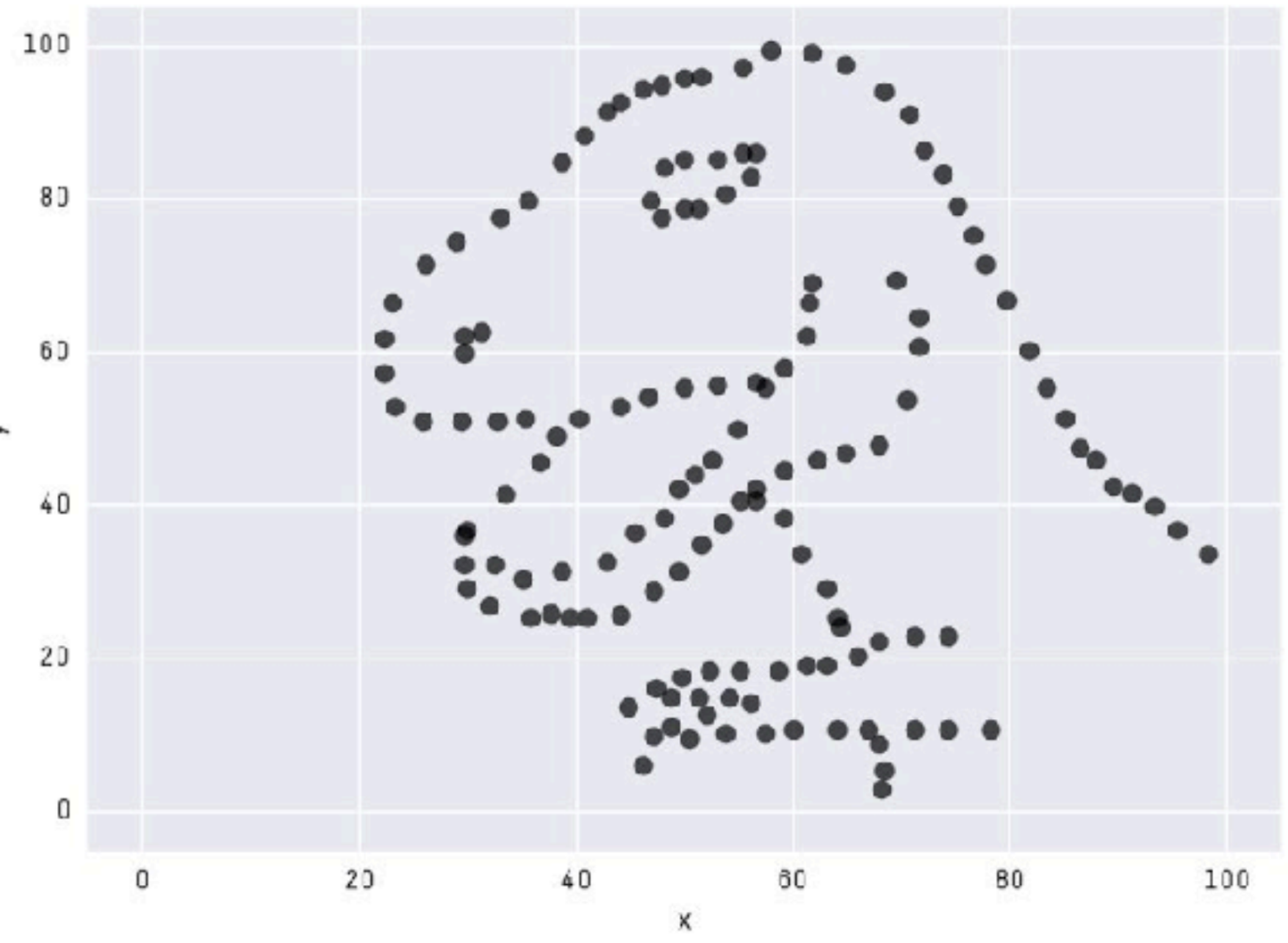
Show me your predictive model

Does datavis help to understand models?
Why it's important?
What tools do we have?

Przemysław Biecek
University of Warsaw
Warsaw University of Technology

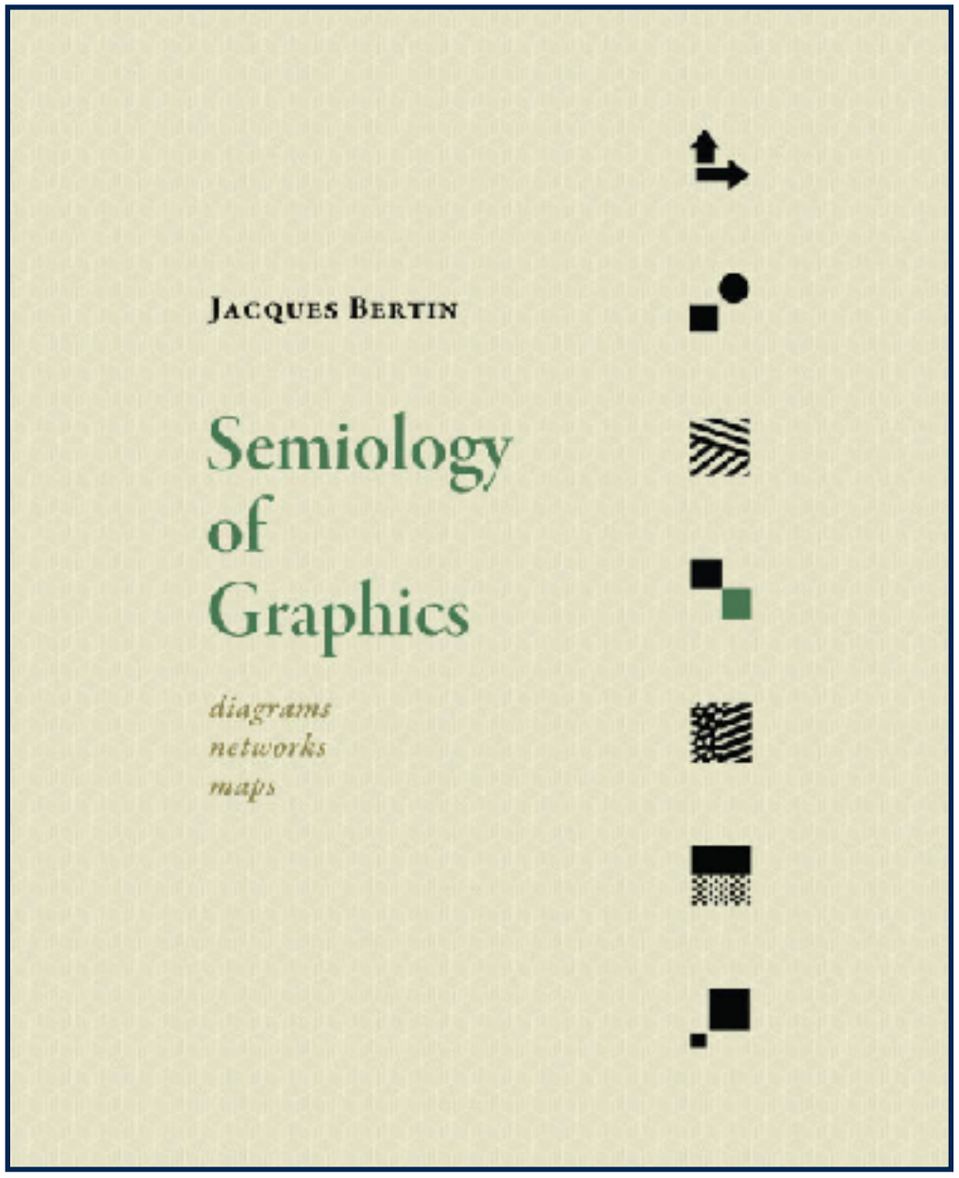
Do we really need plots to understand numbers?

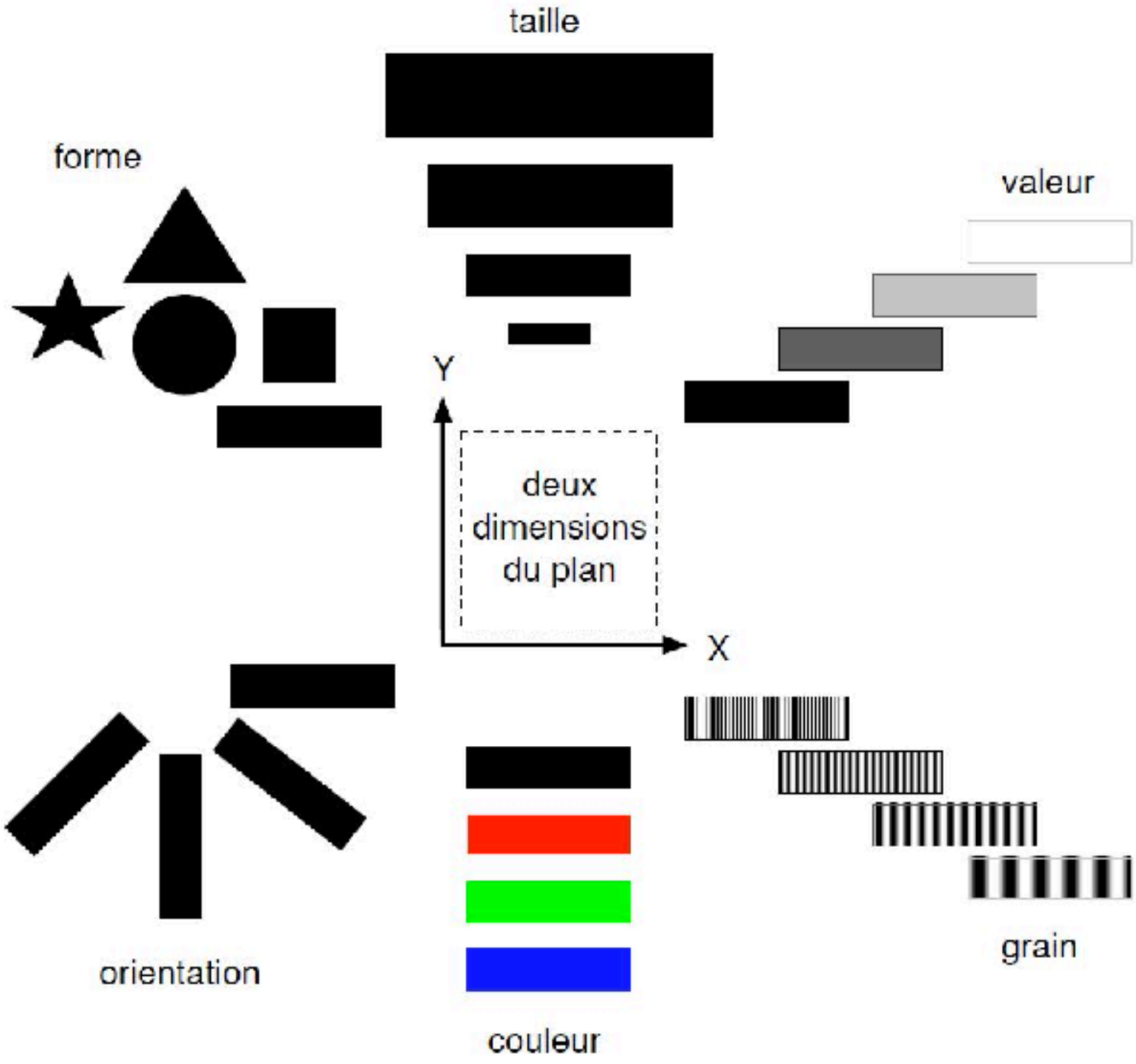
Package datasauRus



X Mean: 54.2632025
Y Mean: 47.8315781
X SD : 16.7650109
Y SD : 26.9353144
Corr. : -0.0645195

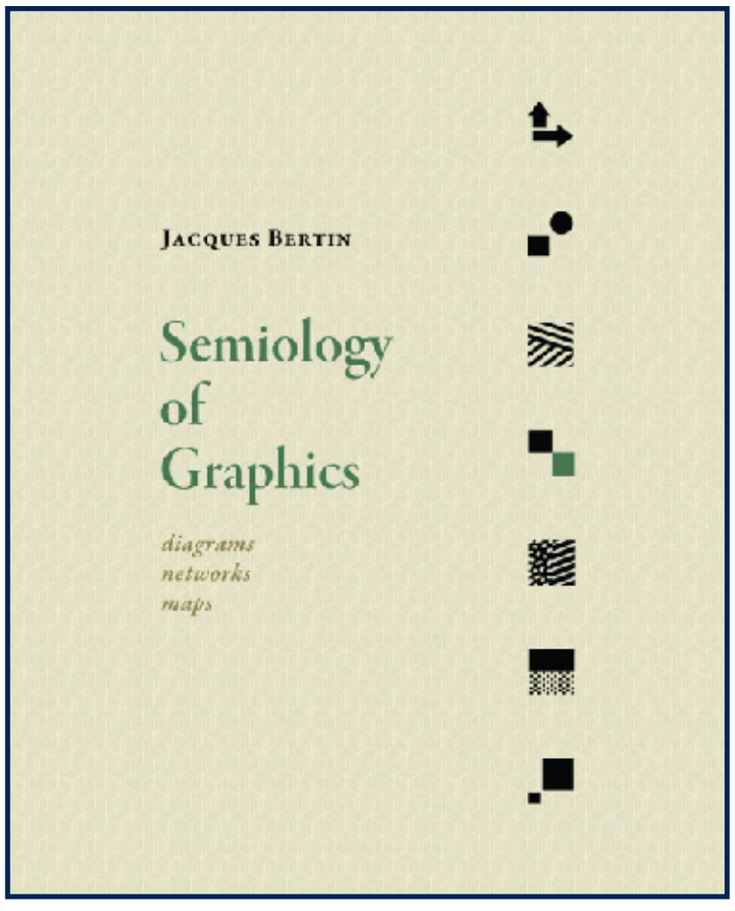
https://www.autodeskresearch.com/publications/samestats

