R SDisc: Integrated methodology for data subtype discovery

Fabrice Colas*

October 28, 2011

Cluster analysis[1] is a statistical technique that aims to subset observations into groups, such that similar items are in the same clusters but are very different from items in other clusters. As a discovery tool, cluster analysis may enable to reveal associations, patterns, relationships, and structure in data. R SDisc is an additional tool to perform cluster analysis.

However, instead of proposing another clustering algorithm to the vast landscape of existing techniques [7], we focused on the development of a pipelined clustering analysis tool that would integrate the necessary tools and methods to run a complete analysis from data processing to subtype validation [1, 3]. It has been primarily designed for, and applied to clinical research on complex pathologies like Parkinson's disease [9], aggressive brain tumours [2] and Osteoarthritis where, more homogeneous patient subtypes from clinical predictors are sought for in order to break down the known clinical heterogeneity of those diseases (one disease-umbrella, different manifestations).

As such, R SDisc includes methods for data treatment and pre-processing, repeated cluster analysis, model selection, model reliability [4] and reproducibility assessment, subtype characterization and validation by visual and table summaries. In the design of R SDisc, we emphasized especially the validity of the inference steps, the accessibility of the cluster analysis protocol, the reproducibility of the results, and the availability as an open source package of the technique. This vignette is an interactive documentation on R SDisc.

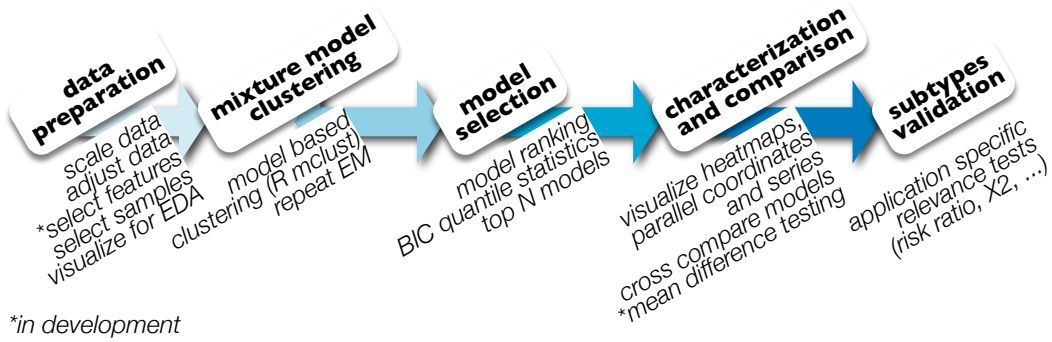

Figure 1: Our pipelined cluster analysis describes as a sequence of steps [3]: data preparation, cluster modeling based on [5, 6], model selection, characterization, comparison of the subtypes and relevance evaluation.

*Leiden University Medical Center, Einthovenweg 20, 2300RC Leiden, the Netherlands

[1] http://www.businessdictionary.com/definition/cluster-analysis.html

Contents

1 Installation

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

```
> library(SDisc)
```

2 Analysing Iris in 10 Lines

In 10 lines of R code, we carry a straightforward SDisc analysis.

First, we create a settings file describing the dataset with SDDataSettings that we update to remove the class Species from the modeling.

```
> iris.set <- SDDataSettings(iris)
> iris.set["Species", ] <- c(NA, "FALSE", NA, NA, NA, NA)
```

Given those settings, with SDData we make an SDisc dataset container, defining the root directory of that analysis by the prefix argument. Next, we verify the transformation with print by reporting a random extract of the original and the transformed datasets. With plot we get regular annotated boxplots, and histograms. With summary, we get the estimates of transformation statistics.

```
> iris.dat <- SDData(iris, prefix = "iris", settings = iris.set)
> print(iris.dat, rseed = 6014, latex = TRUE)
```

	Sepal.Length	Petal.Width	Petal.Length
14	4.30	0.10	1.10
48	4.60	0.20	1.40
62	5.90	1.50	4.20

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

```
> plot(iris.dat, latex = TRUE)
```

```
> summary(iris.dat, latex = TRUE)
```

