pv1<-double(B)
pv2<-double(B)
pv3<-double(B)
pv4<-double(B)
pv5<-double(B)
pv6<-double(B)
pv7<-double(B)
pv8<-double(B)
pv9<-double(B)
pv10<-double(B)
pv11<-double(B)
pv12<-double(B)
pv13<-double(B)
pv14<-double(B)
pv15<-double(B)
pv16<-double(B)
pv17<-double(B)
pv18<-double(B)

for (i in 1:B){
a1<-adonis(dataset[((36*C)*(i-1)+1):((36*C)*(i-1)+(6*C)),8:10]~
ID_blocco[((36*C)*(i-1)+1):((36*C)*(i-1)+(6*C))]+
ID_tratt[((36*C)*(i-1)+1):((36*C)*(i-1)+(6*C))],
method="euclidean",data=dataset,permutations=np)
a2<-adonis(dataset[((36*C)*(i-1)+6*C+1):((36*C)*(i-1)+12*C),8:10]~
ID_blocco[((36*C)*(i-1)+6*C+1):((36*C)*(i-1)+12*C)]+
ID_tratt[((36*C)*(i-1)+6*C+1):((36*C)*(i-1)+12*C)],
method="euclidean",data=dataset,permutations=np)
a3<-adonis(dataset[((36*C)*(i-1)+12*C+1):((36*C)*(i-1)+18*C),8:10]~
ID_blocco[((36*C)*(i-1)+12*C+1):((36*C)*(i-1)+18*C)]+
ID_tratt[((36*C)*(i-1)+12*C+1):((36*C)*(i-1)+18*C)],
method="euclidean",data=dataset,permutations=np)
a4<-adonis(dataset[((36*C)*(i-1)+18*C+1):((36*C)*(i-1)+24*C),8:10]~
ID_blocco[((36*C)*(i-1)+18*C+1):((36*C)*(i-1)+24*C)]+
ID_tratt[((36*C)*(i-1)+18*C+1):((36*C)*(i-1)+24*C)],
method="euclidean",data=dataset,permutations=np)
a5<-adonis(dataset[((36*C)*(i-1)+24*C+1):((36*C)*(i-1)+30*C),8:10]~
ID_blocco[((36*C)*(i-1)+24*C+1):((36*C)*(i-1)+30*C)]+
ID_tratt[((36*C)*(i-1)+24*C+1):((36*C)*(i-1)+30*C)],
method="euclidean",data=dataset,permutations=np)
a6<-adonis(dataset[((36*C)*(i-1)+30*C+1):((36*C)*(i-1)+36*C),8:10]~
ID_blocco[((36*C)*(i-1)+30*C+1):((36*C)*(i-1)+36*C)]+
ID_tratt[((36*C)*(i-1)+30*C+1):((36*C)*(i-1)+36*C)],
method="euclidean",data=dataset,permutations=np)
pv1[i]<-a1$aov.tab[2,6]
pv2[i]<-a2$aov.tab[2,6]
pv3[i]<-a3$aov.tab[2,6]
pv4[i]<-a4$aov.tab[2,6]
pv5[i]<-a5$aov.tab[2,6]
pv6[i]<-a6$aov.tab[2,6]
a7<-adonis(dataset[((60*C)*(i-1)+1+36*B*C):((60*C)*(i-1)+(10*C)+36*B*C),
8:10]~ID_blocco[((60*C)*(i-1)+1+36*B*C):((60*C)*(i-1)+(10*C)+36*B*C)]+
ID_tratt[((60*C)*(i-1)+1+36*B*C):((60*C)*(i-1)+(10*C)+36*B*C)],
method="euclidean",data=dataset,permutations=np)
a8<-adonis(dataset[((60*C)*(i-1)+10*C+1+36*B*C):((60*C)*(i-1)+20*C+36*B*C),
8:10]~ID_blocco[((60*C)*(i-1)+10*C+1+36*B*C):((60*C)*(i-1)+20*C+36*B*C)]+
ID_tratt[((60*C)*(i-1)+10*C+1+36*B*C):((60*C)*(i-1)+20*C+36*B*C)],
method="euclidean",data=dataset,permutations=np)
a9<-adonis(dataset[((60*C)*(i-1)+20*C+1+36*B*C):((60*C)*(i-1)+30*C+36*B*C),
8:10]~ID_blocco[((60*C)*(i-1)+20*C+1+36*B*C):((60*C)*(i-1)+30*C+36*B*C)]+
ID_tratt[((60*C)*(i-1)+20*C+1+36*B*C):((60*C)*(i-1)+30*C+36*B*C)],
method="euclidean",data=dataset,permutations=np)