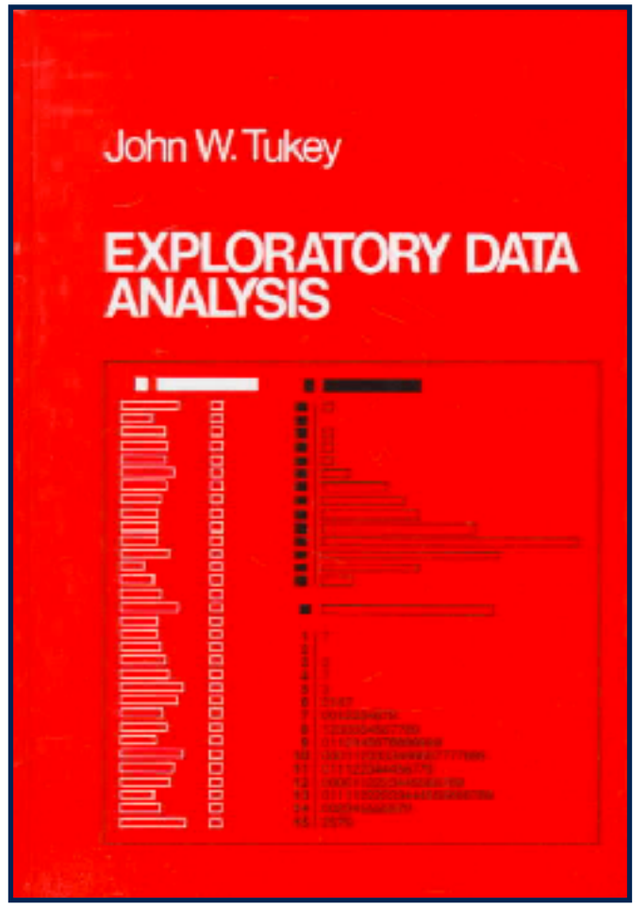


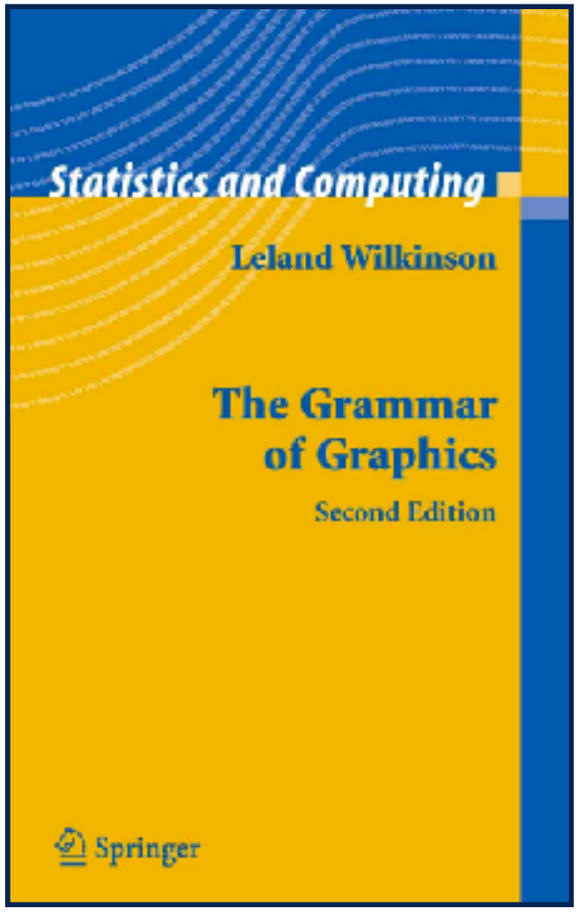


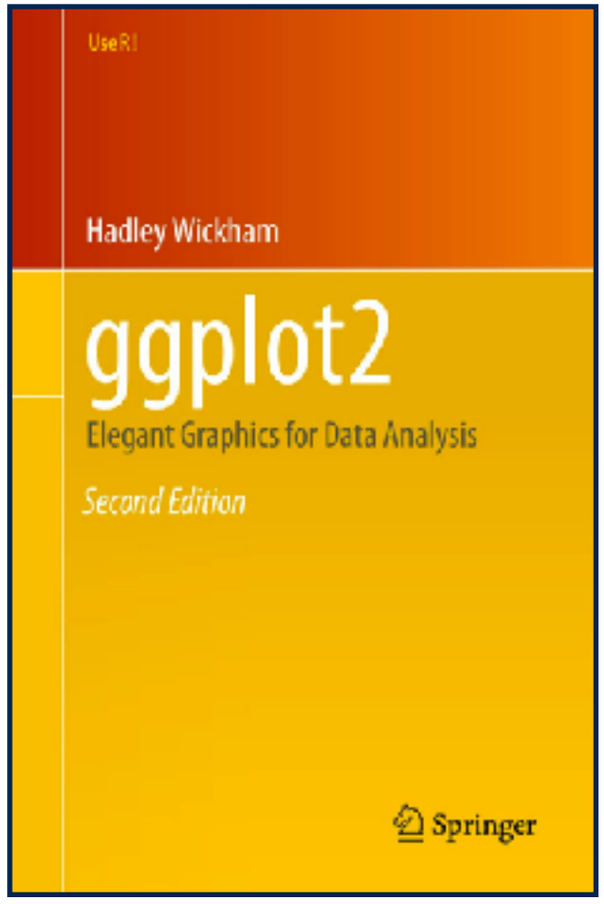
Sémiologie Graphique. Jacques Bertin (1967)

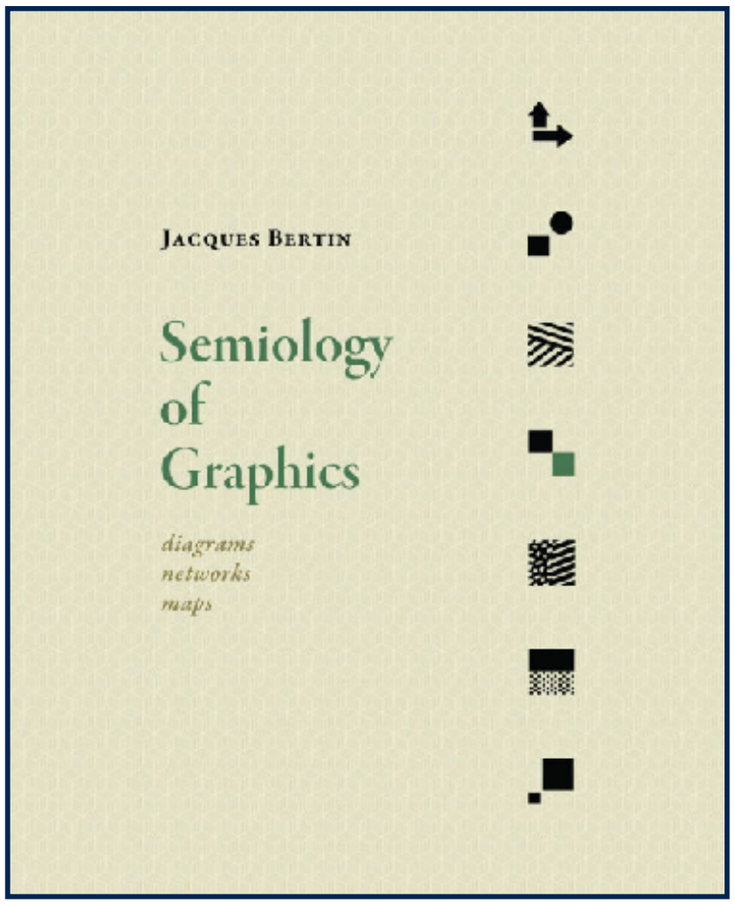
Translation: Semiology of Graphics (1983)

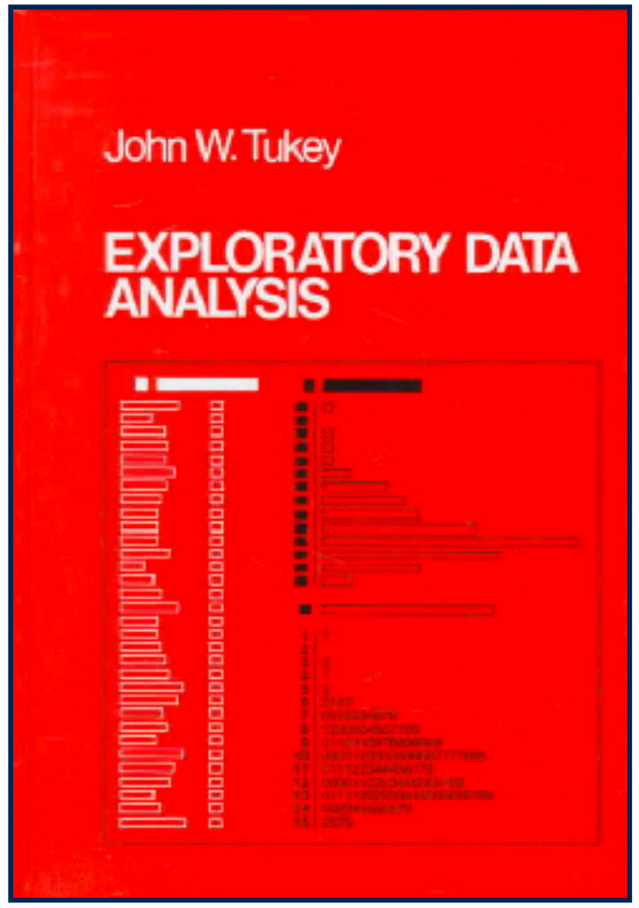

Jacques Bertin
Semiology of Graphics
diagrams
networks
maps

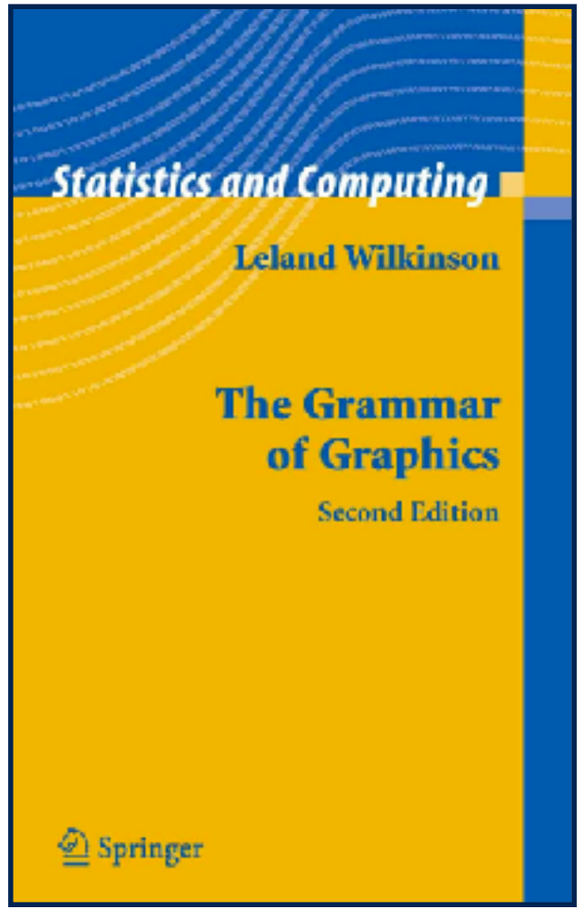

John W. Tukey
EXPLORATORY DATA ANALYSIS

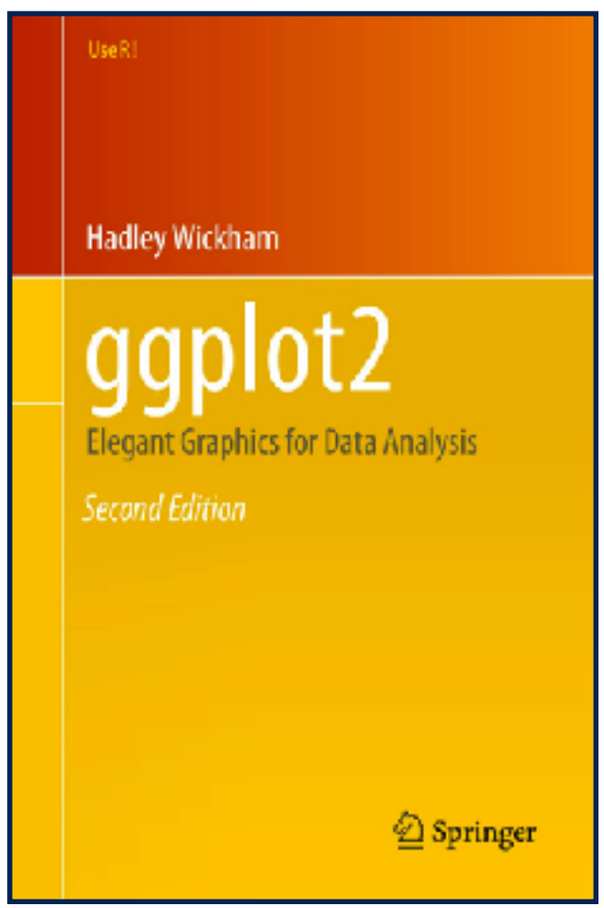

Statistics and Computing
Leland Wilkinson
The Grammar of Graphics
Second Edition
Springer


UseR!
Hadley Wickham
ggplot2
Elegant Graphics for Data Analysis
Second Edition
Springer

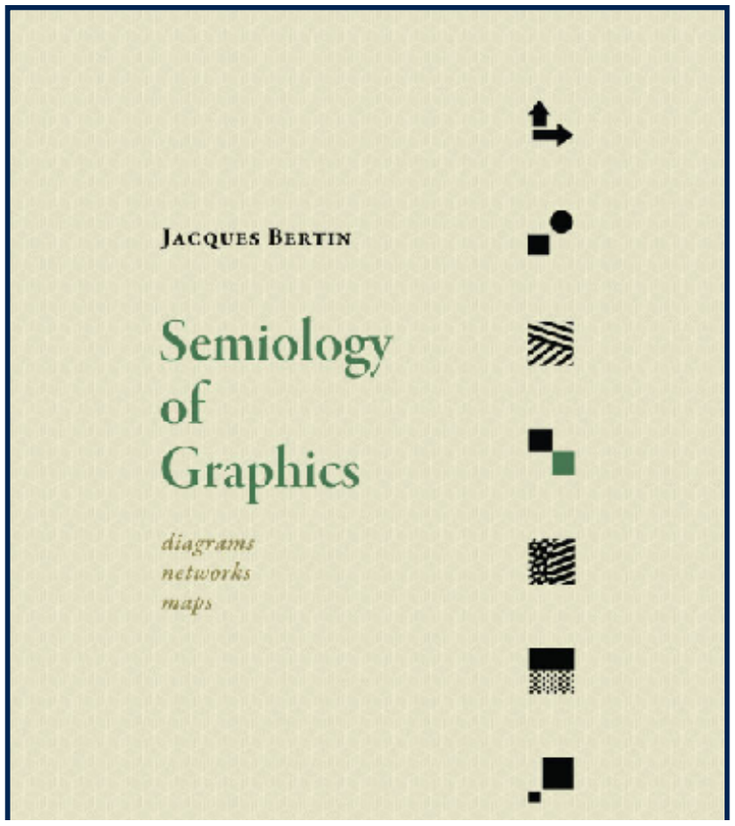

Jacques Bertin
Semiology of Graphics
diagrams
networks
maps

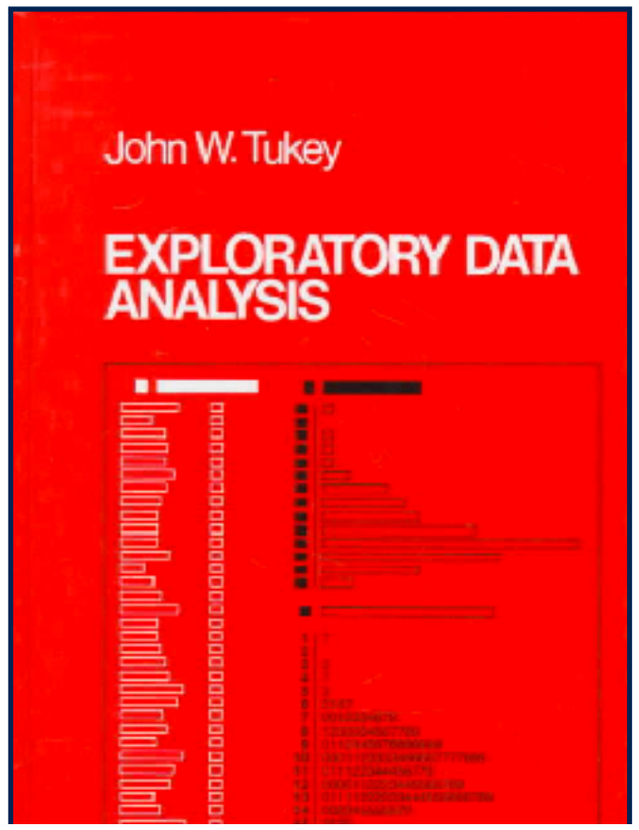

John W. Tukey
EXPLORATORY DATA ANALYSIS

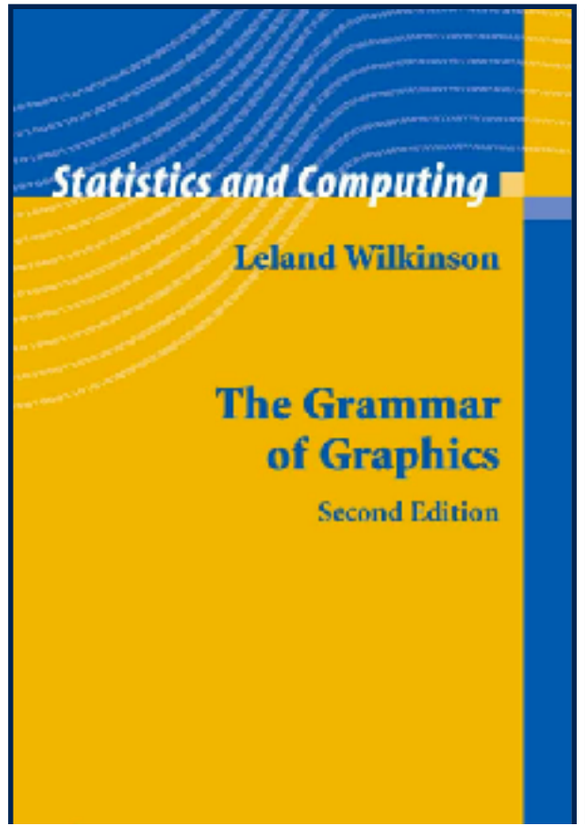

Statistics and Computing
Leland Wilkinson
The Grammar of Graphics
Second Edition
Springer

