R SDisc: Integrated methodology for the identification of homogeneous profiles in data distribution

F Colas

December 9, 2009

The time and the expertise to perform robust subtyping inferences in data are often regarded as limiting factors for the range of analysis hypothesis considered. Indeed, not only competence in cluster analysis is required but also in exploratory data analysis, regression, statistical testing, computational statistics, classifier training and testing, data visualization and scientific programming. Identifying data subtypes is therefore greatly interdisciplinary. Hence, SDisc addresses an essential demand, originally emanating from clinical research, for an integrated scenario performing the different steps of a subtyping analysis.

With SDisc, analyzes also become more straightforward and therefore more accessible to many investigators. The well-defined data structures of the package greatly enhances the analysis reproducibility, whereas with the public release of the package, research teams from elsewhere can benefit of a tested scenario to perform their own analyzes. Additionally, more data analysis hypotheses than before are considered. For instance, adjusting the data preparation at an advanced stage is now possible and only requires new input settings for the scenario. The next calculation will update the graphics, the measurements and the statistics which, in turn, may enable to compare different data treatments at a *meta*-level.

The possible domains of application are in clinical research on complex pathologies like Osteoarthritis, Parkinson's disease and aggressive brain tumor diagnosis. For these pathologies, more homogeneous patient subtypes is expected to help to break down the existing clinical heterogeneity and thus further enhance the understanding of their underlying mechanisms. Hence, the discovered subtypes may help to advance the development of new treatment strategies.

Moreover, SDisc confronts particularly with clinical research requirements in terms of data analysis. It considers the validity aspect of the inference steps carried out in the course of a subtyping analysis, the accessibility facet to enable non-expert computer scientist to perform and/or reproduce analyzes independently and straightforwardly, as well as the availability aspect by the distribution of the generic solution as a documented open source R package.

In the following, **outline**...

Contents

1 Application areas

In the following, we report the rationale of subtype discovery data analyzes by reviewing a number of domains facing this problem, in medical research (Mol LUMC, Neu LUMC, Psy LUMC, SOCO), in chemoinformatics (Pharma-IT) and in recycling (CIFASIS). For each application domain we motivate the research target.

Osteoarthritis (OA) Searching for subtypes in the distribution of OA may allow to study the spread of the disease across different sites and to show whether it is stochastic or follows a particular pattern. Such subtypes could contribute to elucidate the clinical heterogeneity of OA [**?**] and therefore enhance the identification of the disease pathways (genetics, pathophysiological mechanisms).

Parkinson's disease (PD) Among PD patients, there is marked heterogeneity in the clinical phenotype which differs in the presence, the severity, and the progression rate of the various features while differences are also observed in other clinical variables like age at onset [**?**]. This clinical heterogeneity may indicate the existence of subtypes, whose identification may advance our understanding of the underlying pathological mechanisms of PD and thus, advance the development of more focused treatment strategies [**?**].

Major depressive disorders (MDD) and anxiety disorders (ANX) According to the tripartite model, depression and anxiety symptoms are classified into three dimensions reflecting: a common factor of negative affect, and disorder/specific dimensions lack of positive affect (MDD) and somatic arousal (ANX) [**?**]. As there is substantial heterogeneity in these diagnostic categories, identifying more homogeneous subtypes of MDD/ANX based on symptom profiles could help to find prognostic factors, risk factors, and treatment strategies.

Glioblastoma and metastasis We attempt to find discriminative subtypes of aggressive brain tumors using long echo term spectroscopy data. In particular, we search for frequencies of the spectrum making the signals of these pathologies similar and, as a result, difficult to discriminate. Further, as the underlying heterogeneity of the glioblastoma pathology remains uncharacterized at large, subtypes of this brain tumor may enhance our understanding of the different forms of glioblastoma. Last, as effective patient care orientation depends on accurate medical diagnosis, new subtypes of these pathologies may provide a basis to improve their correct discrimination. Our results are reported in [**?**].

Additional analyzes The purpose of the Pharma-IT analysis is to identify subtypes in databases of molecules. As molecules are classified into a number of complex bioactivity classes, an automatic subtyping of the molecules, grouping them based on their similarity, may help to further understand those classes.

Second, with the CIFASIS, an automatic classifier is searched for capable to discriminate between different classes of plastics. In this analysis, the search for subtypes in the distribution of spectroscopy measurements is susceptible to report the most discriminative spectra frequencies, first, and second, to identify whether spectra subtypes exhibit a structure in correlation with the different classes of plastics.

2 Use case of an SDisc analysis

The scenario illustrated by Figure 1, starts with a data preparation step where close collaboration with the domain experts is required to obtain a description of the data. These are written into a settings file that defines how to transform each variable, which variable to include in the cluster modeling, how to summarize variables graphically and statistically. To facilitate the task of writing that file, the package implements a function that generates default settings.

Next, a preliminary subtype discovery analysis is performed to test the flow of statistical inferences, and to commence the discussion with the research team. A graphic report of the data container is produced, which enables exploratory data analysis (EDA). It creates box plots, histograms, and several other variable-specific statistics. To characterize the mixture models, the scenario assembles a number of statistics and of graphics. This output enables to complete with the research team a first instructional walk over the whole inference process.

Subsequently, the subtype discovery can be adjusted given considerations over the number of samples, the number of dimensions, the calculation time, the evaluation of the significance of the subtypes by some statistical test (e.g. a χ^2 test of association or of goodness of fit, a risk ratio) or the posterior characterization of the subtypes. This adjustment may involve additional validation data, alternative data processing, filtering of outliers, re-organization of the graphics. Thus, it may require the preparation of a new settings file and a new data container. The moment these considerations are fixed, a new analysis is performed.

In the succeeding, we present a résumé of the subtyping inference carried out on a cohort study of patients with PD. The results of this analysis are described in [**?**].