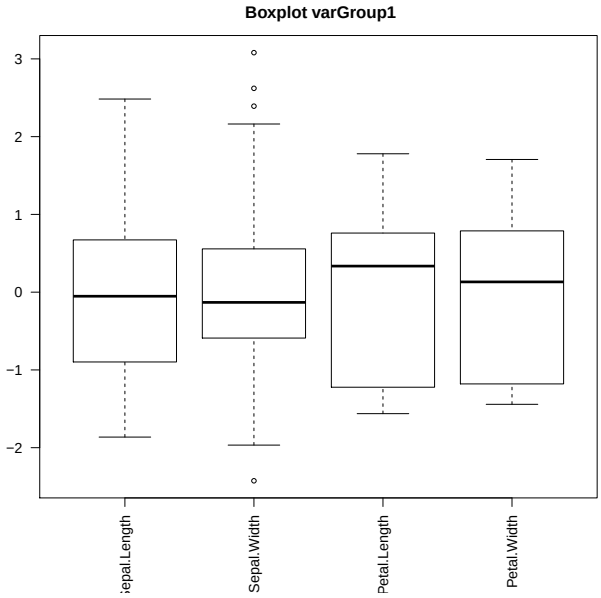

Figure 2: iris, **boxplots** of the variables of the factor **varGroup1**.

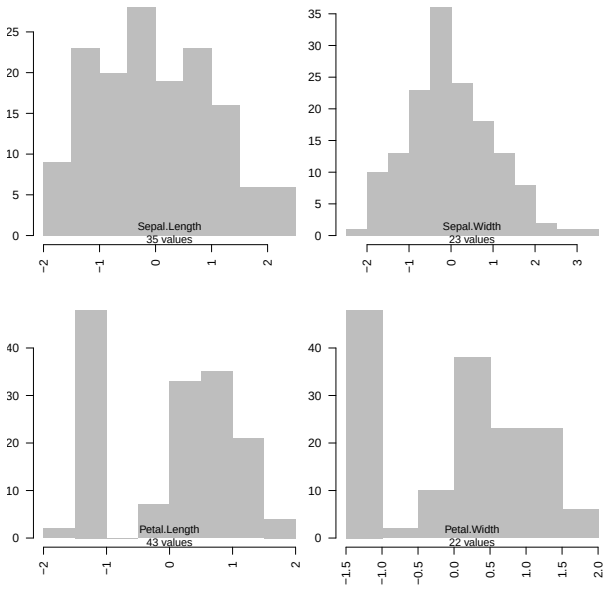

Figure 3: iris, **histograms** of the variables of the factor **varGroup1**.

	Sepal.Length	Petal.Width	Petal.Length
14	-1.86	-1.44	-1.51
48	-1.50	-1.31	-1.34
62	0.07	0.39	0.25

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

	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: iris summary of the different data treatments operated on the data.