a10<-adonis(dataset[((60*C)*(i-1)+30*C+1+36*B*C):((60*C)*(i-1)+40*C+36*B*C),
8:10]~ID_blocco[((60*C)*(i-1)+30*C+1+36*B*C):((60*C)*(i-1)+40*C+36*B*C)]+
ID_tratt[((60*C)*(i-1)+30*C+1+36*B*C):((60*C)*(i-1)+40*C+36*B*C)],
method="euclidean",data=dataset,permutations=np)
a11<-adonis(dataset[((60*C)*(i-1)+40*C+1+36*B*C):((60*C)*(i-1)+50*C+36*B*C),
8:10]~ID_blocco[((60*C)*(i-1)+40*C+1+36*B*C):((60*C)*(i-1)+50*C+36*B*C)]+
ID_tratt[((60*C)*(i-1)+40*C+1+36*B*C):((60*C)*(i-1)+50*C+36*B*C)],
method="euclidean",data=dataset,permutations=np)
a12<-adonis(dataset[((60*C)*(i-1)+50*C+1+36*B*C):((60*C)*(i-1)+60*C+36*B*C),
8:10]~ID_blocco[((60*C)*(i-1)+50*C+1+36*B*C):((60*C)*(i-1)+60*C+36*B*C)]+
ID_tratt[((60*C)*(i-1)+50*C+1+36*B*C):((60*C)*(i-1)+60*C+36*B*C)],
method="euclidean",data=dataset,permutations=np)
pv7[i]<-a7$aov.tab[2,6]
pv8[i]<-a8$aov.tab[2,6]
pv9[i]<-a9$aov.tab[2,6]
pv10[i]<-a10$aov.tab[2,6]
pv11[i]<-a11$aov.tab[2,6]
pv12[i]<-a12$aov.tab[2,6]
a13<-adonis(dataset[((120*C)*(i-1)+1+96*B*C):((120*C)*(i-1)+(20*C)+96*B*C),
8:10]~ID_blocco[((120*C)*(i-1)+1+96*B*C):((120*C)*(i-1)+(20*C)+96*B*C)]+
ID_tratt[((120*C)*(i-1)+1+96*B*C):((120*C)*(i-1)+(20*C)+96*B*C)],
method="euclidean",data=dataset,permutations=np)
a14<-adonis(dataset[((120*C)*(i-1)+20*C+1+96*B*C):((120*C)*(i-1)+40*C+96*B*C),
8:10]~ID_blocco[((120*C)*(i-1)+20*C+1+96*B*C):((120*C)*(i-1)+40*C+96*B*C)]+
ID_tratt[((120*C)*(i-1)+20*C+1+96*B*C):((120*C)*(i-1)+40*C+96*B*C)],
method="euclidean",data=dataset,permutations=np)
a15<-adonis(dataset[((120*C)*(i-1)+40*C+1+96*B*C):((120*C)*(i-1)+60*C+96*B*C),
8:10]~ID_blocco[((120*C)*(i-1)+40*C+1+96*B*C):((120*C)*(i-1)+60*C+96*B*C)]+
ID_tratt[((120*C)*(i-1)+40*C+1+96*B*C):((120*C)*(i-1)+60*C+96*B*C)],
method="euclidean",data=dataset,permutations=np)
```

```
a16<-adonis(dataset[((120*C)*(i-1)+60*C+1+96*B*C):((120*C)*(i-1)+80*C+96*B*C),
8:10]~ID_blocco[((120*C)*(i-1)+60*C+1+96*B*C):((120*C)*(i-1)+80*C+96*B*C)]+
ID_tratt[((120*C)*(i-1)+60*C+1+96*B*C):((120*C)*(i-1)+80*C+96*B*C)],
method="euclidean",data=dataset,permutations=np)
a17<-adonis(dataset[((120*C)*(i-1)+80*C+1+96*B*C):((120*C)*(i-1)+100*C+96*B*C),
8:10]~ID_blocco[((120*C)*(i-1)+80*C+1+96*B*C):((120*C)*(i-1)+100*C+96*B*C)]+
ID_tratt[((120*C)*(i-1)+80*C+1+96*B*C):((120*C)*(i-1)+100*C+96*B*C)],
method="euclidean",data=dataset,permutations=np)
a18<-adonis(dataset[((120*C)*(i-1)+100*C+1+96*B*C):((120*C)*(i-1)+120*C+96*B*C),
8:10]~ID_blocco[((120*C)*(i-1)+100*C+1+96*B*C):((120*C)*(i-1)+120*C+96*B*C)]+
ID_tratt[((120*C)*(i-1)+100*C+1+96*B*C):((120*C)*(i-1)+120*C+96*B*C)],
method="euclidean",data=dataset,permutations=np)
pv13[i]<-a13$aov.tab[2,6]