> The clinical presentation of PD was described by 13 variables from which the variability explained by the disease duration was removed. Standard scores were taken and a model based cluster analysis was repeated from 50 different starting points, for 3, 4 and 5 clusters and for 5 differently parameterized Gaussian models. It resulted in 750 estimated models. Cluster average PD

patterns were visualized using parallel coordinates and heat maps. The distributions of patients in the different cluster solutions were cross-compared in terms of association tables and of a χ^2-based coefficient of nominal association (Cramer's V). Finally, the consistency of the subtypes was evaluated for the reproducibility between the assessments of year one and two.

3 Extending SDisc

When applying the scenario to a growing number of application areas, we develop new methods and extend others to carry out subtype discovery analyzes on new data types and to report field-specific subtype validation methods. Consequently, in what follows, we describe our development methodology to extend the scenario's functionalities.

First, we implement a prototype of the new functionality using the real data of the new application. We update the prototype functionalities gradually, from a field-specific procedure to a more general one. Then, we re-design the procedure as a function, which enables its re-use in other contexts. Ultimately, we implement that procedure and the data structure in an object-oriented mode of programming which in turn, will improve its reliability and guarantee its generality. Later, as the new function stabilizes, or when another application area utilizes it, we include it into the development source code of the package. Periodically, we submit the development source code to the `subversion` system. Before each release of a new version that freezes the functionalities, we update the documentation.

4 SDisc as an R package

The R platform for statistical computing [**?**] as well as the BioConductor project for the comprehension and the analysis of genomic data [**?**] are two projects that gained widespread exposure in the last years. This exposure is partly the result of the abundance of data sources in need of analysis and of a growing demand for analysis reproducibility.

For both projects, Figure 2 portrays the growing number of *new* submissions over the years. It shows the wide acceptance, and thus the relevance, of the R platform for statistical computing as a means to publish scientific programs. In parallel, the BioConductor initiative successfully attracted the creation of softwares in bioinformatics. Yet, for both projects the number of new submissions is reducing. A first hypothesis is that the field of bioinformatics and statistical computing is reaching maturity. A second one is that the total software production is reaching some limit. Or, else, new packages are no longer systematically added to those two repositories, of which SDisc would represent an illustrative example as it was initially submitted to the NBIC gforge.

Thus, SDisc fits in the trend to make available and open source the software used to perform a data analysis. Further, as it was applied to very different application areas, the subtyping problem appears recurrent and thus, very general. Last, the variety of data types analyzed also demonstrates the scenario's flexibility.

5 Hands on R SDisc

5.1 Installing

```
> install.packages("SDisc", dep = TRUE)
```

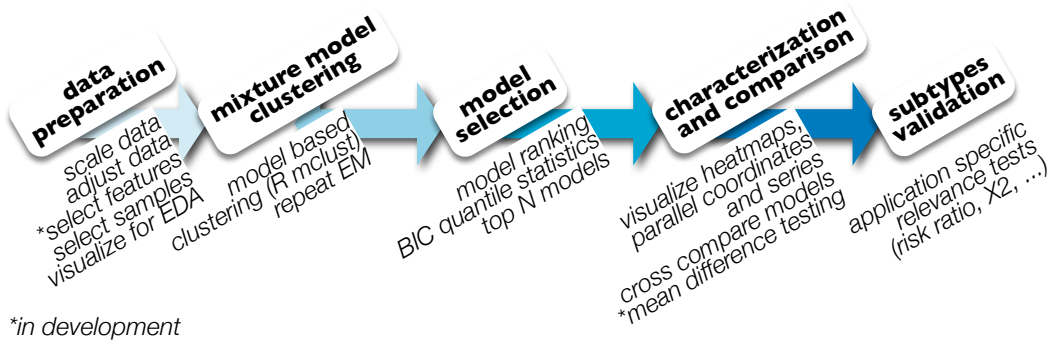


Figure 1: The data mining scenario consists in a sequence of five steps [**?**, Colas et al., 2008a]: the data preparation, the cluster modeling based on [**?**, **?**], the model selection, the characterization and comparison of the subtypes and the relevance evaluation. On top of each step, we illustrate some of the tables and graphics it can produces. For more details, see the vignette documentation [Colas, 2009b].

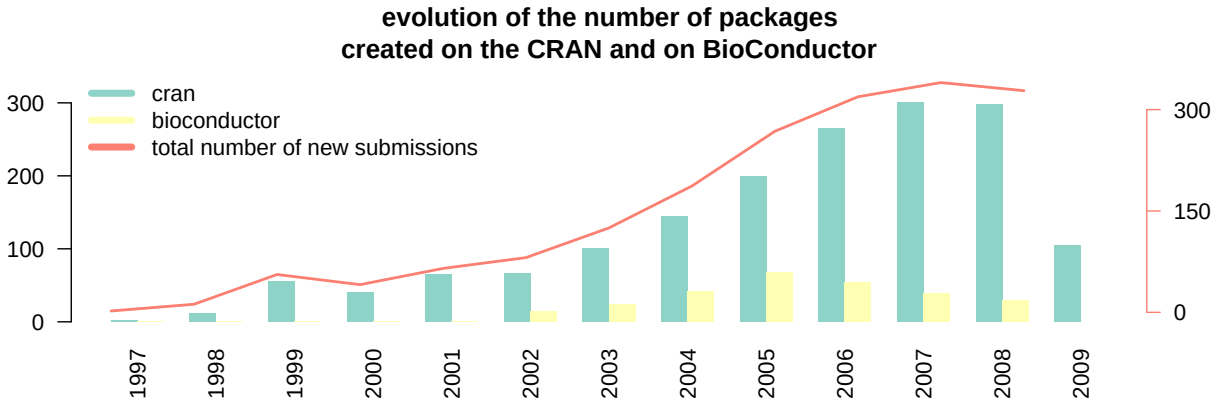


Figure 2: The number of *new* submissions attained 300 packages per year in 2007 and 2008 for the CRAN, and 68 for BioConductor. Yet, in 2008 and 2009, the number of new submissions is slowing down for both projects.

```
> library(SDisc)

by using mclust, you accept the license agreement in the LICENSE file
and at http://www.stat.washington.edu/mclust/license.txt
Package SparseM (0.79) loaded.  To cite, see citation("SparseM")

R CMD INSTALL SDisc_1.18.tar.gz
```

5.2 SDData: the data container of SDisc

rationale: contains all data and transformation within a single container readable by r program. enables repeatability and reproducibility of data transformation. accessibility of the variable estimates. also stores important computing environment informations. object-oriented enabled and as such, capacity to make EDA plots, data summary, data prints...