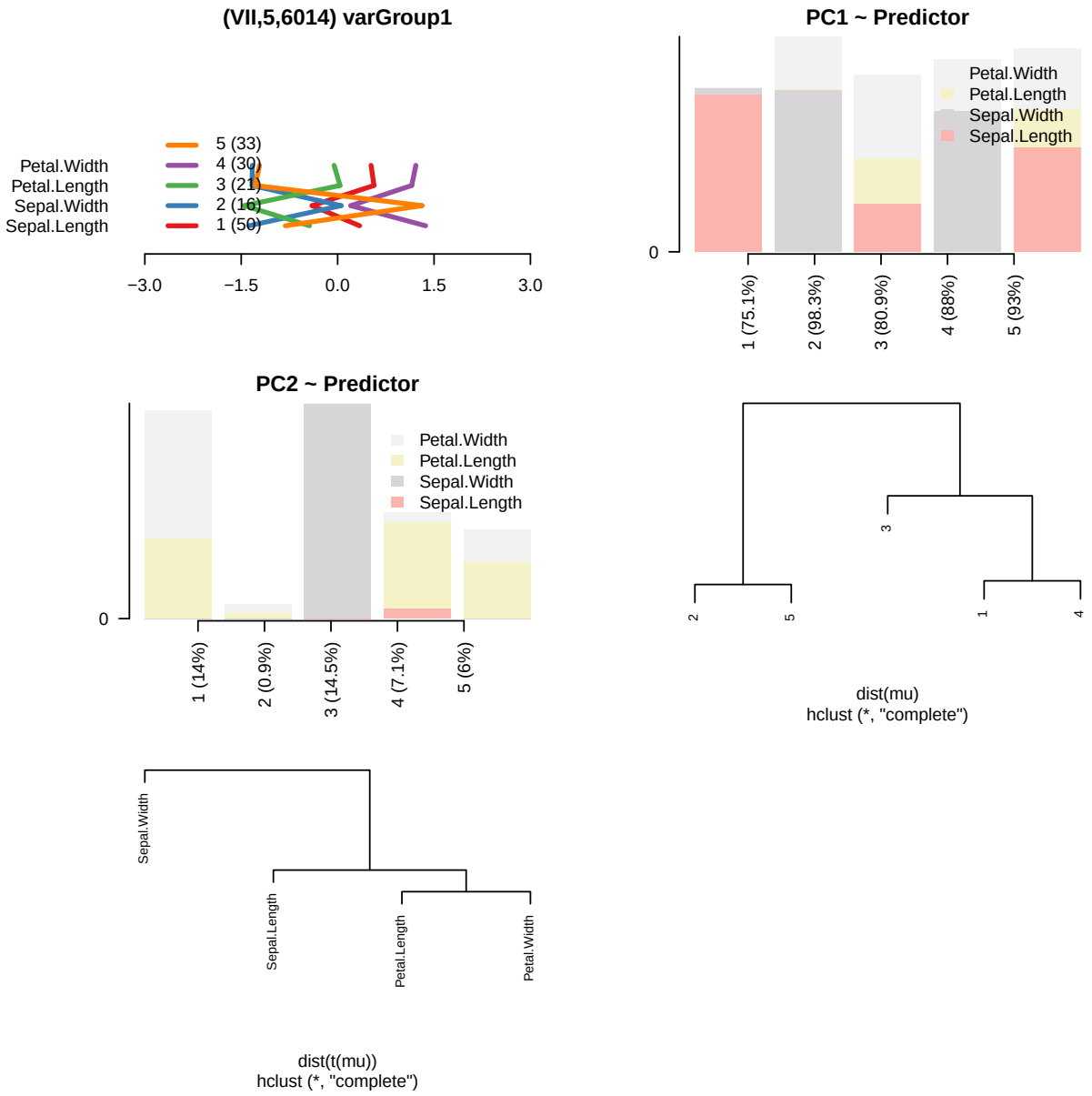

Figure 4: iris, visual representation of **model** VII,5,6014.

Using that SDisc dataset container, we carry a subtyping analysis with default parameters.

```
> iris.sd <- SDisc(iris.dat)
```

With plot, we summarize graphically the analysis with parallel coordinates, the loadings of ANOVA analyses *per* subtype to find the most influencial covariates with respect to each subtype, and two dendrograms -one on the samples, one on the covariates. The print method reports the joint distribution of the most likely models, along with κ- and χ^2-statistics. Finally, the summary method gives a numerical summary.

```
> plot(iris.sd, latex = TRUE)

> print(iris.sd, latex = TRUE)

> summary(iris.sd, latex = TRUE)
```

	3	4	1	2	5
1	50				
4		30			
3			21		
2					16
5				13	20

Table 4: iris, the **comparison of model** VII,5,6014 and VII,5,6013 exhibits a random index 91.3 (a $\kappa = 73.5$, and a relative degree of association $V = 89.1\%$ with $p_{\chi^2} = 0.0005$, $\chi^2 = 476.3$).

	1	2	3	4	5
Petal.Length	1.77	-11.63	-1.41	14.09	-14.07
Petal.Width	3.13	-12.24	-0.81	14.54	-14.69
Sepal.Length	2.04	-12.50	-0.99	15.01	-4.82
Sepal.Width	-1.16	1.20	-14.25	0.09	14.29

Table 5: iris, (Bayesian) **oddratios** for the main factors in model VII,5,6014.

3 SDisc Data Container

R SDisc implements its own data structure. The reason we implemented this structure is because we like to preserve a copy of the original data when transforming the data by either of feature selection, complete cases filering, normalization. Also, this approach enable us to restrict the cluster analysis to only a subset of the predictors, which is sometimes desirable.