pv14[i]<-a14$aov.tab[2,6]
pv15[i]<-a15$aov.tab[2,6]
pv16[i]<-a16$aov.tab[2,6]
pv17[i]<-a17$aov.tab[2,6]
pv18[i]<-a18$aov.tab[2,6]
mpvnp<-c(mean(pv1<0.05),mean(pv2<0.05),mean(pv3<0.05),mean(pv4<0.05),
mean(pv5<0.05),mean(pv6<0.05),mean(pv7<0.05),mean(pv8<0.05),mean(pv9<0.05),
mean(pv10<0.05),mean(pv11<0.05),mean(pv12<0.05),mean(pv13<0.05),
mean(pv14<0.05),mean(pv15<0.05),mean(pv16<0.05),mean(pv17<0.05),
mean(pv18<0.05))
names(mpvnp)<-c("6normS1","6normS2","6tS1","6tS2","6nasimmS1","6nasimmS2",
"10normS1","10normS2","10tS1","10tS2","10nasimmS1","10nasimmS2","20normS1",
"20normS2","20tS1","20tS2","20nasimmS1","20nasimmS2")
}

pv1a<-double(B)
pv2a<-double(B)
pv3a<-double(B)
```

```
pv4a<-double(B)
pv5a<-double(B)
pv6a<-double(B)
pv7a<-double(B)
pv8a<-double(B)
pv9a<-double(B)
pv10a<-double(B)
pv11a<-double(B)
pv12a<-double(B)
pv13a<-double(B)
pv14a<-double(B)
pv15a<-double(B)
pv16a<-double(B)
pv17a<-double(B)
pv18a<-double(B)

for (i in 1:B){
a1a<-manova(as.matrix(dataset[((36*C)*(i-1)+1):((36*C)*(i-1)+(6*C)),
8:10])~ID_blocco[((36*C)*(i-1)+1):((36*C)*(i-1)+(6*C))]+
ID_tratt[((36*C)*(i-1)+1):((36*C)*(i-1)+(6*C))],data=dataset)
a2a<-manova(as.matrix(dataset[((36*C)*(i-1)+6*C+1):((36*C)*(i-1)+12*C),
8:10])~ID_blocco[((36*C)*(i-1)+6*C+1):((36*C)*(i-1)+12*C)]+
ID_tratt[((36*C)*(i-1)+6*C+1):((36*C)*(i-1)+12*C)],data=dataset)
a3a<-manova(as.matrix(dataset[((36*C)*(i-1)+12*C+1):((36*C)*(i-1)+18*C),
8:10])~ID_blocco[((36*C)*(i-1)+12*C+1):((36*C)*(i-1)+18*C)]+
ID_tratt[((36*C)*(i-1)+12*C+1):((36*C)*(i-1)+18*C)],data=dataset)
a4a<-manova(as.matrix(dataset[((36*C)*(i-1)+18*C+1):((36*C)*(i-1)+24*C),
8:10])~ID_blocco[((36*C)*(i-1)+18*C+1):((36*C)*(i-1)+24*C)]+
ID_tratt[((36*C)*(i-1)+18*C+1):((36*C)*(i-1)+24*C)],data=dataset)
a5a<-manova(as.matrix(dataset[((36*C)*(i-1)+24*C+1):((36*C)*(i-1)+30*C),
8:10])~ID_blocco[((36*C)*(i-1)+24*C+1):((36*C)*(i-1)+30*C)]+
ID_tratt[((36*C)*(i-1)+24*C+1):((36*C)*(i-1)+30*C)],data=dataset)
a6a<-manova(as.matrix(dataset[((36*C)*(i-1)+30*C+1):((36*C)*(i-1)+36*C),
8:10])~ID_blocco[((36*C)*(i-1)+30*C+1):((36*C)*(i-1)+36*C)]+
ID_tratt[((36*C)*(i-1)+30*C+1):((36*C)*(i-1)+36*C)],data=dataset)
pv1a[i]<-as.numeric(unlist(summary(a1a))[53])
pv2a[i]<-as.numeric(unlist(summary(a2a))[53])
pv3a[i]<-as.numeric(unlist(summary(a3a))[53])
pv4a[i]<-as.numeric(unlist(summary(a4a))[53])
pv5a[i]<-as.numeric(unlist(summary(a5a))[53])
pv6a[i]<-as.numeric(unlist(summary(a6a))[53])
a7a<-manova(as.matrix(dataset[((60*C)*(i-1)+1+36*B*C):
((60*C)*(i-1)+(10*C)+36*B*C),8:10])~ID_blocco[((60*C)*(i-1)+1+36*B*C):
((60*C)*(i-1)+(10*C)+36*B*C)]+ID_tratt[((60*C)*(i-1)+1+36*B*C):