Example from the documentation: iris data, mean and sd

```
> settings <- SDDataSettings(iris)
> settings["Species", ] <- c(NA, FALSE, NA, NA, NA, NA)
> x <- SDData(iris, settings = settings, prefix = "SDDataBasic")

> print(x, rseed = 6013, latex = TRUE)
```

	Petal.Length	Petal.Width	Sepal.Length
78	5.00	1.70	6.70
109	5.80	1.80	6.70
65	3.60	1.30	5.60

Table 1: SDDataBasic, extract of the **original** data matrix.

	Petal.Length	Petal.Width	Sepal.Length
78	0.70	0.66	1.03
109	1.16	0.79	1.03
65	-0.09	0.13	-0.29

Table 2: SDDataBasic, extract of the **transformed** data matrix.

```
> plot(x, latex = TRUE)
```

```
> summary(x, latex = TRUE)
```

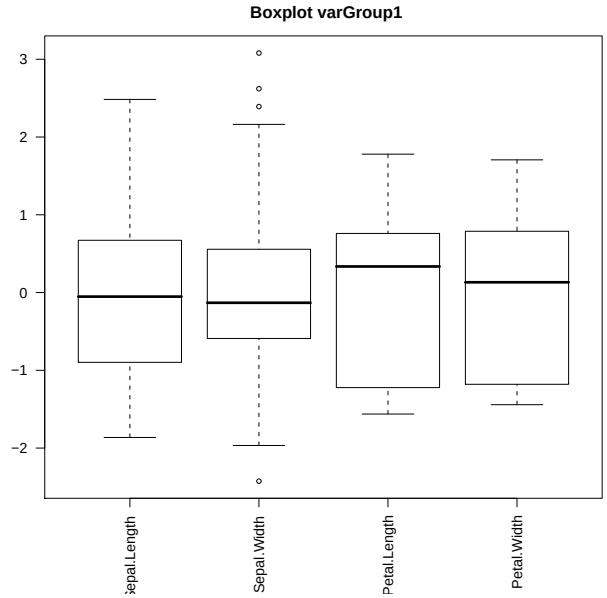


Figure 3: SDDataBasic, **boxplots** of the variables of the factor **varGroup1**.

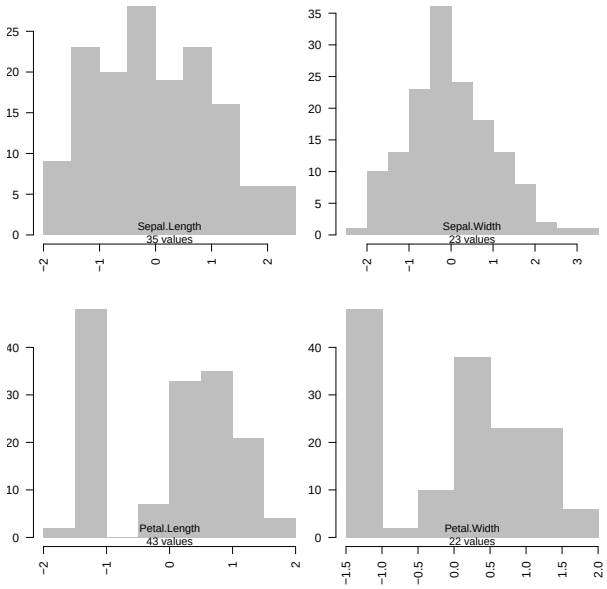


Figure 4: SDDataBasic, **histograms** of the variables of the factor **varGroup1**.

	mean	sd
Sepal.Length	5.84e+00	8.28e-01
Sepal.Width	3.06e+00	4.36e-01
Petal.Length	3.76e+00	1.77e+00
Petal.Width	1.20e+00	7.62e-01

Table 3: SDDataBasic summary of the different data treatments operated on the data.

Example: edit the settings of the file

```
> SDDataSettings(iris, latex = TRUE)
```

	oddGroup	inCAnalysis	tFun	vParGroup	vParY	vHeatmapY
Sepal.Length	Sepal.Length	TRUE	mean sd	varGroup1	1	1
Sepal.Width	Sepal.Width	TRUE	mean sd	varGroup1	2	2
Petal.Length	Petal.Length	TRUE	mean sd	varGroup1	3	3
Petal.Width	Petal.Width	TRUE	mean sd	varGroup1	4	4
Species	Species	TRUE	mean sd	varGroup1	5	5

Table 4: SDDataSettings

```
> SDDataSettings(iris, asCSV = TRUE)
> mySettingsMatrix <- SDDataSettings(iris)
```

Example: two numbers to test the mean and median thing, two gaussian variables to test the mean and sd of scale

```
> set.seed(6014)
> mat <- matrix(c(rnorm(50), rnorm(50, mean = 3, sd = 5), rnorm(50, mean = -2,
      sd = 4)), 50, 3)
> settings <- SDDataSettings(mat)
> x <- SDData(mat, settings = settings, prefix = "SDDataCenter")

> print(x, rseed = 6013, latex = TRUE)
```

	v2	v3	v1
26	3.76	-2.74	-0.01
36	1.36	-9.46	-0.93
22	-3.77	-3.76	2.61

Table 5: SDDataCenter, extract of the **original** data matrix.