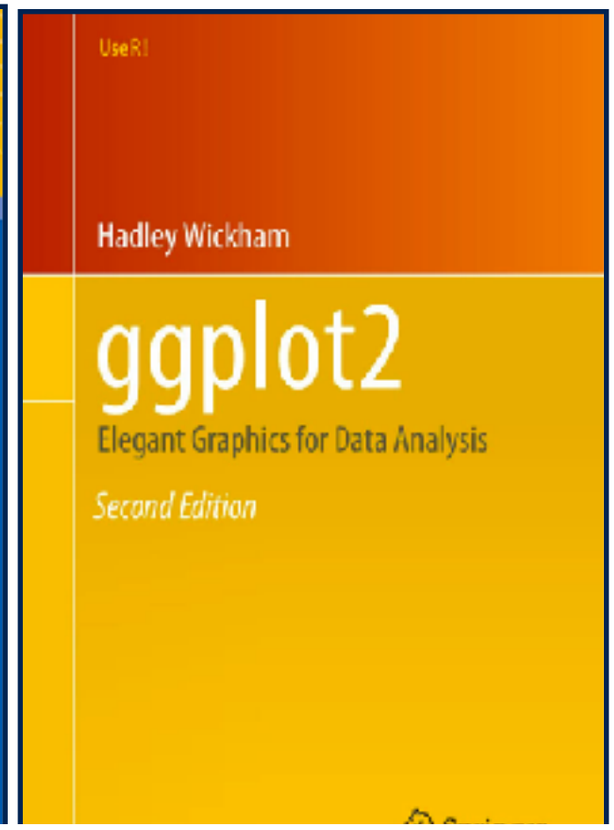

UseR!
Hadley Wickham
ggplot2
Elegant Graphics for Data Analysis
Second Edition
Springer


Przemysław Biecek
Odkrywać! Ujawniać! Objaśniać!
Zbiór esejów o sztuce prezentowania danych







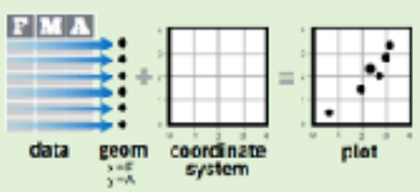


Data Visualization with ggplot2 Cheat Sheet

RStudio

Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same few components: a **data** set, a set of **geoms**—visual marks that represent data points, and a **coordinate system.**



To display data values, map variables in the data set to aesthetic properties of the geom like **size**, **color**, and **x** and **y** locations.

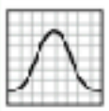
Geoms - Use a geom to represent data points, use the geom's aesthetic properties to represent variables. Each function re

One Variable

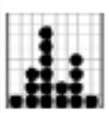
Continuous

```
a <- ggplot(mpg, aes(hwy))
```

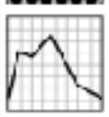
a + geom_area(stat = "bin")
x, y, alpha, color, fill, linetype, size
b + geom_area(aes(y = ..density..), stat = "bin")



a + geom_density(kernel = "gaussian")
x, y, alpha, color, fill, linetype, size, weight
b + geom_density(aes(y = ..county..))



a + geom_dotplot()
x, y, alpha, color, fill



a + geom_freqpoly()
x, y, alpha, color, linetype, size
b + geom_freqpoly(aes(y = ..density..))

a + geom_histogram(binwidth = 5)
x, y, alpha, color, fill, linetype, size, weight
b + geom_histogram(aes(y = ..density..))

Discrete

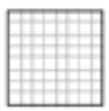
```
b <- ggplot(mpg, aes(fl))
```

b + geom_bar()
x, alpha, color, fill, linetype, size, weight

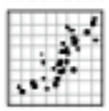
Two Variables

Continuous X, Continuous Y

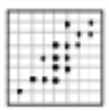
```
f <- ggplot(mpg, aes(cty, hwy))
```



f + geom_blank()



f + geom_jitter()
x, y, alpha, color, fill, shape, size

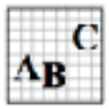


f + geom_point()
x, y, alpha, color, fill, shape, size

f + geom_quantile()
x, y, alpha, color, linetype, size, weight

f + geom_rug(sides = "bl")
alpha, color, linetype, size

f + geom_smooth(model = lm)
x, y, alpha, color, fill, linetype, size, weight



f + geom_text(aes(label = cty))
x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

Continuous Bivariate

```
i <- ggplot(movies, aes(y
```

i + geom_bin2d(binwid
xmax, xmin, ymax, ymin
linetype, size, weight

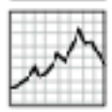
i + geom_density2d()
x, y, alpha, colour, linety

i + geom_hex()
x, y, alpha, colour, fill siz

Continuous Fur

```
j <- ggplot(economics, aes(d
```

j + geom_area()
x, y, alpha, color, fill, line



j + geom_line()
x, y, alpha, color, linetyp

j + geom_step(direction
x, y, alpha, color, linetyp

Visualizing er

```
df <- data.frame(grp = c("A", "B"
```

Do we need plots to understand predictive models?

1) …there is **a lot of opportunity to do visualization for machine learning**. Even many of the people working in the field don't have good intuitions for how their systems work, and they need tools to inspect what they're doing, debug, etc…

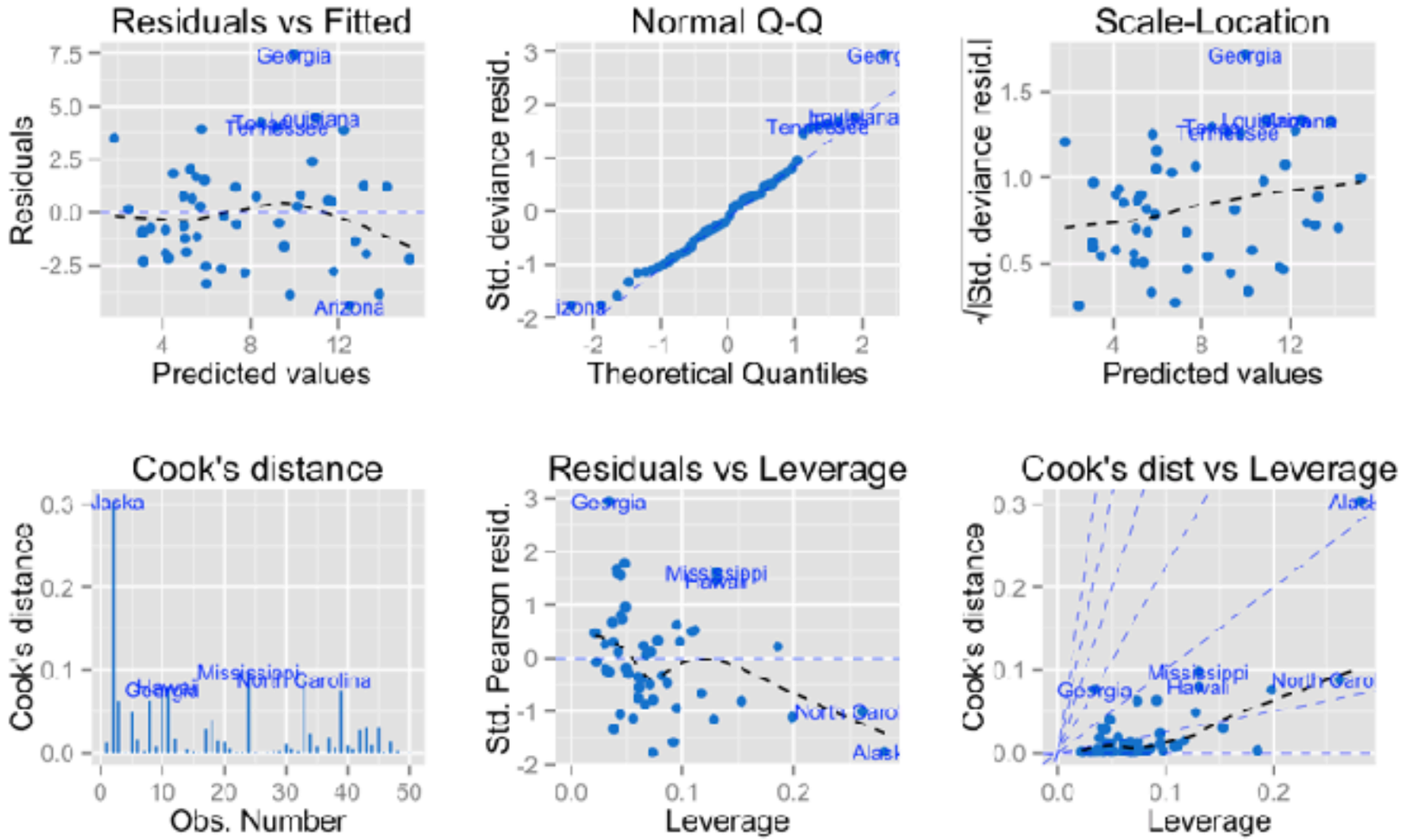
https://eagereyes.org/blog/2017/eurovis-2017-conference-report-part-1
EuroVis 2017 Conference
Robert Kosara
Senior Research Scientist at Tableau

2) Understanding and trust - we need to understand models that makes important decisions for us.
3) Machine learning in regulated industry

"Why Should I Trust You?": Explaining the Predictions of Any Classifier
Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin (2016)

Package ggfortify

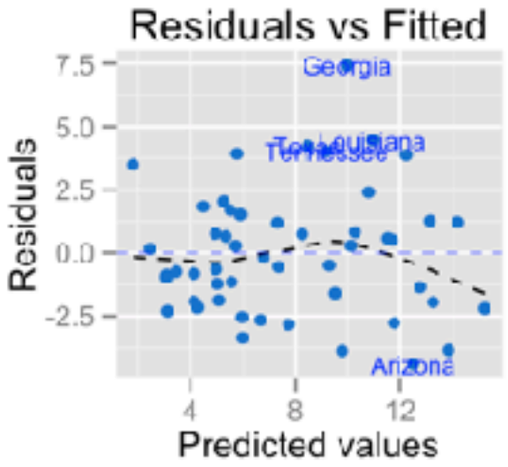
lm

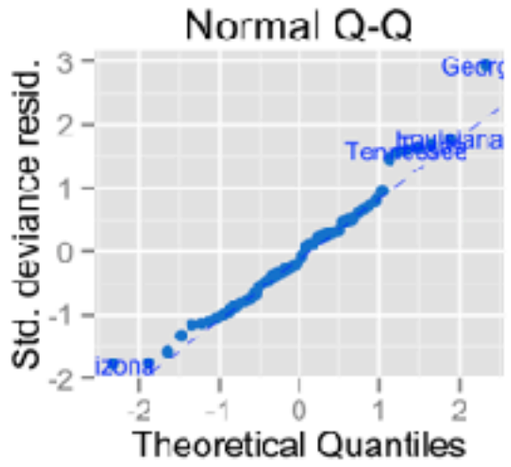


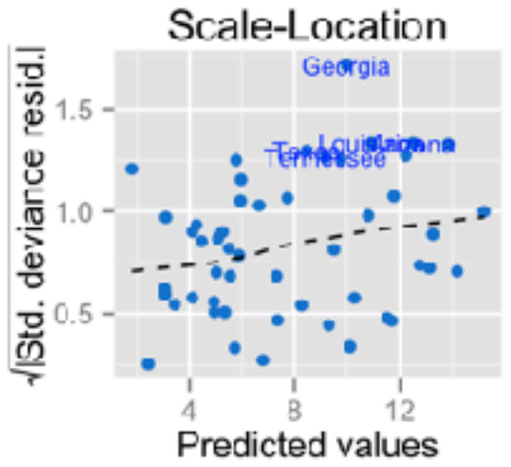
[Yuan Tang, Masaaki Horikoshi, and Wenxuan Li. "ggfortify: Unified Interface to Visualize Statistical Result of Popular R Packages." The R Journal 8.2 (2016): 478-489.]