((60*C)*(i-1)+(10*C)+36*B*C)],data=dataset)
a8a<-manova(as.matrix(dataset[((60*C)*(i-1)+10*C+1+36*B*C):
((60*C)*(i-1)+20*C+36*B*C),8:10])~ID_blocco[((60*C)*(i-1)+10*C+1+36*B*C):
((60*C)*(i-1)+20*C+36*B*C)]+ID_tratt[((60*C)*(i-1)+10*C+1+36*B*C):
((60*C)*(i-1)+20*C+36*B*C)],data=dataset)
a9a<-manova(as.matrix(dataset[((60*C)*(i-1)+20*C+1+36*B*C):
((60*C)*(i-1)+30*C+36*B*C),8:10])~ID_blocco[((60*C)*(i-1)+20*C+1+36*B*C):
((60*C)*(i-1)+30*C+36*B*C)]+ID_tratt[((60*C)*(i-1)+20*C+1+36*B*C):
((60*C)*(i-1)+30*C+36*B*C)],data=dataset)
a10a<-manova(as.matrix(dataset[((60*C)*(i-1)+30*C+1+36*B*C):
((60*C)*(i-1)+40*C+36*B*C),8:10])~ID_blocco[((60*C)*(i-1)+30*C+1+36*B*C):
((60*C)*(i-1)+40*C+36*B*C)]+ID_tratt[((60*C)*(i-1)+30*C+1+36*B*C):
((60*C)*(i-1)+40*C+36*B*C)],data=dataset)
a11a<-manova(as.matrix(dataset[((60*C)*(i-1)+40*C+1+36*B*C):
((60*C)*(i-1)+50*C+36*B*C),8:10])~ID_blocco[((60*C)*(i-1)+40*C+1+36*B*C):
((60*C)*(i-1)+50*C+36*B*C)]+ID_tratt[((60*C)*(i-1)+40*C+1+36*B*C):
((60*C)*(i-1)+50*C+36*B*C)],data=dataset)
a12a<-manova(as.matrix(dataset[((60*C)*(i-1)+50*C+1+36*B*C):
```

```
((60*C)*(i-1)+60*C+36*B*C),8:10])~ID_blocco[((60*C)*(i-1)+50*C+1+36*B*C):
((60*C)*(i-1)+60*C+36*B*C)]+ID_tratt[((60*C)*(i-1)+50*C+1+36*B*C):
((60*C)*(i-1)+60*C+36*B*C)],data=dataset)
pv7a[i]<-as.numeric(unlist(summary(a7a))[53])
pv8a[i]<-as.numeric(unlist(summary(a8a))[53])
pv9a[i]<-as.numeric(unlist(summary(a9a))[53])
pv10a[i]<-as.numeric(unlist(summary(a10a))[53])
pv11a[i]<-as.numeric(unlist(summary(a11a))[53])
pv12a[i]<-as.numeric(unlist(summary(a12a))[53])
a13a<-manova(as.matrix(dataset[((120*C)*(i-1)+1+96*B*C):
((120*C)*(i-1)+(20*C)+96*B*C),8:10])~ID_blocco[((120*C)*(i-1)+1+96*B*C):
((120*C)*(i-1)+(20*C)+96*B*C)]+ID_tratt[((120*C)*(i-1)+1+96*B*C):
((120*C)*(i-1)+(20*C)+96*B*C)],data=dataset)
a14a<-manova(as.matrix(dataset[((120*C)*(i-1)+20*C+1+96*B*C):
((120*C)*(i-1)+40*C+96*B*C),8:10])~ID_blocco[((120*C)*(i-1)+20*C+1+96*B*C):
((120*C)*(i-1)+40*C+96*B*C)]+ID_tratt[((120*C)*(i-1)+20*C+1+96*B*C):
((120*C)*(i-1)+40*C+96*B*C)],data=dataset)
a15a<-manova(as.matrix(dataset[((120*C)*(i-1)+40*C+1+96*B*C):
((120*C)*(i-1)+60*C+96*B*C),8:10])~ID_blocco[((120*C)*(i-1)+40*C+1+96*B*C):
((120*C)*(i-1)+60*C+96*B*C)]+ID_tratt[((120*C)*(i-1)+40*C+1+96*B*C):
((120*C)*(i-1)+60*C+96*B*C)],data=dataset)
a16a<-manova(as.matrix(dataset[((120*C)*(i-1)+60*C+1+96*B*C):
((120*C)*(i-1)+80*C+96*B*C),8:10])~ID_blocco[((120*C)*(i-1)+60*C+1+96*B*C):
((120*C)*(i-1)+80*C+96*B*C)]+ID_tratt[((120*C)*(i-1)+60*C+1+96*B*C):
((120*C)*(i-1)+80*C+96*B*C)],data=dataset)
a17a<-manova(as.matrix(dataset[((120*C)*(i-1)+80*C+1+96*B*C):