	v2	v3	v1
26	0.18	-0.14	0.10
36	-0.38	-1.80	-0.74
22	-1.58	-0.40	2.50

Table 6: SDDataCenter, extract of the **transformed** data matrix.

```
> plot(x, latex = TRUE)
```

```
> summary(x, latex = TRUE)
```

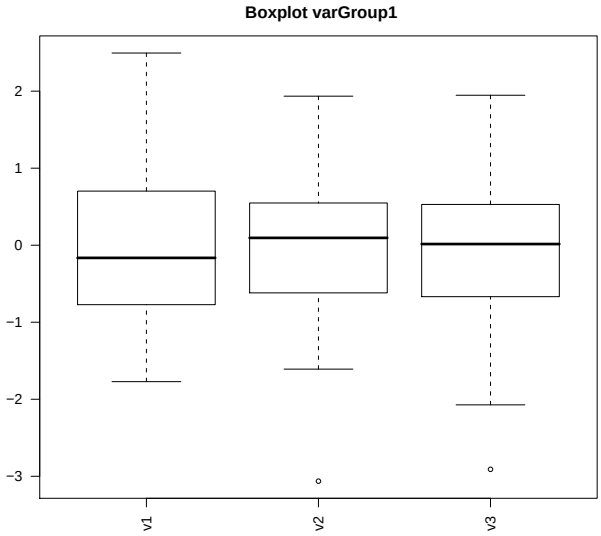


Figure 5: SDDataCenter, **boxplots** of the variables of the factor **varGroup1**.

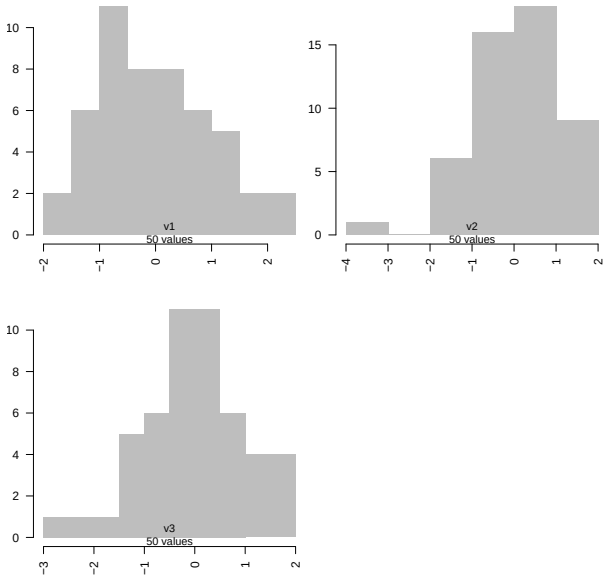


Figure 6: SDDataCenter, **histograms** of the variables of the factor **varGroup1**.

	mean	sd
v1	-1.20e-01	1.09e+00
v2	3.01e+00	4.28e+00
v3	-2.15e+00	4.05e+00

Table 7: SDDataCenter summary of the different data treatments operated on the data.

Example: a random noise uniform variable and a dependent variable to show time association

```
> set.seed(6015)
> epsilon <- runif(50)
> time <- sample(1:5, 50, replace = TRUE)
> vDependent <- 2 * time + epsilon
> mat <- matrix(c(rnorm(50), time, epsilon, vDependent, vDependent), 50, 5)
> colnames(mat) <- c("vNormal", "time", "epsilon", "vDependentOrig", "vDependent")
> settings <- SDDataSettings(mat)
> settings[, "tFun"] <- c("mean sd", "", "", "", "lm(vDependent~time)")
> xLM <- SDData(mat, settings = settings, prefix = "SDDataLinearModel")

> print(xLM, rseed = 6013, latex = TRUE)
```

	epsilon	vDependent	time
26	0.40	10.40	5.00
36	0.24	8.24	4.00
22	0.54	10.54	5.00

Table 8: SDDataLinearModel, extract of the **original** data matrix.

	epsilon	vDependent	time
26	0.40	-0.00	5.00
36	0.24	0.18	4.00
22	0.54	-0.15	5.00

Table 9: SDDataLinearModel, extract of the **transformed** data matrix.

```
> plot(xLM, latex = TRUE)

> summary(xLM, q = "lm", latex = TRUE, sanitize = FALSE)
```

	(Intercept) (SE; Pr(>|t|))	time (SE; Pr(>|t|))	R^2 (adj-R^2; N)
vDependent time	0.53 (0.09; 9.1e-07)	1.97 (0.03; 8.4e-51)	0.99 (0.99; 50)

Table 10: SDDataLinearModel summary of the different data treatments operated on the data.

```
> summary(xLM, q = "mean|sd", latex = TRUE)

> summary(xLM, latex = TRUE)
```

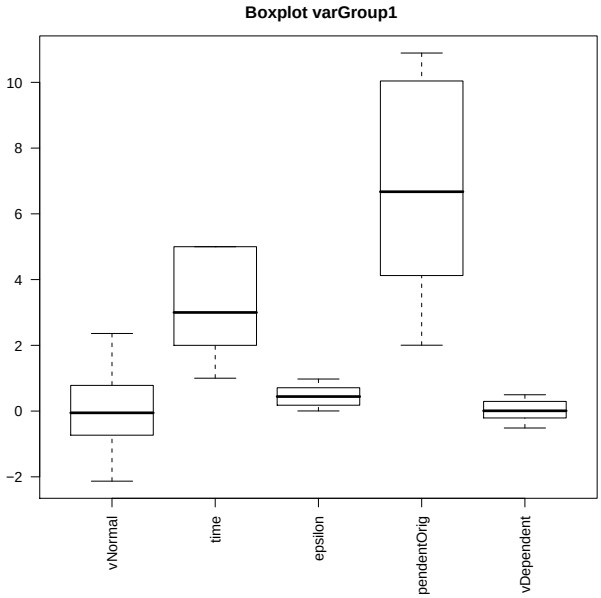


Figure 7: SDDataLinearModel, **boxplots** of the variables of the factor **varGroup1**.