Data Example We simulate an example called `normdep.df` with three predictors; the first follows $v_1 \sim \mathcal{N}(0, 1)$, the second is a time variable $v_2 = t$ and the third depends on the time with $v_3 = 2 \times t + \epsilon$.

```
> set.seed(6015)
> time <- sample(1:5, 50, replace = TRUE)
> normdep.df <- matrix(c(rnorm(50), time, 2 * time + runif(50)), 50,
      3, dimnames = list(list(), c("norm", "t", "v")))
```

Also, we inject 5 missing values at random into the data.

```
> normdep.df[sample(1:nrow(normdep.df))[1:5]] <- NA
```

Data Transformation A common initial step in data analysis is to normalize the data by centering around zero with unit variance or by adjusting for some known effect like age, bmi, gender. Therefore, we define in a matrix the particular transformation to apply to each predictor.

In addition, we also define how predictors are ordered one to each other and whether predictors belong to the same group. While ordering information is used to arrange the graphical summaries (heatmaps, parallel coordinates), grouping is used to calculate odd ratios based on the aggregated sum of groups of predictors. Last, an indicator variable (`inCAnalysis`) identifies those predictors to include in the cluster analysis.

To do the data definition step, we use `SDDataSettings` that generates a sample settings file to be saved `asCSV`. It can be edited from within R or from Excel. In the case

`SDDataSettings` receives an `SDisc` or an `SDData`, it returns the `settings` of the current `SDisc` analysis or of the current `SDData` container. When `latex` is set to `TRUE`, a LaTeX-formated output is returned for use with Sweave reporting mechanism.

```
> normdep.set <- SDDataSettings(normdep.df)
> normdep.set[, "tFun"] <- c("mean sd", "", "lm(v~t)")
> normdep <- SDData(normdep.df, settings = normdep.set, prefix = "normdep")

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

	oddGroup	inCAnalysis	tFun	vParGroup	vParY	vHeatmapY
norm	norm	TRUE	mean sd	varGroup1	1	1
t	t	TRUE		varGroup1	2	2
v	v	TRUE	lm(v~t)	varGroup1	3	3

Table 6: SDDataSettings

Data Exploration Prior to cluster analysis, we proceed to elementary exploratory analysis of the data. With `naPattern`, we either return a character vector of the record IDs that present missing values row-wise or a table (latex=TRUE) report of the missingness pattern for each record presenting at least one missing value.

```
> naPattern(normdep)

[1] "14" "20" "26" "38" "45"

> naPattern(normdep, latex = TRUE)
```

	isNA	isNotMissing	naRate
14	1.00	2.00	33.33
20	1.00	2.00	33.33
26	1.00	2.00	33.33
38	1.00	2.00	33.33
45	1.00	2.00	33.33

Table 7: normdep, index of the observations presenting **missing values** along with the number of missings and non-missings; the observations with a missing value represent 10.00% of the available observations.

The method `print.SDisc` does return the transformed data matrix after the transformation. To verify the validity of the transformations, an `rseed` can be provided to the method in order to report side by side a random extract of the originals and of the transformed data matrix. The `range` parameter gives the number of rows and columns to extract randomly.

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

With `plot.SDData`, exploratory plots such as boxplots and histograms are reported for each predictors. If `latex` is set to TRUE, then the LaTeXcode to include the different figures is returned into the standard output for use with Sweave.

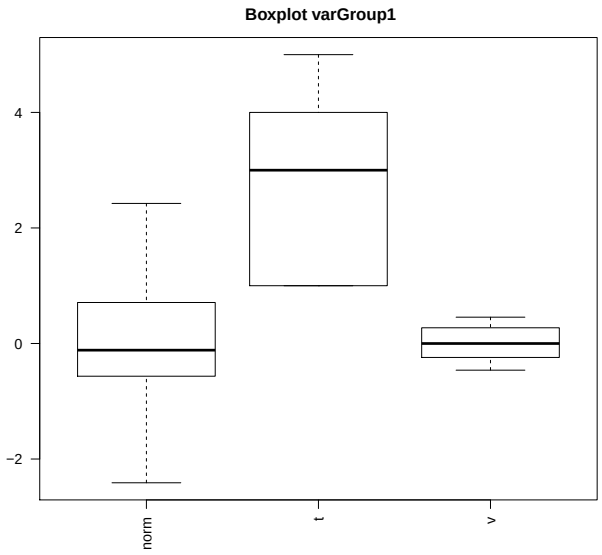

Figure 5: normdep, **boxplots** of the variables of the factor **varGroup1**.

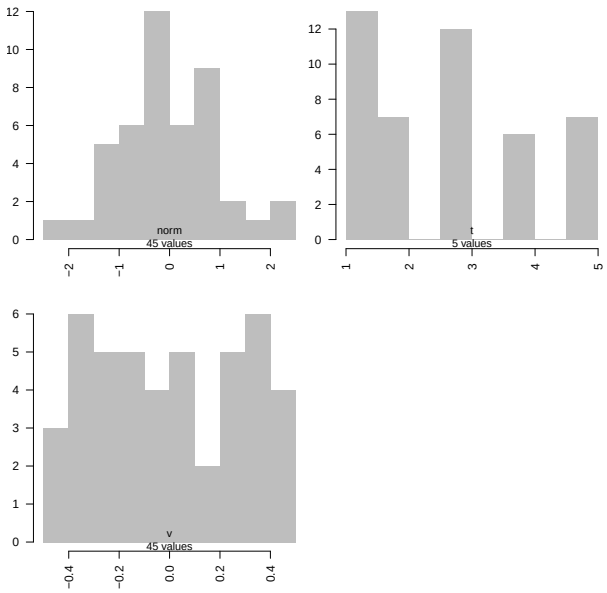

Figure 6: normdep, **histograms** of the variables of the factor **varGroup1**.

	t	v	norm
27	2.00	4.23	0.04
36	2.00	4.68	0.28
21	1.00	2.06	-0.92

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

	t	v	norm
27	2.00	0.27	-0.13
36	2.00	-0.17	0.13
21	1.00	0.45	-1.15

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

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

The `summary.SDData` method provides a summary of the data transformation process. The `q` parameter limits the summary to a subset of the data treatments by regular expression on the data `settings`-file. For mean, sd, scale, max, min it returns the estimates and for lm it reports the coefficient estimates as well as the standard errors, p-value, R^2, adjusted R^2 and number of records on which the estimate was based on.

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

	mean	sd
norm	1.58e-01	9.36e-01

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