((120*C)*(i-1)+100*C+96*B*C),8:10])~ID_blocco[((120*C)*(i-1)+80*C+1+96*B*C):
((120*C)*(i-1)+100*C+96*B*C)]+ID_tratt[((120*C)*(i-1)+80*C+1+96*B*C):
((120*C)*(i-1)+100*C+96*B*C)],data=dataset)
a18a<-manova(as.matrix(dataset[((120*C)*(i-1)+100*C+1+96*B*C):
((120*C)*(i-1)+120*C+96*B*C),8:10])~ID_blocco[((120*C)*(i-1)+100*C+1+96*B*C):
```

```
((120*C)*(i-1)+120*C+96*B*C)]+ID_tratt[((120*C)*(i-1)+100*C+1+96*B*C):
((120*C)*(i-1)+120*C+96*B*C)],data=dataset)
pv13a[i]<-as.numeric(unlist(summary(a13a))[53])
pv14a[i]<-as.numeric(unlist(summary(a14a))[53])
pv15a[i]<-as.numeric(unlist(summary(a15a))[53])
pv16a[i]<-as.numeric(unlist(summary(a16a))[53])
pv17a[i]<-as.numeric(unlist(summary(a17a))[53])
pv18a[i]<-as.numeric(unlist(summary(a18a))[53])
mpvp<-c(mean(pv1a<0.05),mean(pv2a<0.05),mean(pv3a<0.05),mean(pv4a<0.05),
mean(pv5a<0.05),mean(pv6a<0.05),mean(pv7a<0.05),mean(pv8a<0.05),mean(pv9a<0.05),
mean(pv10a<0.05),mean(pv11a<0.05),mean(pv12a<0.05),mean(pv13a<0.05),
mean(pv14a<0.05),mean(pv15a<0.05),mean(pv16a<0.05),mean(pv17a<0.05),
mean(pv18a<0.05))
names(mpvp)<-c("6normS1","6normS2","6tS1","6tS2","6nasimmS1","6nasimmS2"
"10normS1","10normS2","10tS1","10tS2","10nasimmS1","10nasimmS2","20normS1",
"20normS2","20tS1","20tS2","20nasimmS1","20nasimmS2")
}

t<-list(mpvnp,mpvp)
names(t)<-c("NPManova","PManova")
t
}

a<-simula.adonis(mu=mu21,B=500,np=200)
write.table(a$NPManova,"a1.txt",sep="\t",row.names=F,col.names=F)
write.table(a$PManova,"a2.txt",sep="\t",row.names=F,col.names=F)
b<-simula.adonis(mu=mu22,B=500,np=200)
write.table(b$NPManova,"b1.txt",sep="\t",row.names=F,col.names=F)
write.table(b$PManova,"b2.txt",sep="\t",row.names=F,col.names=F)
c<-simula.adonis(mu=mu23,B=500,np=200)
write.table(c$NPManova,"c1.txt",sep="\t",row.names=F,col.names=F)
write.table(c$PManova,"c2.txt",sep="\t",row.names=F,col.names=F)
d<-simula.adonis(mu=mu24,B=500,np=200)
write.table(d$NPManova,"d1.txt",sep="\t",row.names=F,col.names=F)
write.table(d$PManova,"d2.txt",sep="\t",row.names=F,col.names=F)
e<-simula.adonis(mu=mu31,B=500,np=200)
write.table(e$NPManova,"e1.txt",sep="\t",row.names=F,col.names=F)
write.table(e$PManova,"e2.txt",sep="\t",row.names=F,col.names=F)
f<-simula.adonis(mu=mu32,B=500,np=200)
write.table(f$NPManova,"f1.txt",sep="\t",row.names=F,col.names=F)
write.table(f$PManova,"f2.txt",sep="\t",row.names=F,col.names=F)
g<-simula.adonis(mu=mu33,B=500,np=200)
write.table(g$NPManova,"g1.txt",sep="\t",row.names=F,col.names=F)
write.table(g$PManova,"g2.txt",sep="\t",row.names=F,col.names=F)
h<-simula.adonis(mu=mu34,B=500,np=200)
write.table(h$NPManova,"h1.txt",sep="\t",row.names=F,col.names=F)