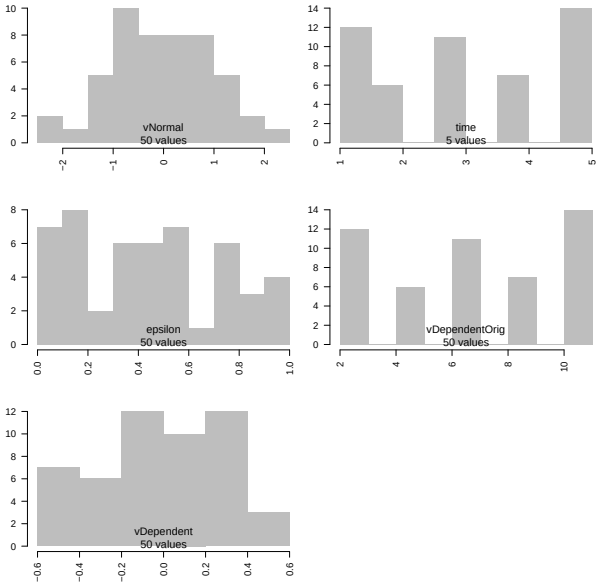


Figure 8: SDDataLinearModel, **histograms** of the variables of the factor **varGroup1**.

	mean	sd
vNormal	-1.82e-01	9.02e-01

Table 11: SDDataLinearModel summary of the different data treatments operated on the data.

	mean	sd	(Intercept) (SE; Pr(>|t|))	time (SE; Pr(>|t|))	R^2 (ad
vNormal	-1.82e-01	9.02e-01			
vDependent~time			0.53 (0.09; 9.1e-07)	1.97 (0.03; 8.4e-51)	0.9

Table 12: SDDataLinearModel summary of the different data treatments operated on the data.

Example: predict new data given previous estimates

```
> set.seed(6016)
> epsilon <- runif(30)
> time <- sample(1:5, 30, replace = TRUE)
> vDependent <- 2 * time + epsilon
> mat <- matrix(c(rnorm(30), time, epsilon, vDependent, vDependent), 30, 5)
> colnames(mat) <- c("vNormal", "time", "epsilon", "vDependentOrig", "vDependent")
> xLMPredicted <- predict(xLM, newdata = mat, prefix = "LMPredicted")
> summary(xLMPredicted, q = "lm", latex = TRUE, sanitize = FALSE)
```

	(Intercept) (SE; Pr(>|t|))	time (SE; Pr(>|t|))	R^2 (adj-R^2; N)
vDependent time	0.53 (0.09; 9.1e-07)	1.97 (0.03; 8.4e-51)	0.99 (0.99; 50)

Table 13: LMPredicted summary of the different data treatments operated on the data.

```
> summary(xLMPredicted, q = "mean|sd", latex = TRUE)
```

	mean	sd
vNormal	-1.82e-01	9.02e-01

Table 14: LMPredicted summary of the different data treatments operated on the data.

Example: Sample SDisc analysis
http://www.maxmind.com/app/state_latlon

```
> df <- read.csv("state.csv", row.names = 1)
> settings <- SDDataSettings(df, asCSV = "stateSettings.csv")
> settings <- read.csv2("stateSettingsEdited.csv", row.names = 1)

> x <- SDisc(df, settings = settings, prefix = "state")

 Prepare the data

 Load and test for consistency: state/IMAGE.RData
```

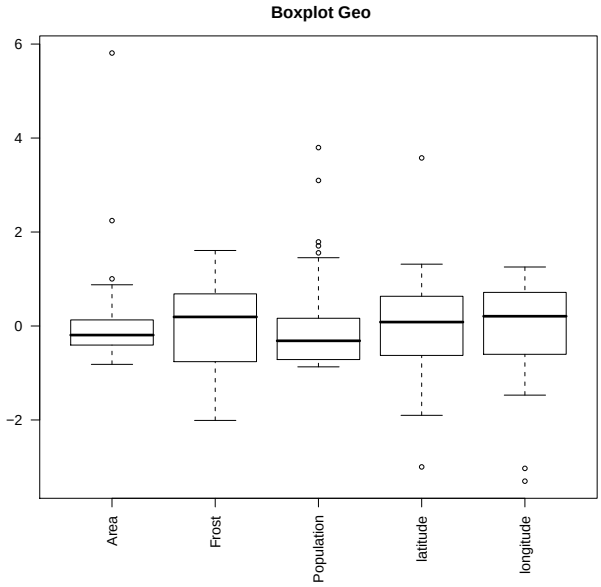


Figure 9: state, **boxplots** of the variables of the factor **Geo**.

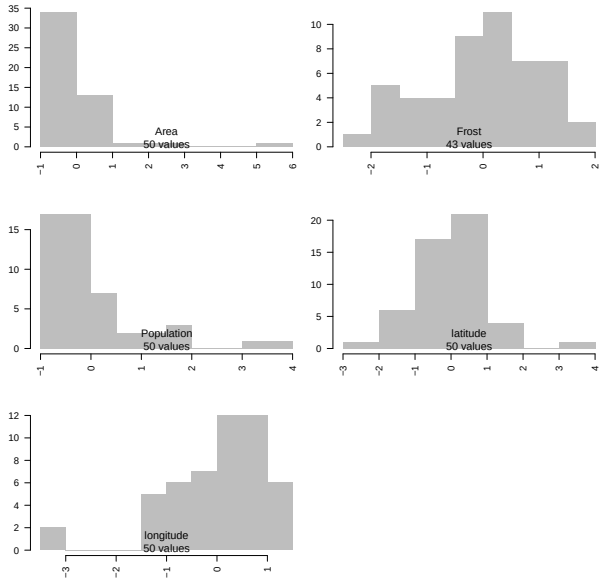


Figure 10: state, **histograms** of the variables of the factor **Geo**.