```
> summary(normdep, q = "lm", latex = TRUE)
```

	(Intercept) (SE; Pr(>|t|))	t (SE; Pr(>|t|))	R^2 (adj-R^2; N)
v~t	0.51 (0.09; 2.6e-06)	2.00 (0.03; 2.1e-44)	0.99 (0.99; 45)

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

Applying Previous Transformation Estimates to New Data We illustrate how the transform estimates from the `normdep` data are re-used to transform `normdep.df2`, a new data set generated from the same process but with a different random seed.

```
> set.seed(6016)
> time <- sample(1:5, 30, replace = TRUE)
> normdep.df2 <- matrix(c(rnorm(30), time, 2 * time + runif(30)),
      30, 3, dimnames = list(list(), c("norm", "t", "v")))
> normdep2 <- predict(normdep, newdata = normdep.df2, prefix = "normdep2")

> summary(normdep2, q = "lm", latex = TRUE, sanitize = FALSE)

> summary(normdep2, q = "mean|sd", latex = TRUE)
```

	(Intercept) (SE; Pr(>|t|))	t (SE; Pr(>|t|))	R^2 (adj-R^2; N)
v t	0.51 (0.09; 2.6e-06)	2.00 (0.03; 2.1e-44)	0.99 (0.99; 45)

Table 12: normdep2 summary of the different data treatments operated on the data.

	mean	sd
norm	1.58e-01	9.36e-01

Table 13: normdep2 summary of the different data treatments operated on the data.

χ^2 **Feature Selection of Spectral Data** As used in [2]: TO DO

4 Subtyping with SDisc

4.1 Cluster Modeling

We base our cluster analysis procedure on the model based clustering framework (`mclust`) from Fraley and Raftery [5, 6]. As the EM model likelihood optimization procedure is sensible to its starting value, Fraley and Raftery start EM with a (model based) hierarchical clustering. Yet, it may happen that initializing EM by the hierarchical clustering does not lead to the most likely model. To address this uncertainty, we start EM from a series of random initialization points (here `rseed` $\in [6013; 6023]$) and we study the class of resulting models.