write.table(h$PManova,"h2.txt",sep="\t",row.names=F,col.names=F)
i<-simula.adonis(mu=mu41,B=500,np=200)
write.table(i$NPManova,"i1.txt",sep="\t",row.names=F,col.names=F)
write.table(i$PManova,"i2.txt",sep="\t",row.names=F,col.names=F)
j<-simula.adonis(mu=mu42,B=500,np=200)
write.table(j$NPManova,"j1.txt",sep="\t",row.names=F,col.n
ames=F)
write.table(j$PManova,"j2.txt",sep="\t",row.names=F,col.names=F)
k<-simula.adonis(mu=mu43,B=500,np=200)
write.table(k$NPManova,"k1.txt",sep="\t",row.names=F,col.names=F)
write.table(k$PManova,"k2.txt",sep="\t",row.names=F,col.names=F)
l<-simula.adonis(mu=mu44,B=500,np=200)
write.table(l$NPManova,"l1.txt",sep="\t",row.names=F,col.names=F)
write.table(l$PManova,"l2.txt",sep="\t",row.names=F,col.names=F)
m<-simula.adonis(mu=mu51,B=500,np=200)
write.table(m$NPManova,"m1.txt",sep="\t",row.names=F,col.names=F)
write.table(m$PManova,"m2.txt",sep="\t",row.names=F,col.names=F)
n<-simula.adonis(mu=mu52,B=500,np=200)
write.table(n$NPManova,"n1.txt",sep="\t",row.names=F,col.names=F)
write.table(n$PManova,"n2.txt",sep="\t",row.names=F,col.names=F)
o<-simula.adonis(mu=mu53,B=500,np=200)
write.table(o$NPManova,"o1.txt",sep="\t",row.names=F,col.names=F)
write.table(o$PManova,"o2.txt",sep="\t",row.names=F,col.names=F)
p<-simula.adonis(mu=mu54,B=500,np=200)
write.table(p$NPManova,"p1.txt",sep="\t",row.names=F,col.names=F)
write.table(p$PManova,"p2.txt",sep="\t",row.names=F,col.names=F)
```

Bibliography

[1] F. Aiello, M. Attanasio (2004). *How to transform a batch of simple indicators to make up a unique one?* Atti del convegno SIS, Bari, June 2004 - Sessioni Specializzate.

[2] F. Aiello, M. Attanasio (2006). *Some Issues in Constructing Composite Indicators*. VIII International Meeting on Quantitative Methods for Applied Sciences, Certosa di Pontignano, 11-13 September 2006.

[3] AISE (2009). URL http://www.aise.eu/.

[4] R. Arboretti Giancristofaro, D. Basso, S. Bonnini, L. Corain (2008). *A robust approach for treatment ranking within the multivariate one-way ANOVA layout*.

[5] R. Arboretti Giancristofaro, S. Bonnini, L. Corain, L. Salmaso (2008). *Statistical Approaches for Improving the Product/Service Quality with applications to the University System evaluation*. Cleup, Padova, 1-143.

[6] A. Azzalini (1985). *A class of distributions, which inludes the normal ones*. Scandinavian Journal of Statistics 12,171-178.

[7] W. O. Bearden, R. G. Netemeyer (1999). *Handbook of marketing scales: multi-item measures for marketing consumer behavior research.* Thousand Oaks, California: Sage Publications.