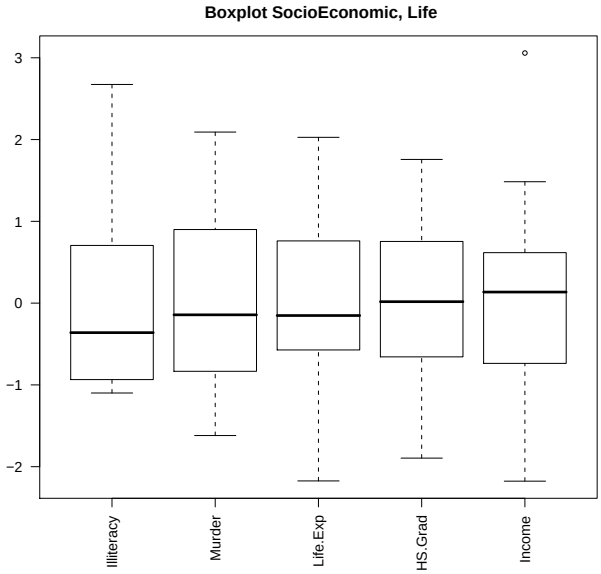


Figure 11: state, **boxplots** of the variables of the factor **SocioEconomic, Life**.

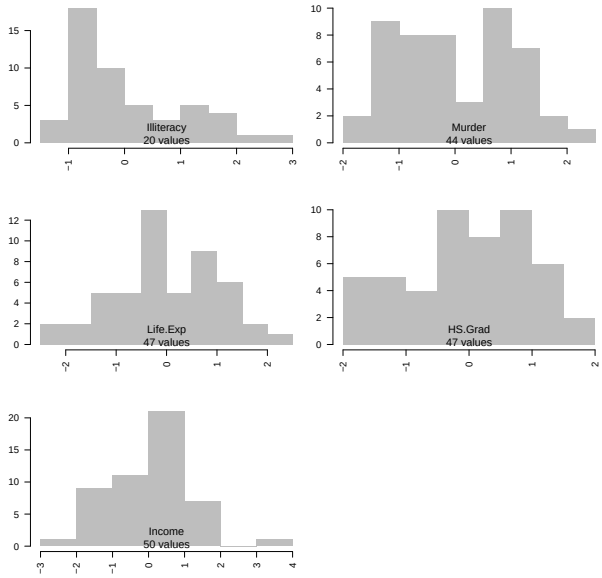


Figure 12: state, **histograms** of the variables of the factor **SocioEconomic, Life**.

	mean	sd
Area	7.07e+04	8.53e+04
Frost	1.04e+02	5.20e+01
Illiteracy	1.17e+00	6.10e-01
Murder	7.38e+00	3.69e+00
Life.Exp	7.09e+01	1.34e+00
HS.Grad	5.31e+01	8.08e+00
Population	4.25e+03	4.46e+03
Income	4.44e+03	6.14e+02
latitude	3.95e+01	6.12e+00
longitude	-9.37e+01	1.93e+01

Table 15: state summary of the different data treatments operated on the data.

```
> plot(SDData(x), latex = TRUE)
```

```
> summary(SDData(x), latex = TRUE)
```

```
> plot(x, latex = TRUE)
```