```
> normdep <- SDisc(normdep, settings = normdep.set, prefix = "normdep",
       cFunSettings = list(modelName = c("EII", "VII", "VEI", "VVI"),
           G = 3:5, rseed = 6013:6023))
```

4.2 Selecting Between Cluster Models

The larger the number of parameters, the more likely the model overfits the data, which restricts its generality and understandability. For model selection, Kass and Raftery [8] prefer the Bayesian Information Criterion (BIC) to the Akaike Information Criterion (AIC) because it approximates the Bayes Factor. In our protocol, we use the BIC which defines as:

$$BIC = -2 \log \mathcal{L}_{MIX} + \log (N \times \#params), \quad (1)$$

with $\mathcal{L}_{MIX}$ the Gaussian-mixture model likelihood, N the number of observations and $\#params$ the number of parameters of the model.

Yet, we found inappropriate to select for a model based on a single BIC value because we questioned whether models 'less likely' (e.g. $< 5\%$) were significantly different or not, whether local or global maxima were attained by EM, and if cluster results from different starting values of EM were *reliable* (consistency).

To address these questions, we repeat the model based cluster analysis by initializing EM from a number of starting values. Here, we rank models based on their BIC values, which enables to retrieve a subset of top-ranking models among which there may be more simple models with less parameters.

```
> summary(bicTable(normdep), latex = TRUE)

> print(bicTable(normdep), modelName = "VII", G = 4, latex = TRUE)
```

	EII	VII	VEI	VVI
3	16.63 (16.63, 16.80)	16.55 (16.55, 17.93)	0.00 (0.00, 8.67)	NA (10.56, 10.57)
4	15.40 (15.40, 15.41)	14.21 (14.44, 18.64)	2.75 (2.75, 6.84)	NA (8.87, 9.62)
5	14.95 (14.95, 16.90)	16.88 (16.88, 21.68)	5.03 (5.03, 7.21)	NA (14.17, 15.74)

Table 14: normdep, model VEI,3,6013 shows the **highest BIC** score over: the repeated random starts, type of model and number of component.

	modelName	G	rseed	BIC	relativeBic
VII,4,6013	VII	4	6013	-374.15	14.21
VII,4,6019	VII	4	6019	-377.23	15.15
VII,4,6022	VII	4	6022	-377.23	15.15
VII,4,6023	VII	4	6023	-387.94	18.42
VII,4,6017	VII	4	6017	-387.94	18.42
VII,4,6015	VII	4	6015	-387.94	18.42
VII,4,6018	VII	4	6018	-387.94	18.42
VII,4,6016	VII	4	6016	-387.94	18.42
VII,4,6020	VII	4	6020	-387.94	18.42
VII,4,6021	VII	4	6021	-388.66	18.64
VII,4,6014	VII	4	6014	-388.66	18.64

Table 15: Top ranking **normdep** models.

4.3 Comparing and Characterizing Subtypes

To report the main characteristics of the clusters, we use odd ratios calculated on the group of predictors (see `SDDataSettings`). Based on Table 16, this ratio is calculated as follows

$$logodds_{kl} = \log \frac{A \times D}{B \times C}. \tag{2}$$

Table 16: For each sum score l, we consider a middle value δ_l such as the dataset *mean* or *median*. For cells A and B, we use it to count how many observations i in the cluster S_k have a sum score above and below its value. For cells C and D, we proceed to a similar count but on the rest of the observations $i \in \{S - S_k\}$.

	$x_i < \delta_l$	$x_i \geq \delta_l$
$i \in S_k$	A	B
$i \in \{S - S_k\}$	C	D

To compare clusters two by two, we report the joint distribution of the cluster affectations. If the table has many empty cells, then the two cluster results are highly related; if the joint distribution over all cells is even, then the two cluster results are unrelated (independent).

We summarize those tables by the rand index that measures the similarity between two data clusterings, the χ^2 statistics, the Cramer's V which, similarly to the Pearson's correlation coefficient takes one for completely correlated variables and zero for stochastically

independent ones; V is defined as follows

$$V = \sqrt{\frac{\chi^2}{N \times m}} \tag{3}$$

with $V \in [0; 1]$ and N the sample size and $m = min(rows, columns) - 1)$.

```
> print(normdep, latex = TRUE)
```

	3	1	2
1	21		
2			4
3		20	

Table 17: normdep, the **comparison of model** VEI,3,6013 and VEI,3,6015 exhibits a random index 100.0 (a $\kappa = 100.0$, and a relative degree of association $V = 100.0\%$ with $p_{\chi^2} = 0.0005$, $\chi^2 = 90.0$).