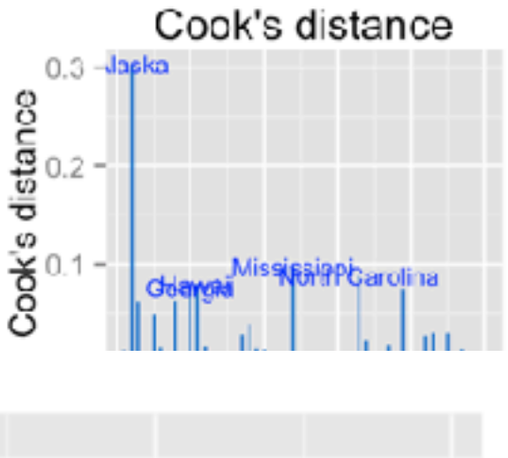
Package ggfortify

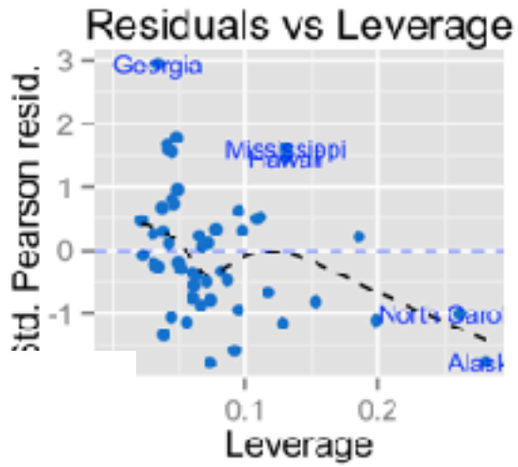
lm

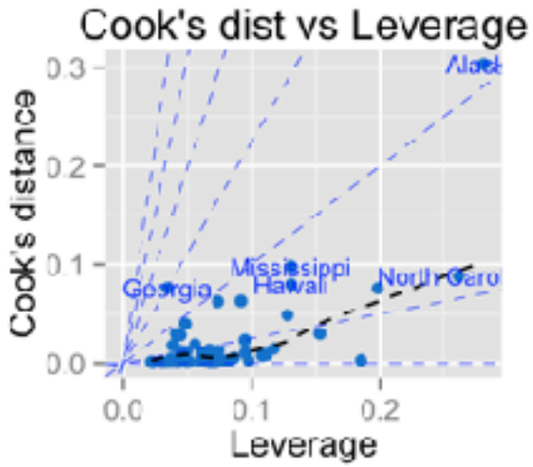




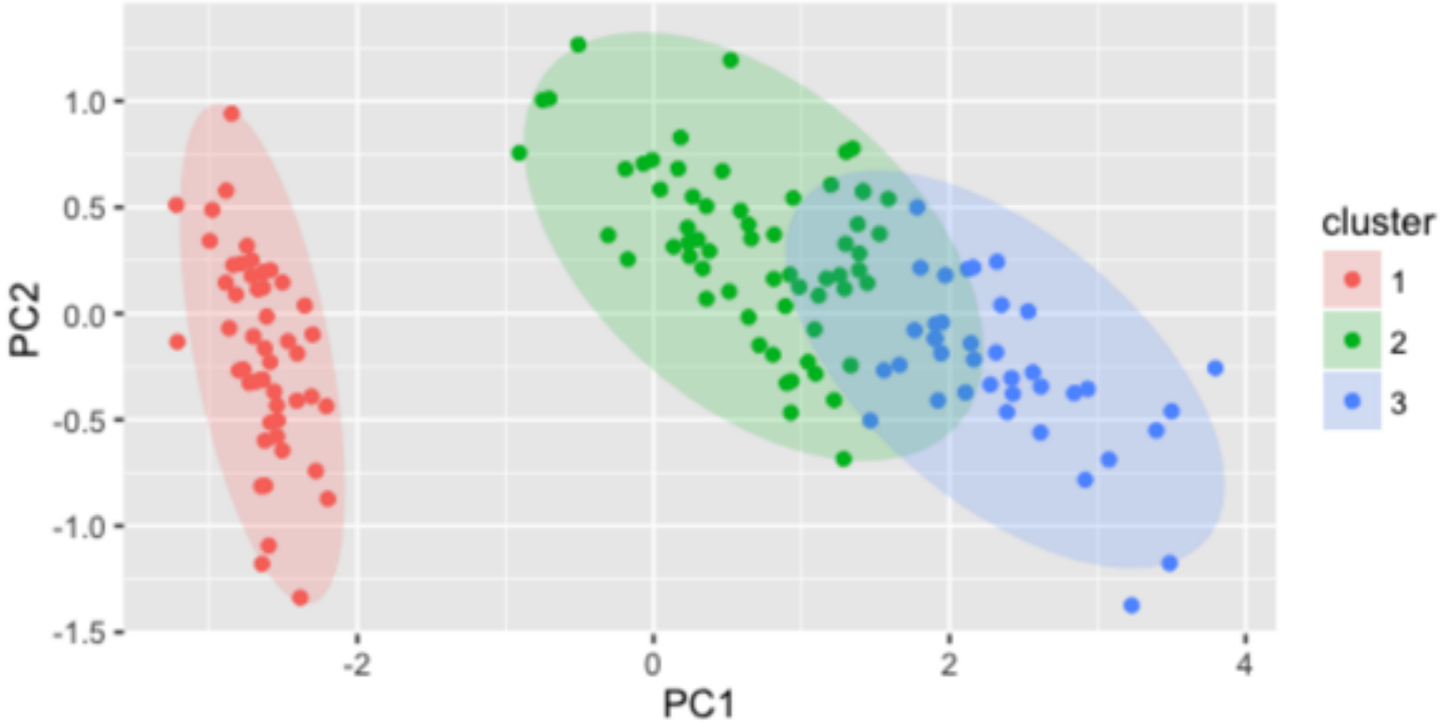






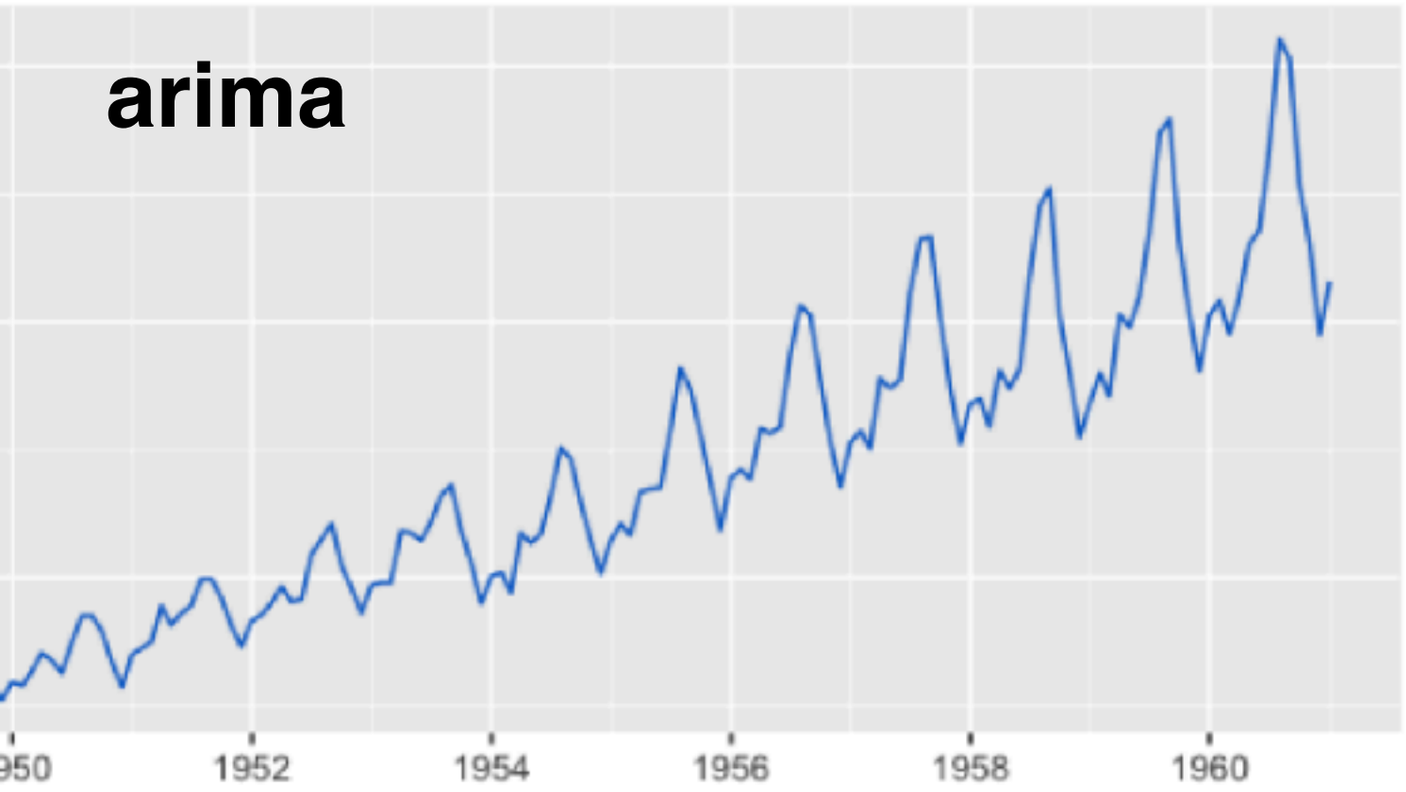


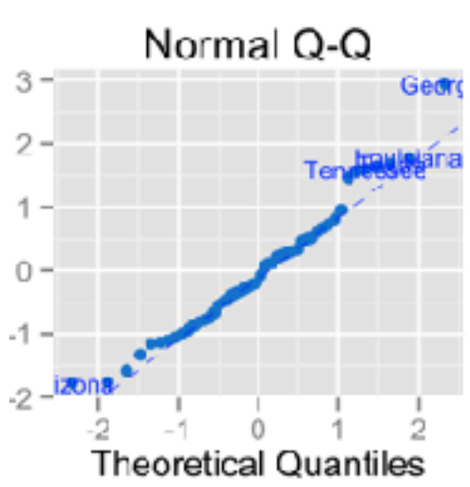
PAM

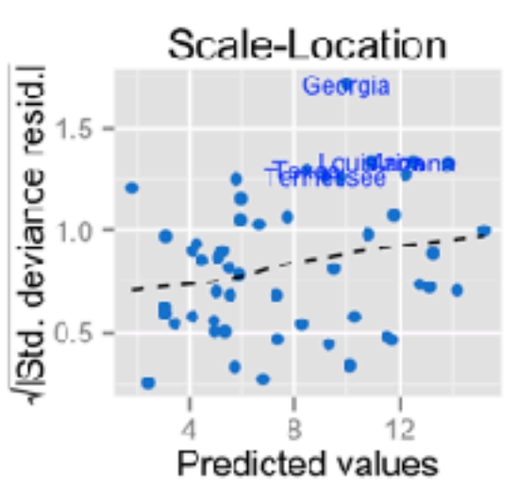


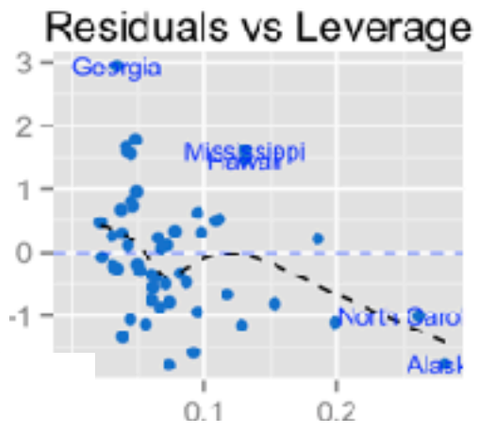
[Yuan Tang, Masaaki Horikoshi, and Wenxuan Li. "ggfortify: Unified Interface to Visualize Statistical Result of Popular R Packages." The R Journal 8.2 (2016): 478-489.]