[8] L. Bernardi (2006), *Dispense per il corso di Metodologia della Ricerca 2006/07.* Dipartimento di Scienze Statistiche, Università degli Studi di Padova.

[9] S. Bonnini, L. Corain, A. Cordellina, A. Crestana, R. Musci, L. Salmaso (2006). *A novel global performance score with application to the evaluation of new detergents.*

[10] K. K. Boyer, R. Verma (2000). *Multiple raters in survey-based operations management research: a review and tutorial.* Production and Operations Management, Vol. 9, 2.

[11] A. S. Choo, K. W. Linderman, R. G. Schroeder (2007). *Method and psychological effects on learning behaviors and knowledge creation in quality improvement projects.* Management Science, Vol. 53, 3, 437-450.

[12] D. L. Clason, T. J. Dormody (1994). *Analyzing data measured by individual Likert-type items.* Journal of Agricultural Education, Vol. 35, 4, 31-35.

[13] L. Corain, L. Salmaso (2007). *A nonparametric method for defining a global preference ranking of industrial products.* Journal of Applied Statistics, 34, 2, 203-216.

[14] L. Corain, L. Salmaso (2008). *Dispense per il corso di Metodi e Applicazioni Statistiche.* Dipartimento di Tecnica e Gestione dei Sistemi Industriali, Università degli Studi di Padova.

[15] P. Corbetta (2003). *La ricerca sociale: metodologia e tecniche II. Le tecniche quantitative.* Il Mulino, Bologna.

[16] F. Delvecchio (1995). *Scale di misura e indicatori sociali.* Caucci, Bari.

[17] L. Fabbris (2007). *Dispense per il corso di Statistica Sociale 2007/08.* Dipartimento di Scienze Statistiche, Università degli Studi di Padova.

[18] P. Fayers, D. Hand (2002). *Casual variables, indicator variables and measurement scales: an example from quality of life.* Journal of the Royal Statistical Society Series A, 165, 233-261.

[19] B. Flynn et al. (2009). *The impact of supply chain complexity on manufacturing plant performance.* Journal of Operations Management 27, 78-93.

[20] W. H. Glick (1985). *Conceptualizing and measuring organizational and psychological climate: pitfalls in multilevel research.* Academy of Management Review, Vol. 10, 3, 601-616.

[21] J. A. Gliem, R. R. Gliem (2003). *Calculating, interpreting, and reporting Cronbach's alpha reliability coefficient for Likert-type scales.* Midwest research to practice conference in adult, continuing, and community

education.

[22] E. Gori, G. Vittadini (1999). *La valutazione dell'efficienza ed efficacia dei servizi alla persona: impostazione e metodi.* In: E. Gori, G. Vittadini (eds.), Qualità e valutazione nei servizi di pubblica utilità, Etas, RCS Libri, Milano, 121-241.

[23] R. A. Henry (1995). *Improving group judgement accuracy: information sharing and determining the best member.* Organizational behavior nad human decision processes, Vol. 62, 2, 190-197.

[24] D. V. Hinkley (1975). *On power transformations to symmetry.* Biometrika, 101-111.

[25] S. Jamieson (2004). *Likert scales: how to (ab)use them.* Medical Education, 38, 1212-1218.

[26] M. Ketovivi, R. Schroeder (2004). *Manufacturing practices, strategic fit and performance: a routine-based view.* International Journal of Operations & Production Management, Vol. 24, 2, 171-191.

[27] M. Ketovivi, R. Schroeder (2004). *Perceptual measures of performance: fact or fiction?.* Journal of Operations Management 22, 247-264.

[28] T. Knapp (1990). *Treating ordinal scales as ordinal scales: an attempt to resolve the controversy.* Nursing Research, Vol. 39, 2.

[29] A. Lago, F. Pesarin (2000). *Non parametric combination of dependent rankings with application to the quality assessment of industrial products.* Metron, 58, 39-52.

[30] M. Lalla (1998). *La scala di Likert per la valutazione della didattica.*

[31] R. Likert (1932). *A Technique for the Measurement of Attitudes.* Archives of Psychology 140, 1-55.

[32] G. Liu, R. Shah, R. G. Schroeder (2006). *Linking work design to mass customization: a sociotechnical systems perspective.* Decision Sciences, Vol. 37, 4.

[33] M. Lu (1999). *Determinants of residential satisfaction: ordered logit vs. regression models.* Growth and Change, Vol. 30, 264-287.