```
> bestAll <- bestModel(normdep)[1]
> bestAll

[1] "VEI,3,6013"

> bestG4 <- bestModel(normdep, G = 4)[1]
> bestG4

[1] "VEI,4,6013"

> print(normdep, m1 = bestAll, m2 = bestG4, latex = TRUE)
```

	1	2	3	4
1	20		1	
2		4		
3			9	11

Table 18: normdep, the **comparison of model** VEI,3,6013 and VEI,4,6013 exhibits a random index 100.0 (a $\kappa = 100.0$, and a relative degree of association $V = 97.8\%$ with $p_{\chi^2} = 0.0005$, $\chi^2 = 86.0$).

To compare visually the characteristics of the cluster results, we use heatmaps, parallel coordinates and dendrograms. With heatmaps we report the mean/median/quantile patterns of the clusters. With parallel coordinates we show the mean/median/quantile patterns, bringing forward the information about ordering and grouping of predictors. With dendrograms, we report the similarity between the mean/median patterns of each cluster result or group of predictor.

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

	1	2	3
norm	3.08	-10.72	-0.13
t	1.03	-13.43	-0.24
v	-3.84	-10.70	11.80

Table 19: normdep, (Bayesian) **oddratios** for the main factors in model VEI,3,6013.

4.4 Testing Validity and Reproducibility of Subtypes

to do

A Session Info

```
> sessionInfo()

R version 2.13.1 (2011-07-08)
Platform: x86_64-apple-darwin9.8.0/x86_64 (64-bit)

locale:
[1] C/en_GB.UTF-8/C/C/en_GB.UTF-8/en_GB.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
 [1] SDisc_1.24          SparseM_0.89        snow_0.3-6          e1071_1.5-26
 [5] class_7.3-3         digest_0.5.0        xtable_1.5-6        abind_1.3-0
 [9] RColorBrewer_1.0-5 mclust_3.4.10

loaded via a namespace (and not attached):
[1] tools_2.13.1
```

References

[1] Fabrice Colas. *Data Mining Scenarios for the Discovery of Subtypes and the Comparison of Algorithms*. phd, Leiden University, 2009.

[2] Fabrice Colas, Joost N. Kok, and Alfredo Vellido. Finding discriminative subtypes of aggressive brain tumours using magnetic resonance spectroscopy. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, 1:1065–8, 2010 2010.

[3] Fabrice Colas, Ingrid Meulenbelt, Jeanine J. Houwing-Duistermaat, Margreet Kloppenburg, Iain Watt, Stephanie M. van Rooden, Martine Visser, Johan Marinus, Edward O. Cannon, Andreas Bender, Jacobus J. van Hilten, P. Eline Slagboom, and Joost N. Kok. A scenario implementation in r for subtypediscovery examplified on chemoinformatics data. In *Leveraging Applications of Formal Methods, Verification and Validation, Communications in Computer and Information Science*, volume 17, pages 669–683. Springer Berlin Heidelberg, Springer Berlin Heidelberg, 2008.

[4] Fabrice Colas, Ingrid Meulenbelt, Jeanine J. Houwing-Duistermaat, Margreet Kloppenburg, Iain Watt, Stephanie M. van Rooden, Martine Visser, Johan Marinus, Jacobus J. van Hilten, P Slagboom, and Joost N. Kok. Reliability of cluster results for different types of time adjustments in complex disease research. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, 2008:4601–4, 2008 2008.

[5] C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis and density estimation. *Journal of the American Statistical Association*, 97:611–631, 2002.

[6] C. Fraley and A. E. Raftery. MCLUST version 3 for R: Normal mixture modeling and model-based clustering. Technical Report 504, University of Washington, Department of Statistics, September 2006.

[7] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31:264–323, September 1999.

[8] Robert E. Kass and Adrian E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795, 1995.

[9] Stephanie M. van Rooden, Fabrice Colas, Pablo Martínez-Martín, Martine Visser, Dagmar Verbaan, Johan Marinus, Ray K Chaudhuri, Joost N. Kok, and Jacobus J. van Hilten. Clinical subtypes of parkinson's disease. *Movement disorders : official journal of the Movement Disorder Society*, 2010 Nov 16 2010.