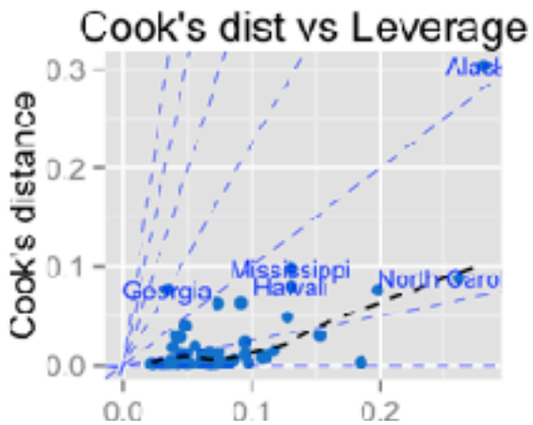
arima

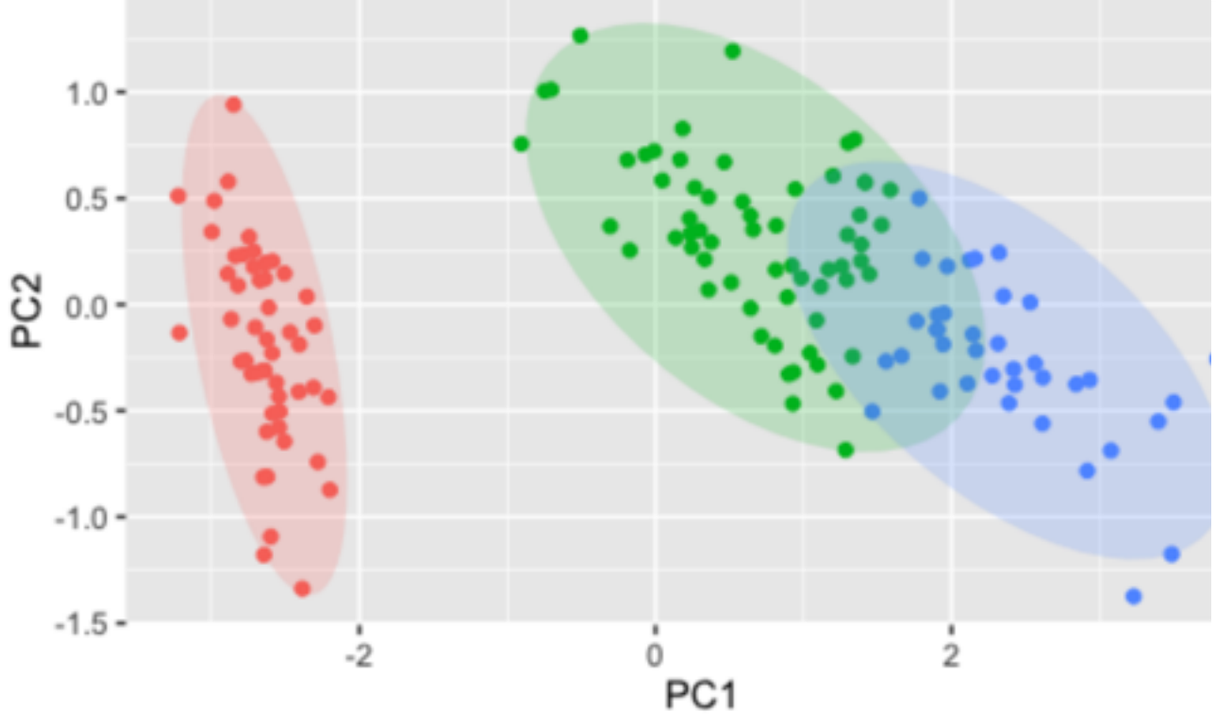






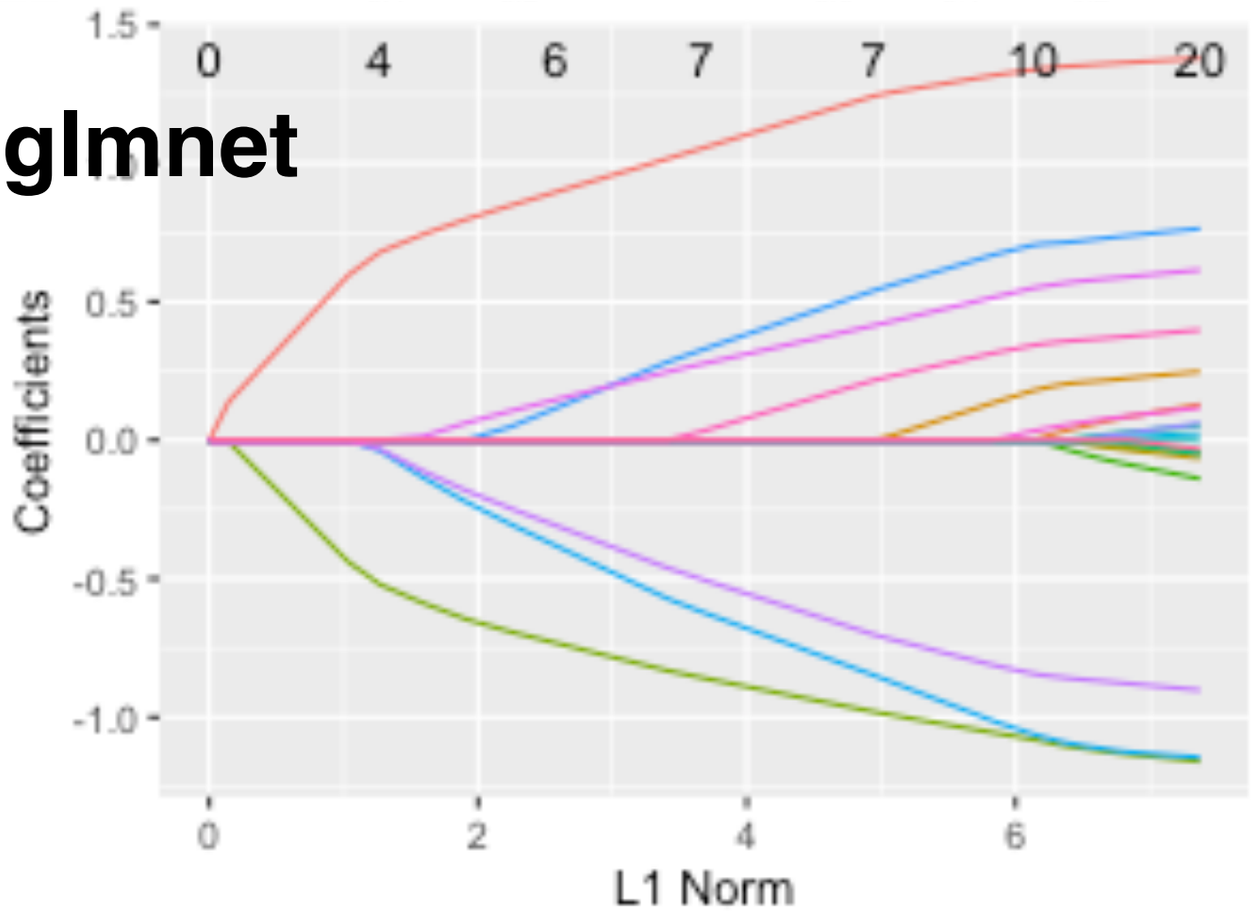


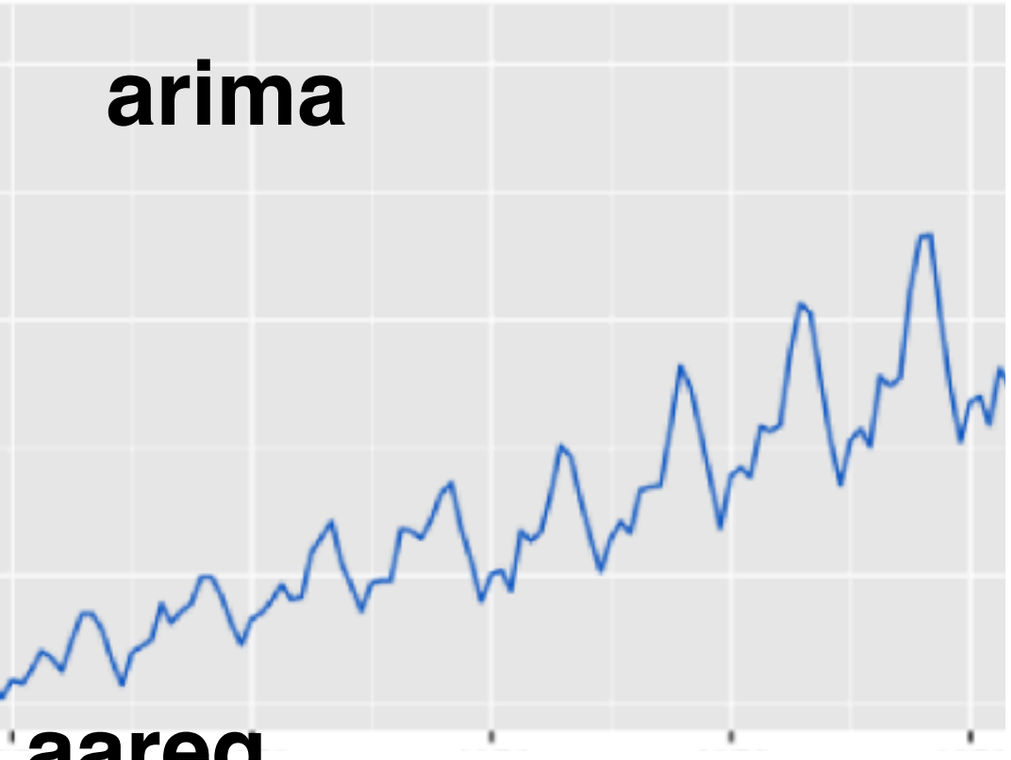


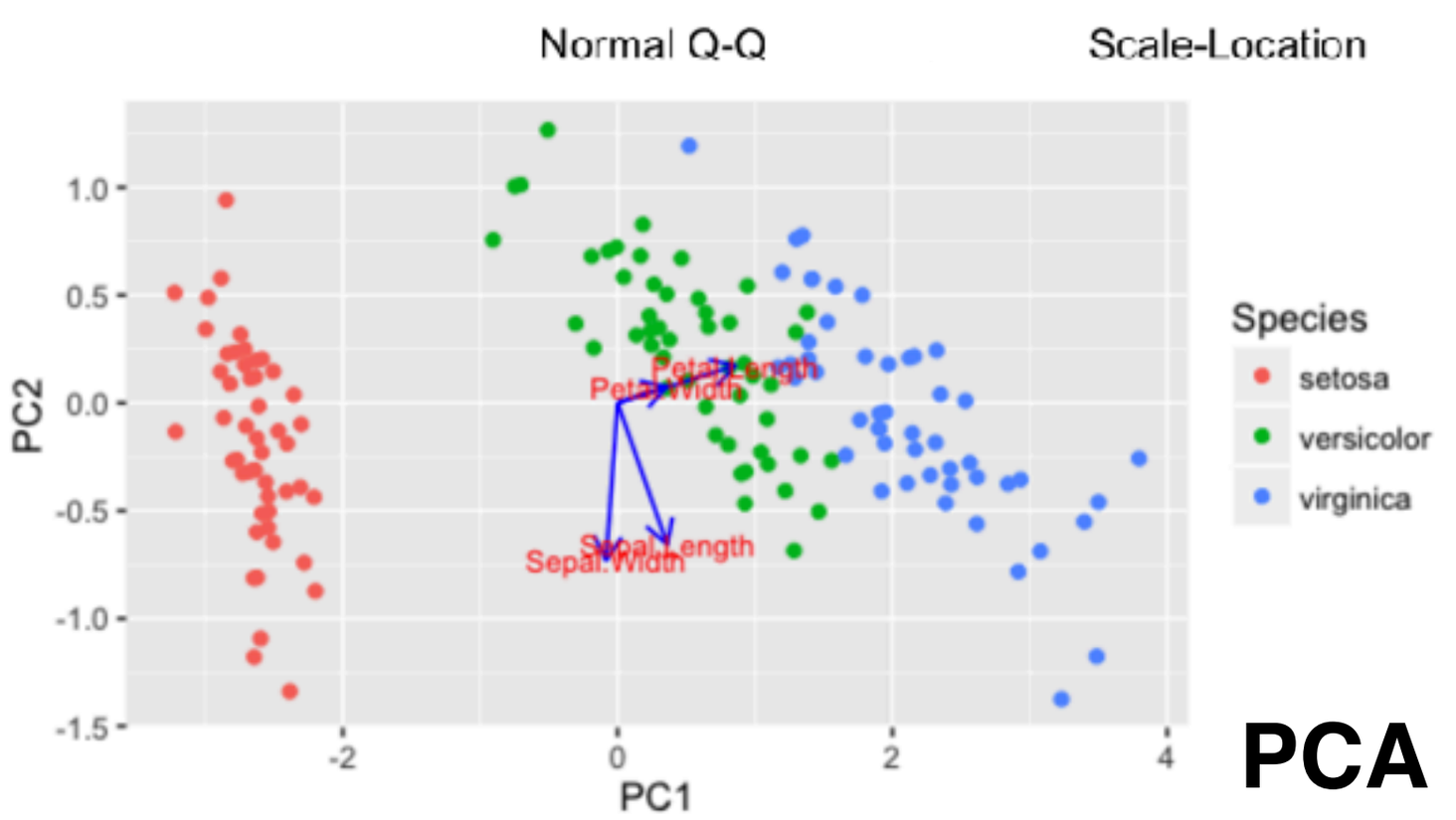


[Yuan Tang, Masaaki Horikoshi, and Wenxuan
Statistical Result of Popular R Packages." Th

glmnet



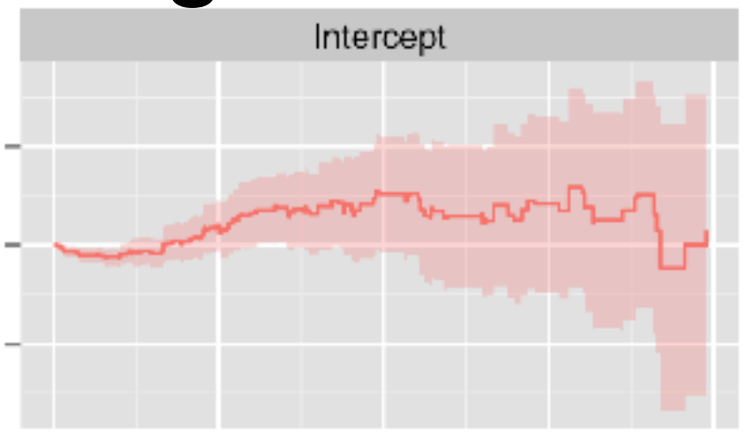


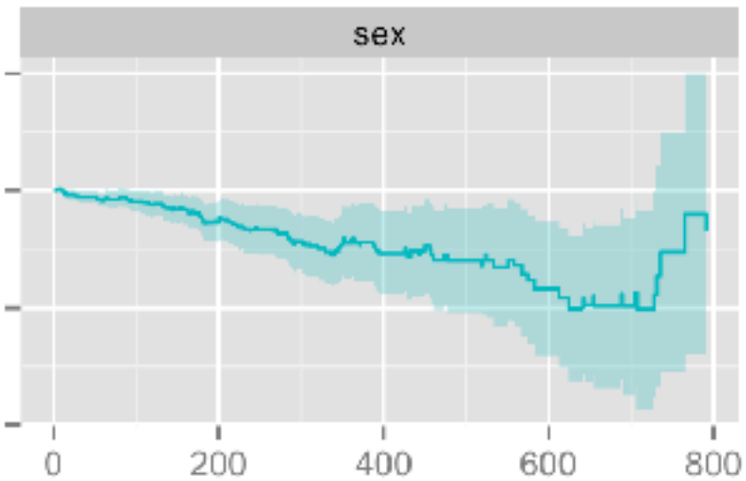


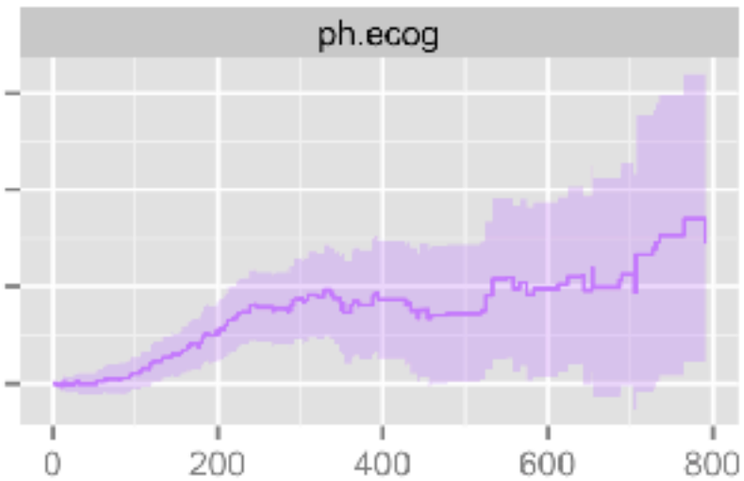
PCA

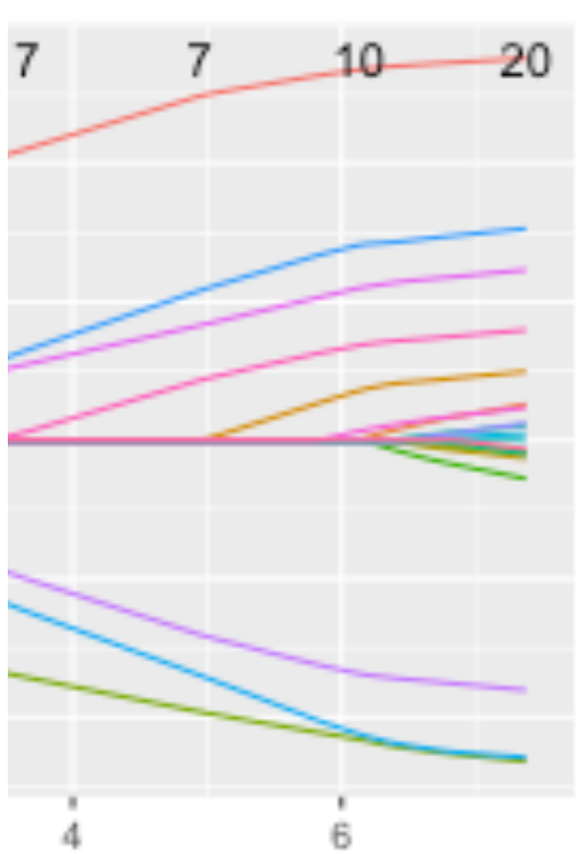
Figure 3: PCA with eigen-vectors and labels.

aareg









Package ggfortify

Table 1: Supported packages

package	supported types	package	supported types
base	"matrix", "table"	**sp**	"SpatialPoints", "SpatialPolygons", "Line", "Lines", "Polygon", "Polygons", "SpatialLines", "SpatialLinesDataFrame", "SpatialPointsDataFrame", "SpatialPolygonsDataFrame"
cluster	"clara", "fanny", "pam"	**stats**	"HoltWinters", "lm", "acf", "ar", "Arima", "stepfun", "stl", "ts", "cmdscale", "decomposed.ts", "density", "factanal", "glm", "kmeans", "princomp", "spec"
changepoint	"cpt"	**survival**	"survfit", "survvfit.cox"
dlm	"dlmFilter", "dlmSmooth"	**strucchange**	"breakpoints", "breakpointsfull"
fGarch	"fGARCH"	**timeSeries**	"timeSeries"
forecast	"bats", "forecast", "ets", "nnetar"	**tseries**	"irts"
fracdiff	"fracdiff"	**vars**	"varprd"
glmnet	"cv.glmnet", "glmnet"	**xts**	"xts"
KFAS	"KFS", "signal"	**zoo**	"zooreg"
lfda	"lfda", "klfda", "self"	**MASS**	"isoMDS", "sammon"
maps	"map"		

[Yuan Tang, Masaaki Horikoshi, and Wenxuan Li. "ggfortify: Unified Interface to Visualize Statistical Result of Popular R Packages." The R Journal 8.2 (2016): 478-489.]

Package ggfortify

- The main function: autoplot()
- Works for large collection for different statistical models
- Plots different things for different models
- Produces ggplot2 plots

Yuan Tang, Masaaki Horikoshi, and Wenxuan Li. "ggfortify: Unified Interface to Visualize Statistical Result of Popular R Packages." The R Journal 8.2 (2016): 478-489.

CA with FactoMineR and factoextra

Basics

CA (Correspondence Analysis) is a descriptive technique, that explains dependency between two categorical variable. It is usually applied to contingency tables — decomposes the chi-squared statistic into orthogonal factors, but may be applied to any table with nonnegative values.

It is conceptually similar to principal component analysis. The preprocessed contingency table is decomposed (Generalized Singular Value Decomposition) into a orthogonal profiles of rows and columns.

The Game of Thrones

This example uses the Kaggle dataset about battles in *Game of Thrones* (see http://bit.ly/2ojUfQ2). The **battles** table presents number of battles in which given House (row) was an attacking army with a given attacking strategy (column).

```
head(battles)
          Battle_Type
House       ambush pitched battle razing siege
  Baratheon      0              2      0     3
  Frey           1              0      0     2
  Lannister      1              6      0     2
  Stark          5              3      0     0
  Tully          3              0      0     0
```

Use the **FactoMineR ::CA()** function for correspondence analysis.

```
library("FactoMineR")
res.ca <- CA(battles, graph = FALSE)
summary(res.ca)

The chi square of independence between the two variables
is equal to 25.82556 (p-value =  0.183814 ).

Eigenvalues
                       Dim.1   Dim.2   Dim.3
Variance               0.302   0.201   0.099
% of var.             54.617  36.360  18.024
Cumulative % of var.  54.617  84.976 100.000

Rows
                Iner*1000     Dim.1     ctr    cos2
Baratheon     |   113.333 |   0.861  26.281  0.839
Bolton        |    45.200 |  -0.588   5.481  0.438
Frey          |    78.632 |   0.227   1.099  0.051
Greyjoy       |    83.333 |  -0.874   9.307  0.813
...

Columns
                Iner*1000     Dim.1     ctr    cos2
ambush        |   217.236 |  -0.747  59.386  0.989
pitched battle|   168.847 |   0.427  16.766  0.377
razing        |    99.359 |  -0.122   0.106  0.024
siege         |   184.781 |   0.578  23.742  0.485
...
```

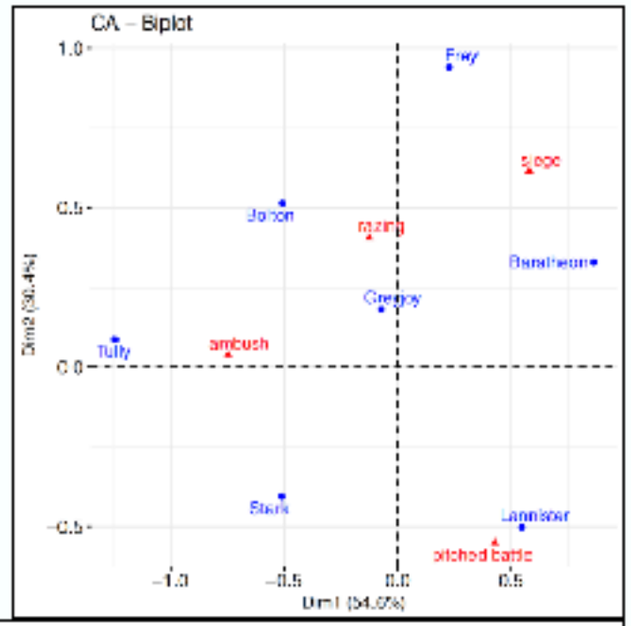
FactoMineR (for multivariate data analysis) and factoextra (for visualisation of CA results)