[34] D. N. Mallick, R. G. Schroeder (2005). *An integrated framework for measuring product development performance in high technology industries.* Production and Operations Management Society, Vol. 14, 2, 142-158.

[35] A. Marradi (1988). *Costruire il dato.* Angeli, Milano.

[36] D. C. Montgomery (2005). *Design and analysis of experiments, 6th edition.* John Wiley and sons, Chichester.

[37] M. Nardo, M. Saisana, A. Saltelli, S. Tarantola, A. Hoffman, E. Giovannini (2008). *Handbook on constructing composite indicators:*

methodology and user guide. OECD Publishing.

[38] M. A. M. Primo, S. D. Amundson (2002). *An exploratory study of the effects of supplier relationships on new product development outcomes.* Journal of Operations Management 20, 33-52.

[39] R Development Core Team (2009). *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

[40] J. P. Robinson, P. R. Shaver, L. S. Wrightsman (1991). *Criteria for scale selection and evaluation.* Measures of Personality and Social Psychological Attitudes.

[41] F. M. Shrive, H. Stuart, H. Quan, W. A. Ghali (2006). *Dealing with missing data in a multi-question depression scale: a comparison of imputation methods.* BMC Medical Research Methodology, 6, 57.

[42] G. I. Susman, J. M. Ray (1999). *Test of a model of organizational contributors to product development team effectiveness.* Journal of Engineering and Technology Management 16, 223-245.

[43] E. Svensson (2001). *Construction of a single global scale for multi-item assessments of the same variable.* Statist. Med., 20:3831-3846 (DOI: 10.1002/sim.1148).

[44] E. Svensson (2001). *Guidelines to statistical evaluation of data from rating scales and questionnaires.* J Rehab Med, 33, 47-48.

[45] G. H. Van Bruggen et al. (2002). *Informants in organizational marketing research: why use multiple informants and how to aggregate responses.* Journal of Marketing Research, Vol. 39, 469-478.

[46] L. Ventura (2007). *Dispense per il corso di Modelli Statistici II 2006/07.* Dipartimento di Scienze Statistiche, Università degli Studi di Padova.

[47] R. Verma, M. E. Pullman (1998). *An analysis of the supplier selection process.* Omega, International Journal of Management Science, Vol. 26, 6, 739-750.

[48] Wikipedia website (english version), entry: *Likert scale.*
URL http://en.wikipedia.org/wiki/Likert_scale.

[49] Wikipedia website (italian version), entry: *Scala Likert.*
URL http://it.wikipedia.org/wiki/Scala_Likert.

[50] B. D. Zumbo, C. O. Ochieng (2002). *The effects of various configurations of Likert, ordered categorical, or rating scale data on the ordinal logistic regression pseudo R-squared measure of fit: the case of the cumulative logit model.* Paper presented in the symposium “Monte Carlo Studies of Statistical Procedures” at the American Educational Research Association conference April 2002, New Orleans.

A lot of other papers and books have been seen and read in order to perform this work, but hear we have quoted only the main ones.

Acknowledgements

Sincere thanks are for all people who have helped me with this thesis and who were always helpful: my relator prof. Fortunato Pesarin, my correlator prof. Luigi Salmaso, prof. Livio Corain, prof. Roberto Filippini, prof. Anna Nosella and postgraduate student Silvia Cantarello.

Sincere thanks are also naturally given to all people who have been always close to me in these months (and not only in this months, obviously), even in difficult times: my parents Franz and Maria and my brother Alessandro above all, my grandparents Giuseppe and Angela, my uncle Antonio and his family, my aunt Silvana and my cousin Samuel, my second cousin Eros, which has started to study statistics too, and all my other closer relatives whose name I do not report here because they are really a lot; my friends of the university or friends of them that are also my friends, in particular Riccardo, Pietro, Alessandro, Federico, Emanuele, Andrea, Matteo, Tommaso, Alessio, William, Katia, Eleonora, Silvia, Alessandra and Elena; my friends of the high school, in particular Mattia, Luisa, Claudio, Chiara, Giulio, Martina and Luca; my friends of the medium school Mattia, Davide and Daniele.

In general particular thanks are given to all people who love me and are close to me, even if I have not reported their name here.

Finally a last great thank is really due to Emanuele, Andrea and his sister Sabrina for the time spent to correct my english in this thesis before I have delivered it.