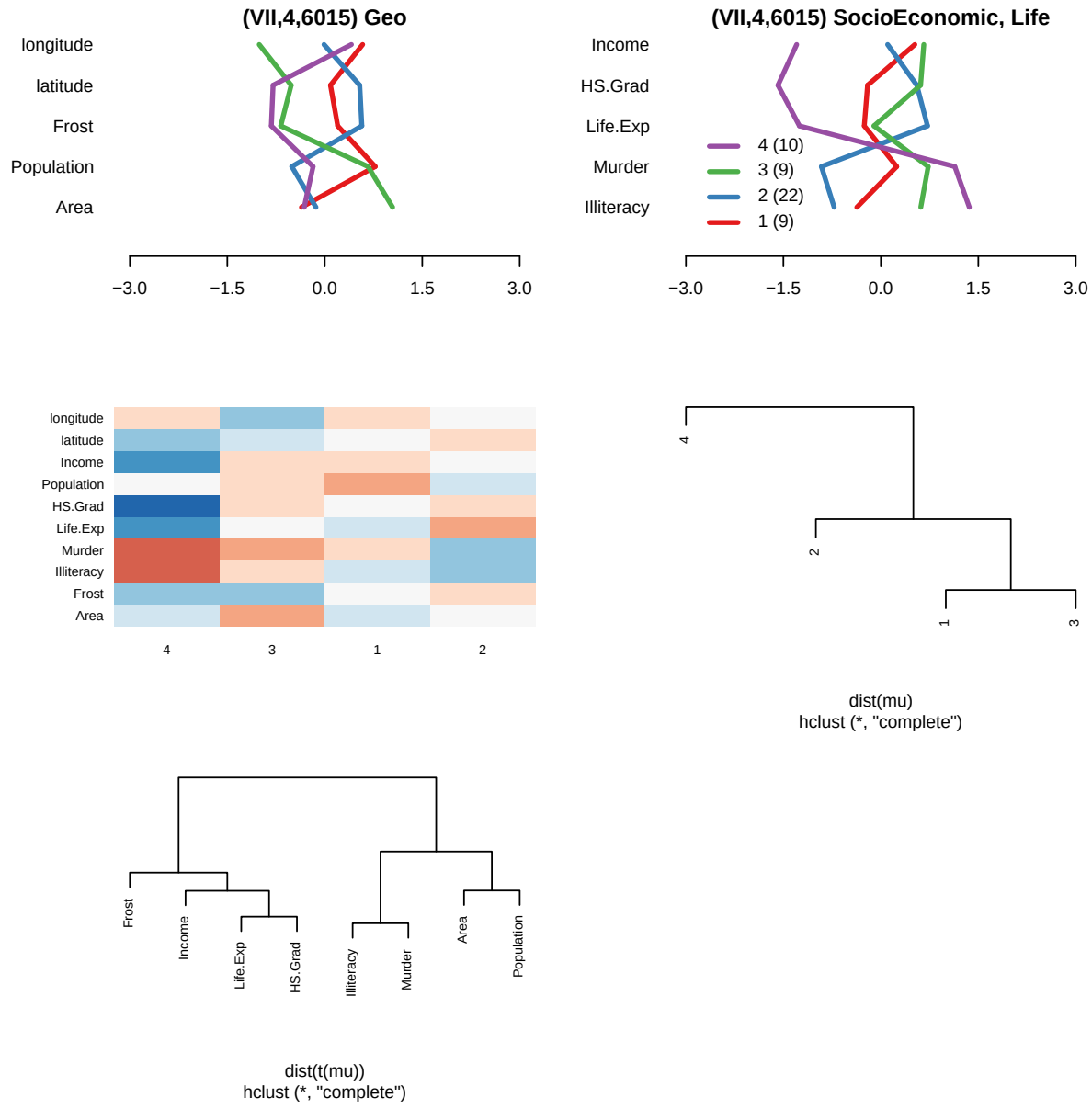
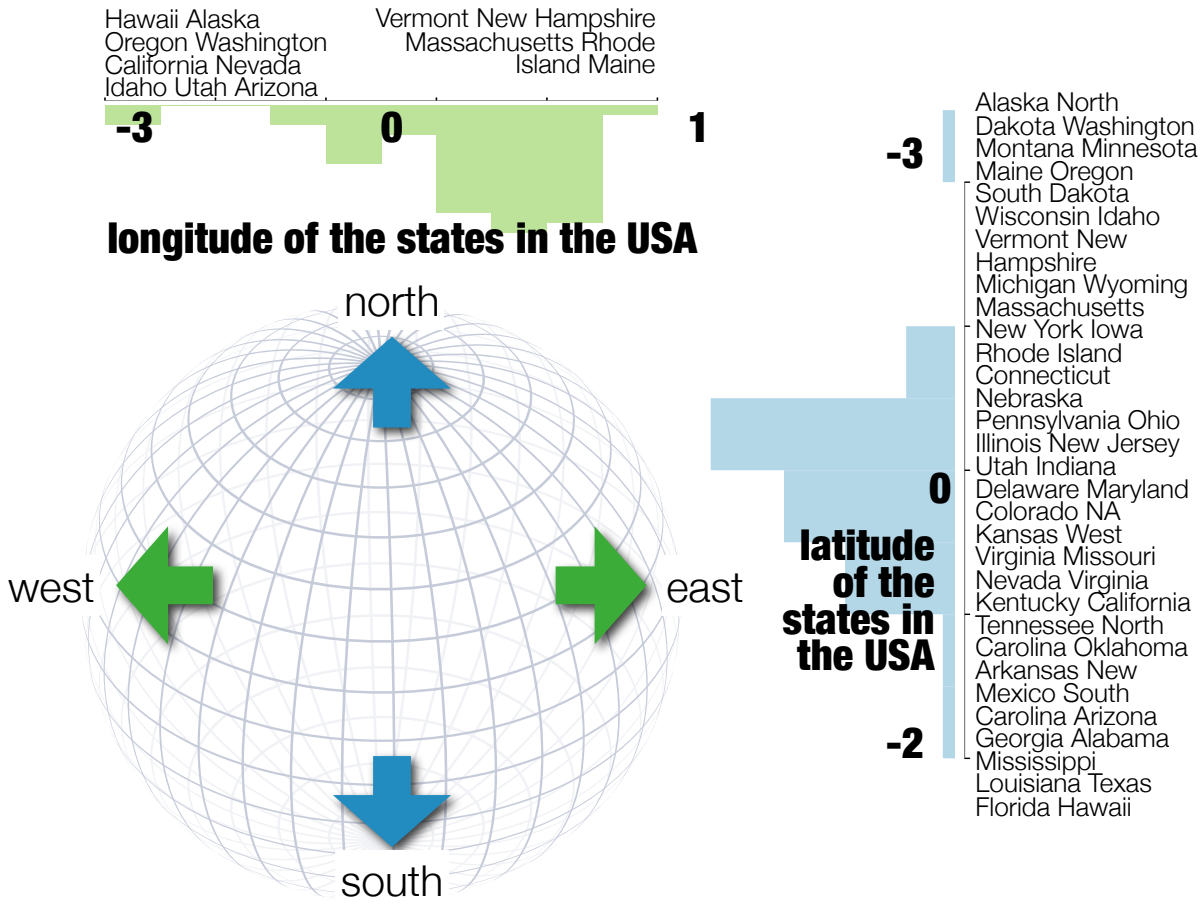


Figure 13: state, visual representation of **model** VII,4,6015.




```
> summary(x, q = 1, latex = TRUE)
```

	1	2	3	4
Geo	2.68	0.47	2.40	-12.51
Life	-0.56	-0.03	0.70	-0.37
SocioEconomic	1.65	0.03	12.10	-12.25
latitude	0.43	3.01	-0.46	-12.07
longitude	2.11	-0.60	-1.04	0.58

Table 16: state, (Bayesian) **oddratios** for the main factors in model VII,4,6015.