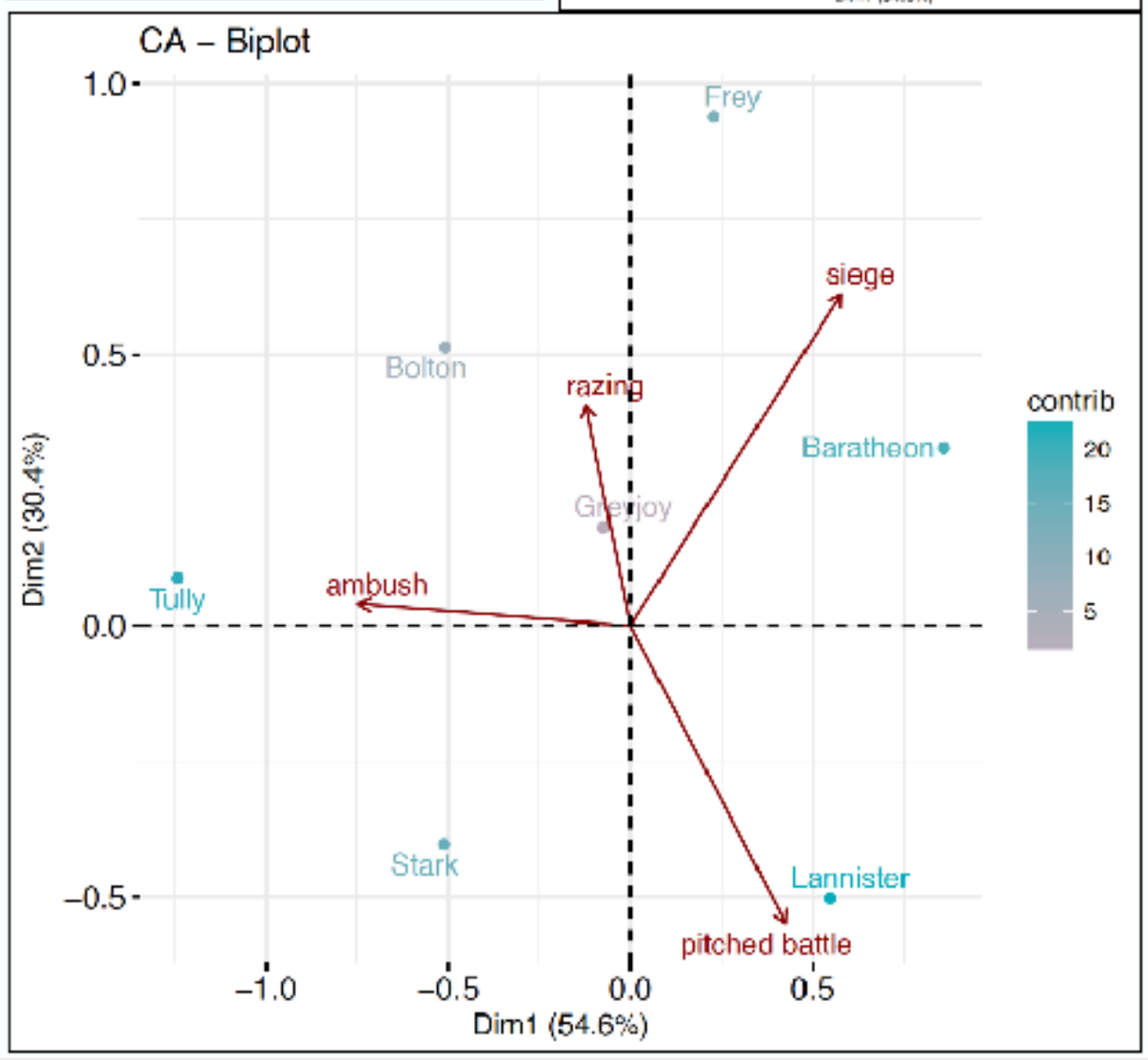
Biplots

Use the **factoextra::fviz_ca_biplot()** function to plot the biplot for a given model. Use **fviz_ca_row()** or **fviz_ca_col()** if you are interested only in profiles of rows/columns.

Consult the help page for numerous parameters of the **fviz_ca_biplot()** function, that allows to customise of biplot plots.

```
# Basic biplot, results in the right plot ->
fviz_ca_biplot(res.ca)
# Customised biplot, results in the bottom plot
fviz_ca_biplot(res.ca, repel = TRUE,
        arrow = c(FALSE, TRUE),
        col.row = "contrib",
        col.col = "red4",
        gradient.cols = c("#BBAFBB", "#00AFBB"))
```



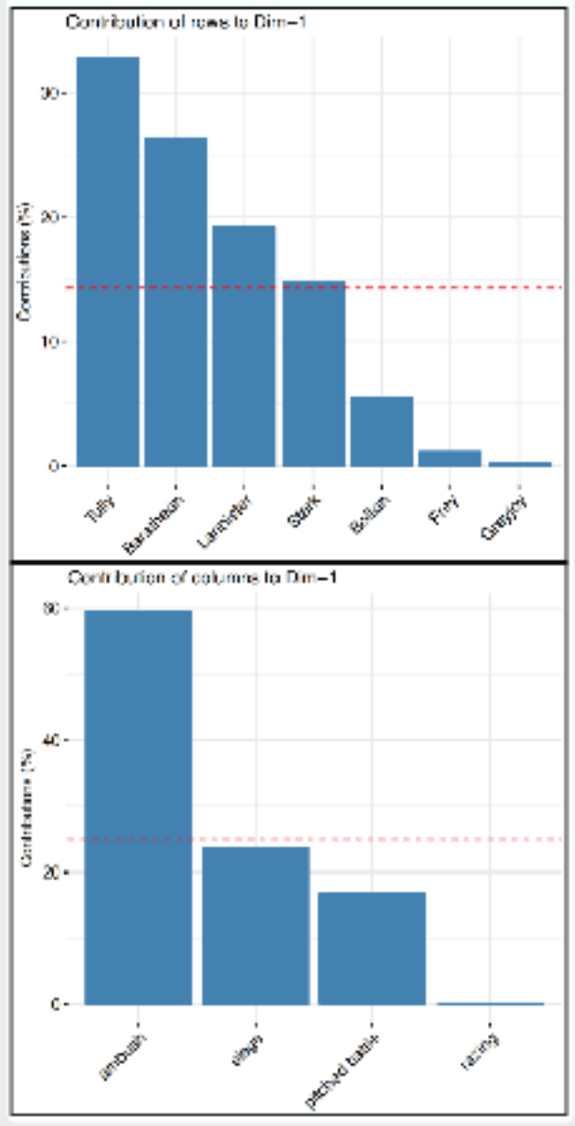


Row/Column contributions plot

Use the **factoextra::fviz_contrib()** function to plot contributions of selected dimension (rows or columns) onto a selected axis.

Numerical values for contributions may be extracted with **get_ca_row()** or **get_ca_col()** functions.

```
# Row contributions to Dimension 1
fviz_contrib(res.ca, choice ="row", axes = 1)
# Column contributions to Dimension 1
fviz_contrib(res.ca, choice ="col", axes = 1)
# Numeric summaries
get_ca_col(res.ca)
Correspondence Analysis - Results for columns
=================================================
 Name       Description
1 "$coord"   "Coordinates for the columns"
2 "$cos2"    "Cos2 for the columns"
3 "$contrib" "contributions of the columns"
4 "$inertia" "Inertia of the columns"
```



This cheatsheet presents functions from **FactoMineR** package (François Husson, Julie Josse, Sebastien Le, Jeremy Mazet, http://factominer.free.fr/) in version 1.35 and **factoextra** package (Alboukadel Kassambara, Fabian Mundt, http://www.sthda.com/english/rpkgs/factoextra/) in version 1.0.4

PCA with FactoMineR and factoextra

Basics

PCA (Principal Component Analysis) is a dimension-reduction method. It finds principal factors - orthogonal linear combinations of original variables that explain maximum amount of variance.

$$W_{n \times q} = X_{n \times p} A_{p \times q}$$

The p dimensional input data $\boldsymbol{X}$ is projected into a q dimensional subspace by a linear transformation defined by $\boldsymbol{R}$. New q dimensional data $\boldsymbol{W}$ has orthogonal variables. The transformation may be done through SVD decomposition or eigen value decomposition.

The Example

This example uses data about Hollywood action movies from 2015. Six quantitative variables with movie ratings scrapped from Rotten Tomato and Metacritic websites.

```
> head(movies2015)
                    Rotten                        Rotten Metacritic
                Tomatoes Metacritic Audience   Audience
Spectre               64         60       65         67
Furious 7             81         67       84         68
Terminator Genisys    25         38       59         63
San Andreas           50         43       56         55
Point Break            9         38       37         22
```

Use the **FactoMineR ::PCA()** function for PCA with supplementary quantitative and categorical variables. Missing values will be replaced by colMeans.

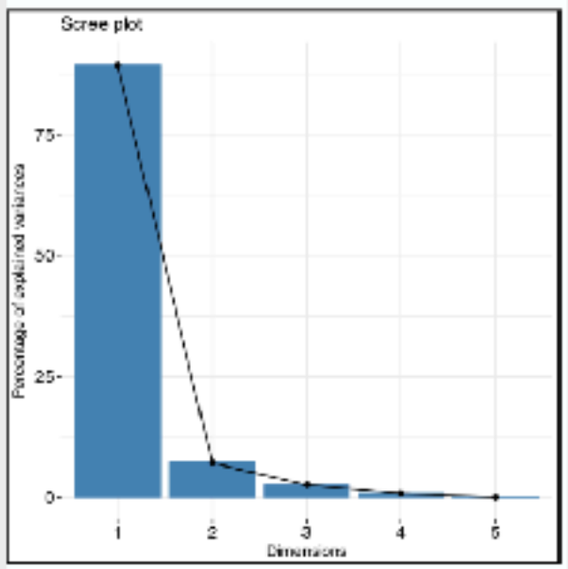
```
> library("FactoMineR")
> model <- PCA(movies2015)
> summary(model)
Eigenvalues
                       Dim.1   Dim.2   Dim.3   Dim.4   Dim.5
Variance               4.474   0.355   0.151   0.040   0.00
% of var.             89.481   7.093   2.027   0.798   0.00
Cumulative % of var.  89.481  96.574  98.202 100.000 100.00

Individuals
                        Dist    Dim.1    ctr    cos2
Spectre                | 1.077 |  0.980  2.184  0.842 |
Furious 7              | 2.408 |  2.321 12.865  0.930 |
Terminator Genisys     | 1.694 | -1.304  4.361  0.677 |
San Andreas            | 0.811 | -0.784  1.188  0.754 |
Point Break            | 3.643 | -3.451 20.767  0.902 |
Run All Night          | 1.192 |  0.842  1.584  0.490 |
No Escape              | 1.076 | -0.508  0.577  0.223 |
...

Variables
                      Dim.1    ctr   cos2    Dim.2
Rotten.Tomatoes     | 0.988 21.836  0.977 | -0.050
Metacritic          | 0.931 19.389  0.867 | -0.330
Average.critics     | 0.986 21.721  0.972 | -0.156
Rotten.Tomatoes.Audience | 0.943 19.885  0.890 |  0.135
Metacritic.Audience | 0.876 17.169  0.768 |  0.447
...
```

FactoMineR (for multivariate data analysis) and factoextra (for visualisation of PCA results)

Scree plot

Use the **factoextra::get_eig()** function to extract information about eigenvalues. The **factoextra::fviz_screeplot()** function will plot the percentage of variance explained by each principal factor.

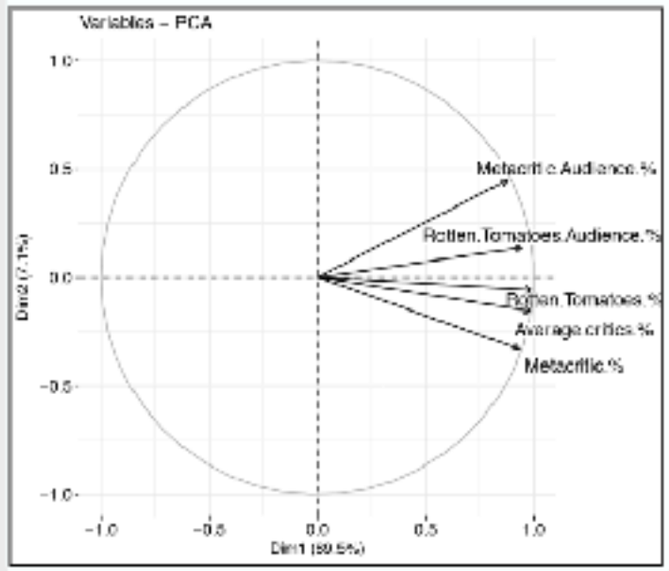
```
> get_eig(model)
       eigenvalue variance.percent cum.variance.percent
Dim.1 4.474839e+00       8.9488e-01                89.48
Dim.2 3.546786e-01       7.0936e+00                96.57
Dim.3 1.313722e-01       2.6273e+00                99.20
Dim.4 3.991824e-02       7.9836e-01               100.00
Dim.5 5.250294e-32       1.0501e-30               100.00
> fviz_screeplot(model)
```



PCA variables' plot

Use the **factoextra::fviz_pca_var()** function to plot contribution of original variables into selected (the **axes** argument) principal components . Show variables through text labels or arrows (the **geom** argument). Result of this function is the **ggplot2** plot.

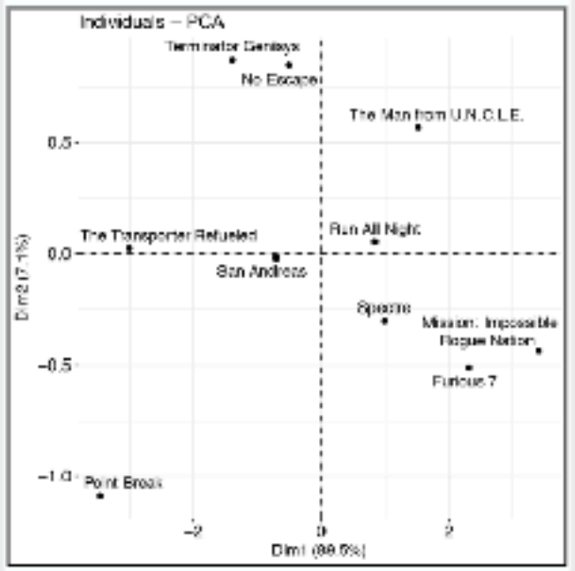
```
> fviz_pca_var(model)
```



PCA individuals' plot

Use the **factoextra::fviz_pca_ind()** function to plot observations with selected (the **axes** argument) principal coordinates. With the **habillage** argument one can select a grouping variable which will be color-coded in the plot. Use **addEllipses** to plot ellipses for each group.

```
> fviz_pca_ind(model)
```

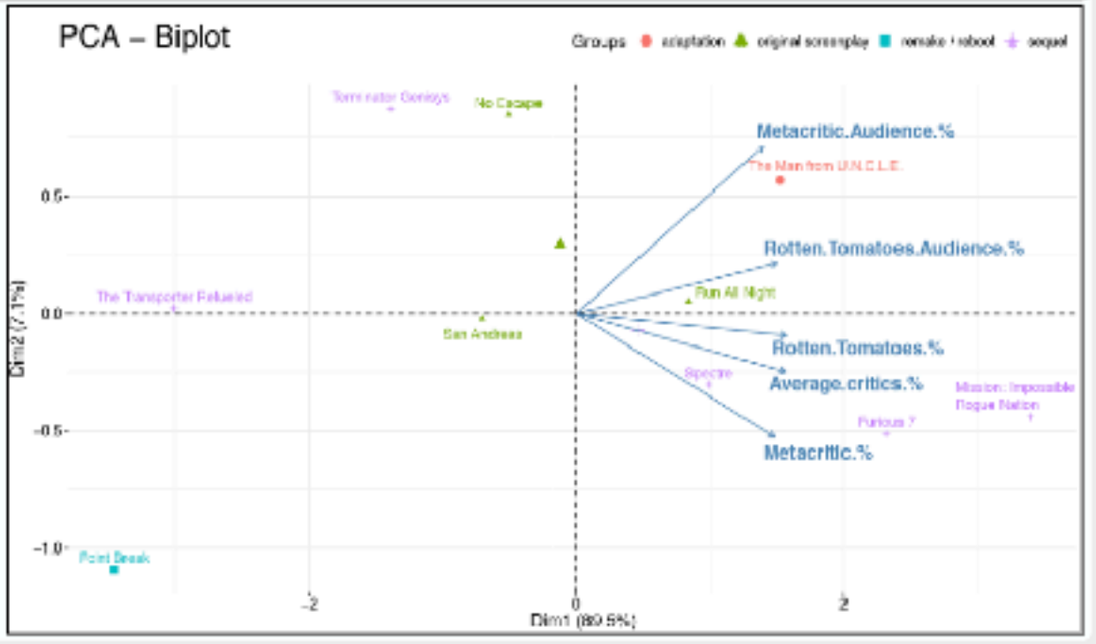


PCA - Biplot

Use the **factoextra::fviz_pca_biplot()** function to combine results for individuals and variables into a single bi-plot. With the **habillage** argument one can select a grouping variable which will be color-coded in the plot. Use **addEllipses** to plot ellipses for each group.

In the presented example, the first principal coordinate is highly correlated with average rating from all sources (audience and critics) while the second principal coordinate discriminate between audience and critics. Thus one can easily identify movies that are preferred by critics and these preferred by audience.

```
> fviz_pca_biplot(model, habillage = film2015$script.type) +
  theme(legend.position = "top")
```



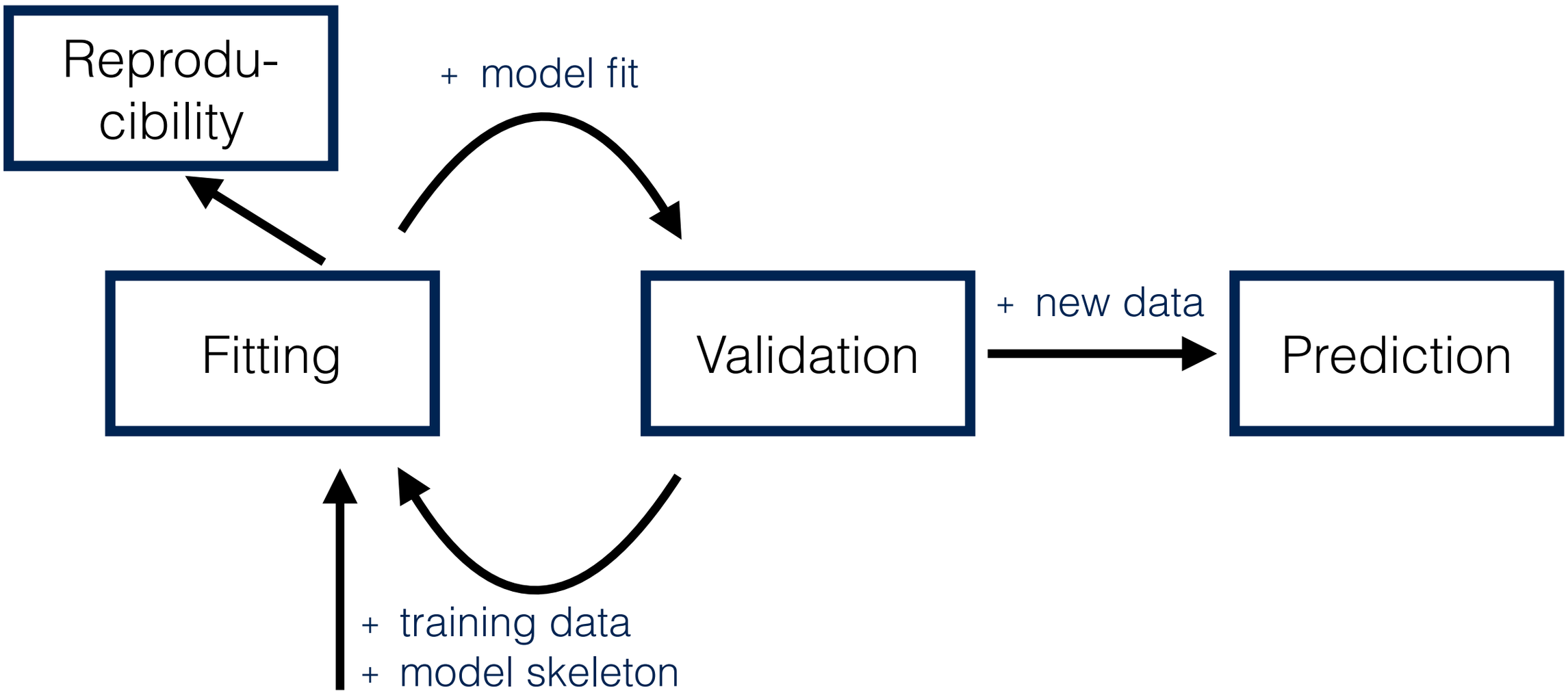
This cheatsheet presents functions from **FactoMineR** package (Francois Husson, Julie Josse, Sebastien Le, Jeremy Mazet, http://factominer.free.fr/) in version 1.35 and **factoextra** package (Alboukadel Kassambara, Fabian Mundt, http://www.sthda.com/english/rpkgs/factoextra/) in version 1.0.4

Package factoextra

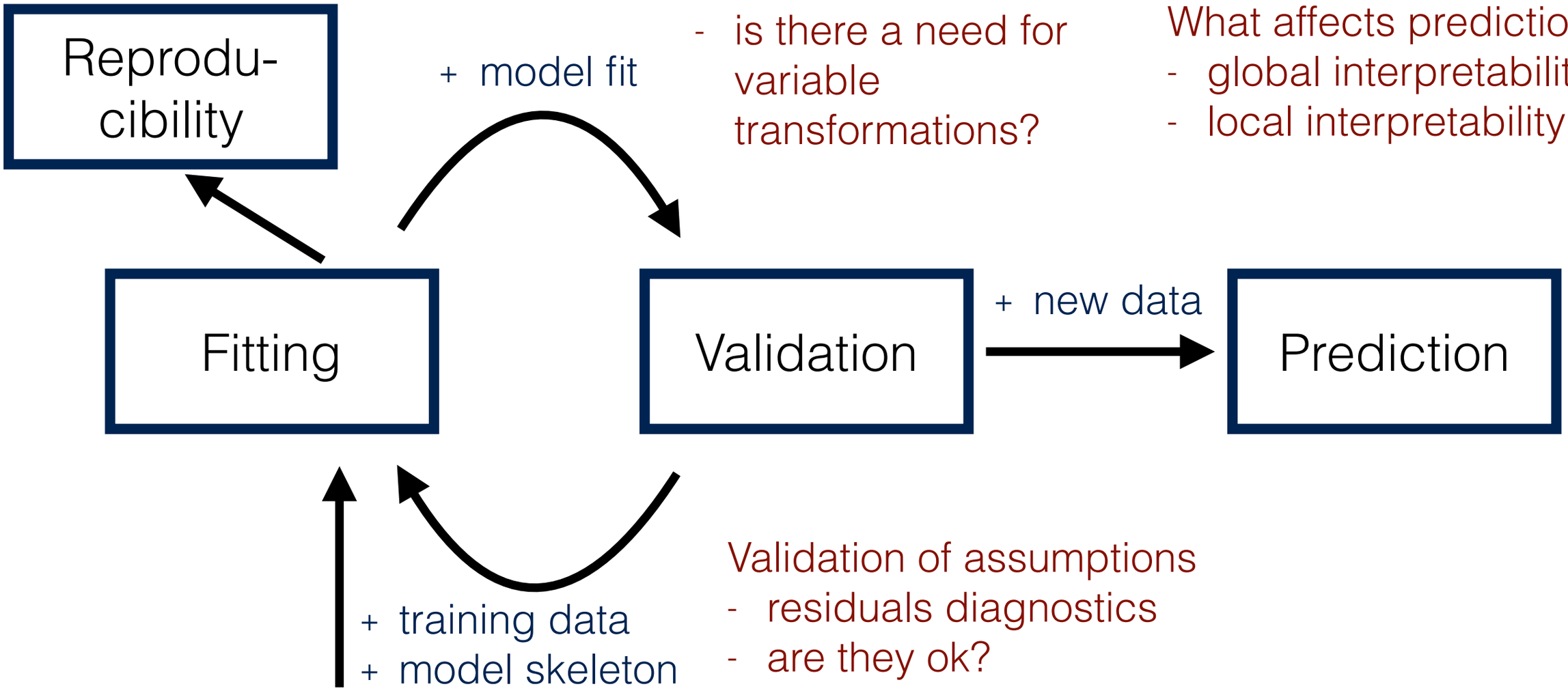
- Set of functions with many arguments
- Works for few selected models
- Plots many (all popular) graphical summaries for given models
- Produces ggplot2 plots

Alboukadel Kassambara, Fabian Mundt (2016) http://www.sthda.com/english/rpkgs/factoextra/
Sebastien Le, Julie Josse, Francois Husson (2008) FactoMineR: An R Package for Multivariate Analysis. Journal of Statistical Software, 25(1), 1-18. 10.18637/jss.v025.i01

Life-cycle of a typical prognostic model



Life-cycle of a typical prognostic model



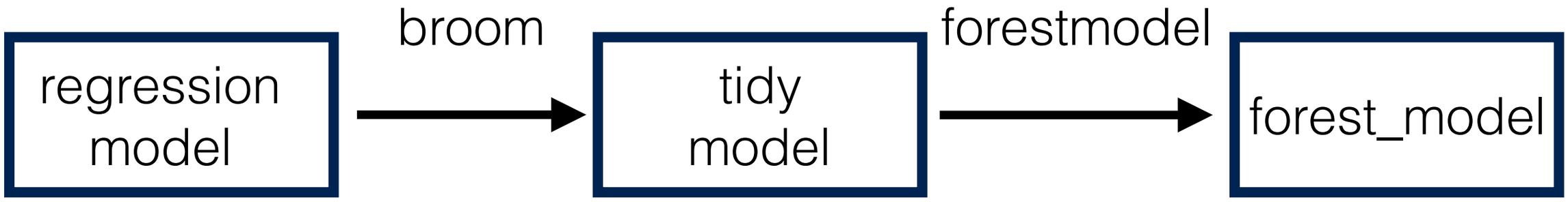
Fitting

Package forestmodel

Variable		N	Hazard ratio	p
Age		180		1.01 (0.99, 1.03) 0.467
Sex	Male	113		Reference
	Female	67		0.60 (0.41, 0.88) 0.008
ECOG	0	50		Reference
	1	86		1.32 (0.85, 2.04) 0.215
	2	43		2.19 (1.33, 3.61) 0.002
	3	1		6.00 (0.79, 45.70) 0.083
Meal.Cal		180		1.00 (1.00, 1.00) 0.793

Nick Kennedy (2017) https://github.com/NikNakk/forestmodel

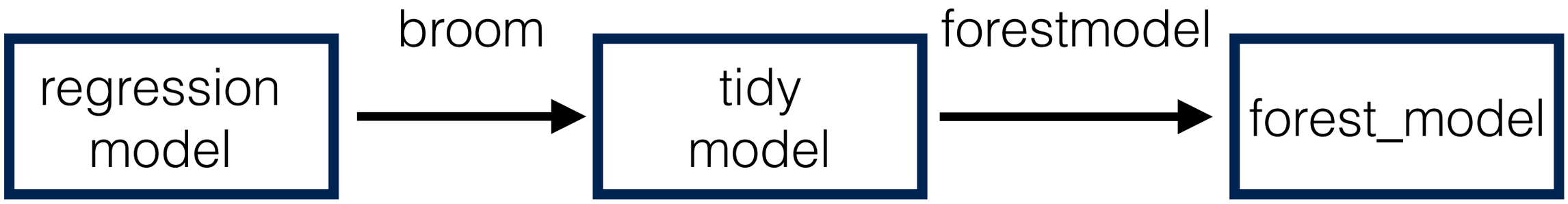
Package forestmodel



Variable		N	Hazard ratio	p
Age		180		1.01 (0.99, 1.03) 0.467
Sex	Male	113		Reference
	Female	67		0.60 (0.41, 0.88) 0.008
ECOG	0	50		Reference
	1	86		1.32 (0.85, 2.04) 0.215
	2	43		2.19 (1.33, 3.61) 0.002
	3	1		6.00 (0.79, 45.70) 0.083

Nick Kennedy (2017) https://github.com/NikNakk/forestmodel

Package forestmodel



Class	tidy	glance
aareg	x	x
acf	x	
anova	x	
aov	x	
aovlist	x	
Arima	x	x
betareg	x	x
biglm	x	x

spec	x		
stanfit	x		
stanreg	x	x	
summary.glht	x		
summary.lm	x	x	
summaryDefault	x	x	
survexp	x	x	
survfit	x	x	
survreg	x	x	x

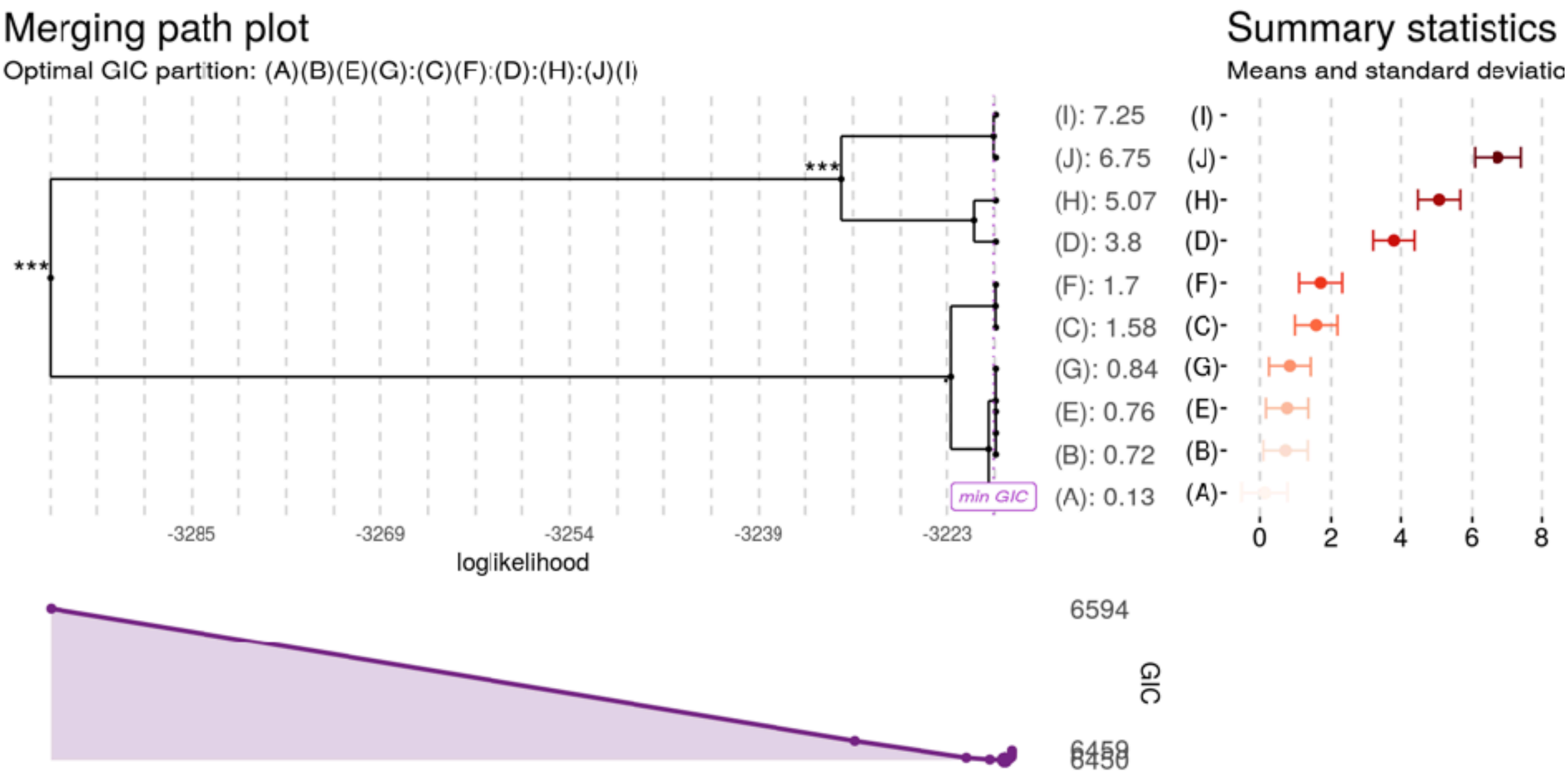
broom: An R Package for Converting Statistical Analysis Objects Into Tidy Data Frames, David Robinson (2014) arXiv:1412.3565v2 [stat.CO]