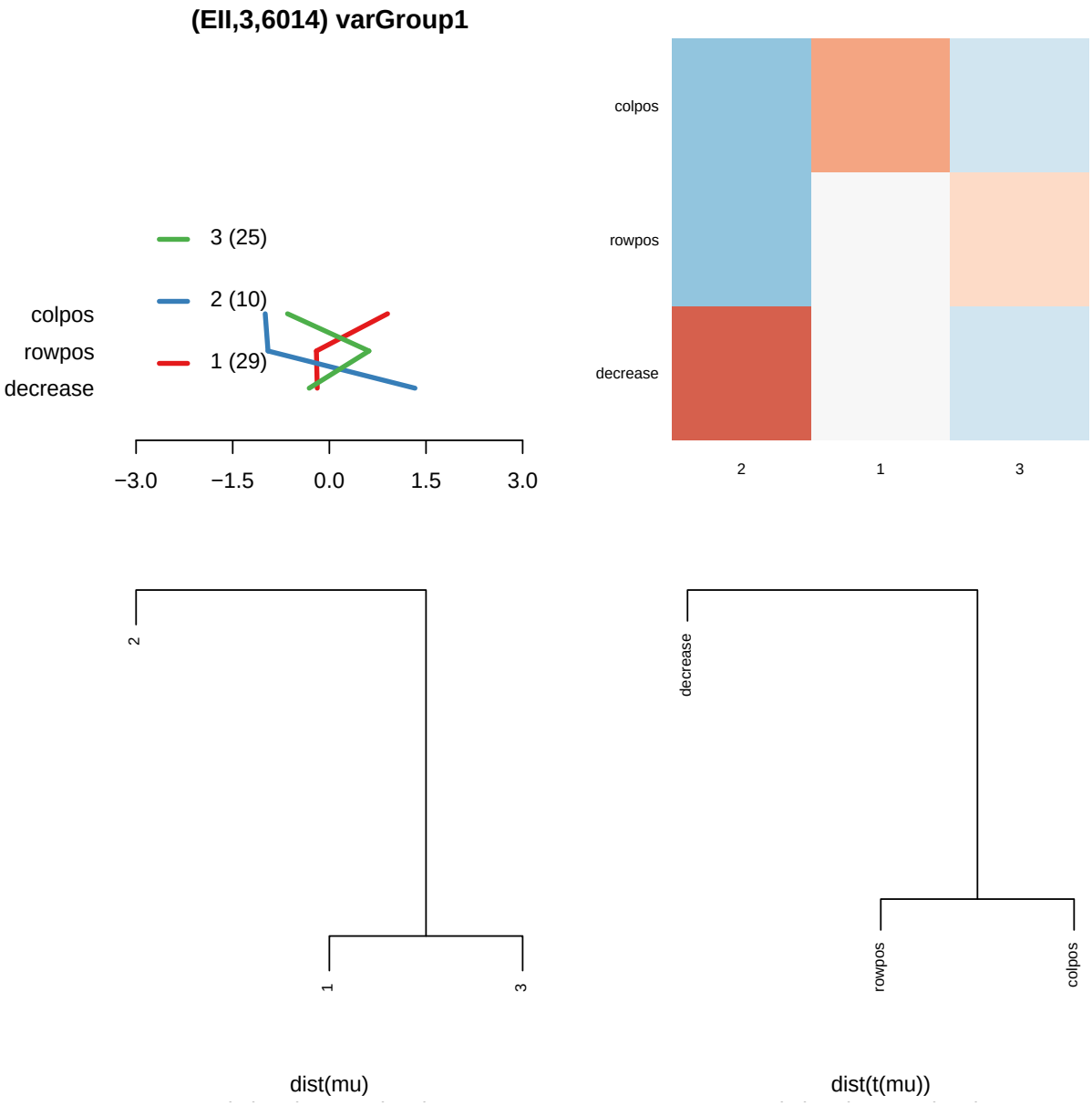


Figure 14: OrchardSprays, visual representation of **model** EII,3,6014.

Example: orchard sprays

```
> settings <- SDDataSettings(OrchardSprays)
> settings["treatment", ] <- NA
> x <- SDisc(OrchardSprays, settings = settings, prefix = "OrchardSprays")

 Prepare the data

 Load and test for consistency: OrchardSprays/IMAGE.RData

> plot(x, latex = TRUE)

> summary(x, latex = TRUE)

> summary(x, type = "chi2test", target = "treatment", latex = TRUE)
```

	1	2	3
colpos	5.67	-11.13	-2.84
decrease	1.46	11.94	-3.10
rowpos	1.12	-3.05	0.80

Table 17: OrchardSprays, (Bayesian) **oddratios** for the main factors in model EII,3,6014.

	1	2	3
1	0.722	-1.118	-0.071
2	-0.328	-1.118	1.061
3	0.197	-0.224	-0.071
4	0.197	-0.224	-0.071
5	0.197	0.671	-0.636
6	-0.328	-0.224	0.495
7	0.197	-0.224	-0.071
8	-0.853	2.460	-0.636

Table 18: For **treatment**: $p_{\chi^2} = 0.560$ ($\chi^2 = 13.0$) in model EII,3,6014.

List of Tables

List of Figures

List of institutes and main investigators

LIACS, F Colas, Dr [Colas, 2009a, Colas et al., 2008a, Colas et al., 2008b]. *Leiden Institute of Advanced Computer Science, Leiden University, NL-2333CA Leiden, the Netherlands*

References

[Colas, 2009a] Colas, F. (2009a). *Data Mining Scenarios for the Discovery of Subtypes and the Comparison of Algorithms*. PhD thesis, Leiden University.

[Colas, 2009b] Colas, F. (2009b). *R SubtypeDiscovery Vignette: a data mining scenario for the inference of subtypes by cluster analysis*. LIACS, Leiden University.

[Colas et al., 2008a] Colas, F., Meulenbelt, I., Houwing-Duistermaat, J. J., Kloppenburg, M., Watt, I., van Rooden, S. M., Visser, M., Marinus, J., Cannon, E. O., Bender, A., van Hilten, J. J., Slagboom, P. E., and Kok, J. N. (2008a). A scenario implementation in r for subtype discovery examplified on chemoinformatics data. In *ISoLA*, pages 669–683.

[Colas et al., 2008b] Colas, F., Meulenbelt, I., Houwing-Duistermaat, J. J., Kloppenburg, M., Watt, I., van Rooden, S. M., Visser, M., Marinus, J., van Hilten, J. J., Slagboom, P. E., and Kok, J. N. (2008b). Stability of clusters for different time adjustments in complex disease research. In *30th Annual International IEEE EMBS Conference (EMBC'08), Vancouver, British Columbia, Canada*.