Package forestmodel

- The main function: forest_model()
- Works for many regression models (thanks to the broom package)
- Plot a single summary - forest models, rich in information about model parameters
- Produces ggplot2 plots

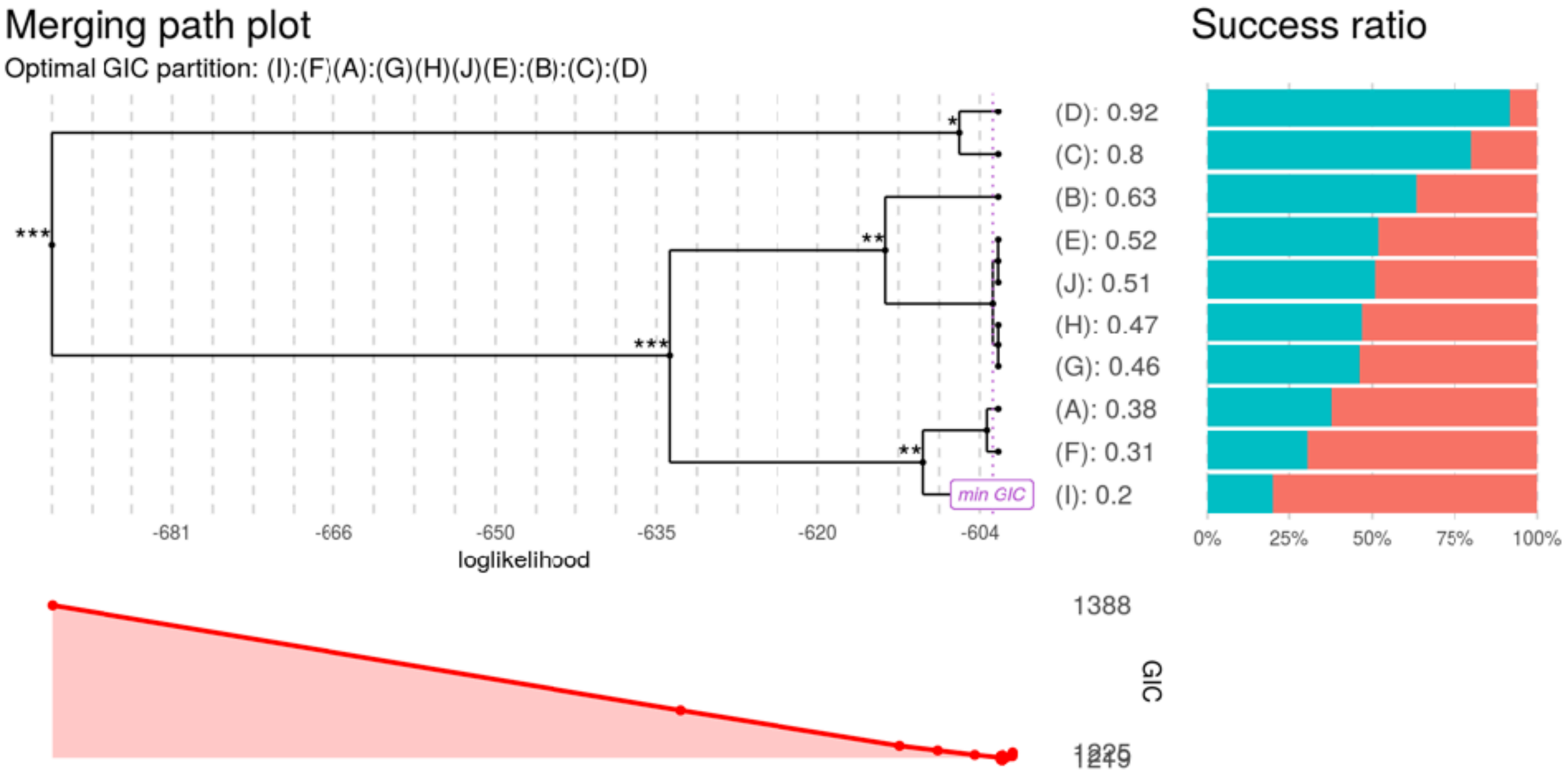
Nick Kennedy (2017) https://github.com/NikNakk/forestmodel

Package factorMerger
Visualisation for post-hoc testing



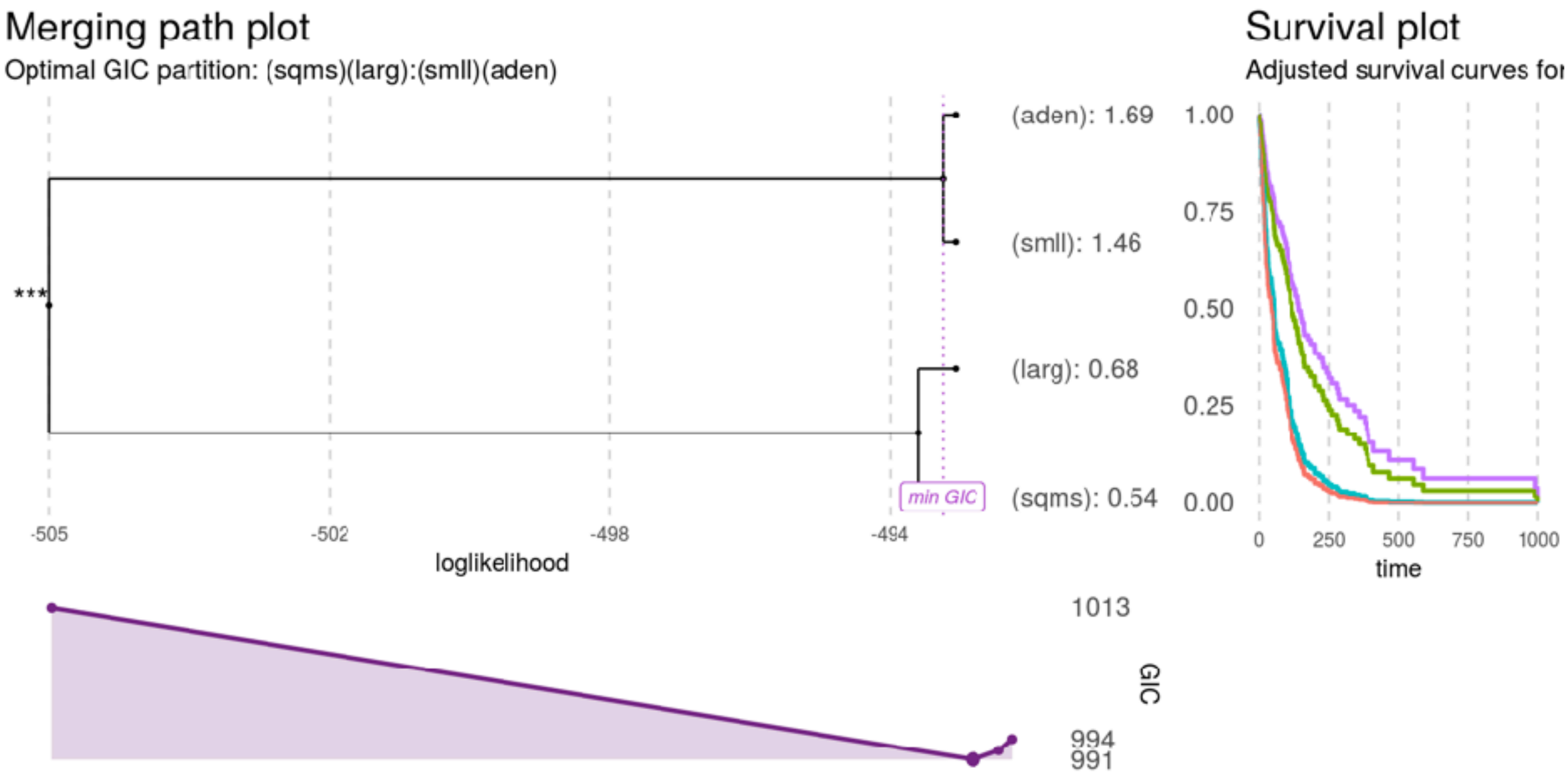
Agnieszka Sitko, Przemysław Biecek (2017)
https://github.com/geneticsMiNIng/FactorMerger

Package factorMerger
Visualisation for post-hoc testing



Agnieszka Sitko, Przemysław Biecek (2017)
https://github.com/geneticsMiNIng/FactorMerger

Package factorMerger
Visualisation for post-hoc testing



Agnieszka Sitko, Przemysław Biecek (2017)
https://github.com/geneticsMiNIng/FactorMerger

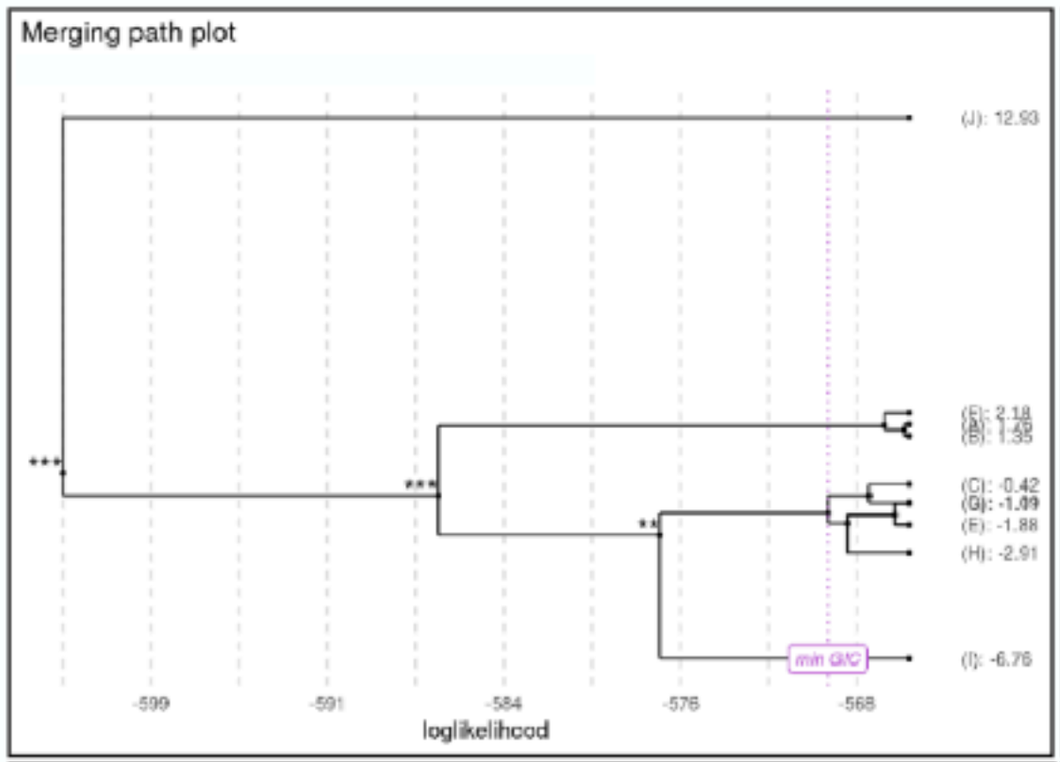
FactorMerger - set of tools to support results from post hoc testing

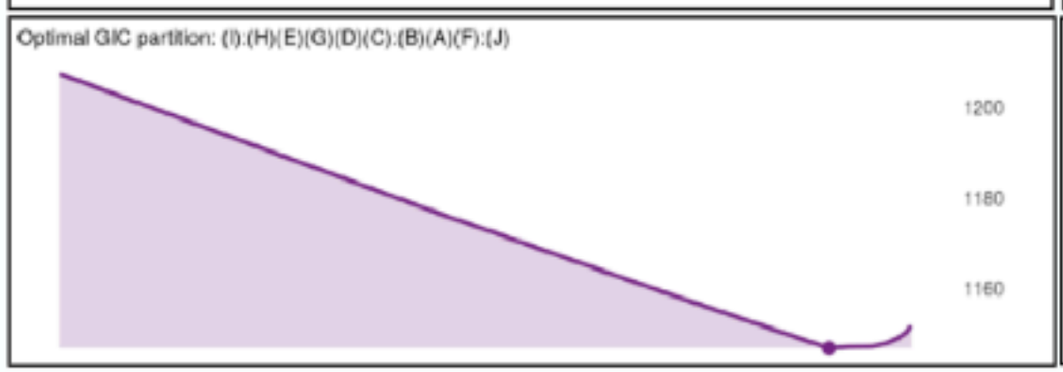
Level fusing plot

The top-left plot shows level fusing paths (merging paths). With arguments **family=**, **show=**, **fuse=**, **spacing=**, one can select how to merge factors and what shall be presented on OX/OY axes.

Argument	Summary
`panel = "all"`	All panels
`panel = "left"`	Only left two panels
`panel = "top"`	Only top two panels
`panel = "merging"`	Merging path plot

Argument	Summary
`show = "likelihood"`	Plot likelihood on OX axis
`show = "p-value"`	Plot p-values on OX axis
`fuse = "all2all"`	Compare all pairs of groups
`fuse = "nearby"`	Compare nearby groups
`fuse = "cluster"`	DMR4glm algorithm
`spacing = "equidistant"`	Levels equidistant on OY scale
`spacing = "effects"`	Levels according to their effects
`family = "gaussian"`	For one-dimensional Gaussian
`family = "mgaussian"`	For multi dimensional Gaussian
`family = "binomial"`	For binomial regression
`family = "survival"`	For Cox regression





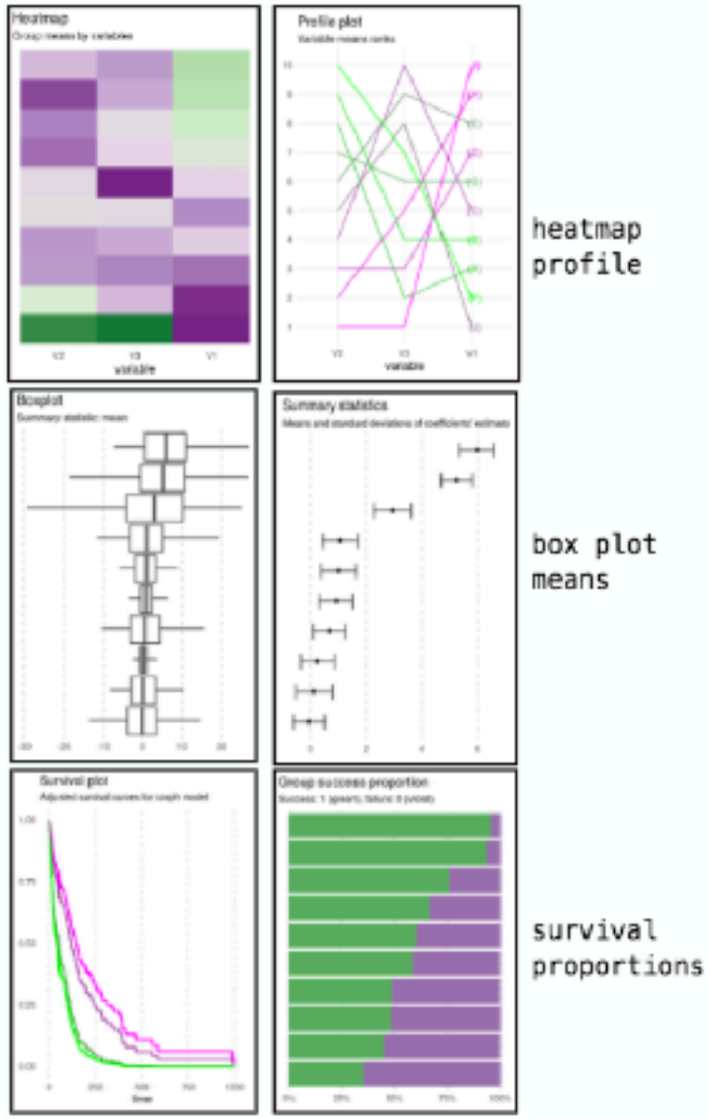
Model statistics

Group summaries

The top-right panel shows group characteristics.

Use the parameter **summary=** to select the most suitable presentation.

Argument	Summary
`summary = "heatmap"`	For mgaussian
`summary = "profile"`	For mgaussian
`summary = "boxplot"`	For gaussian
`summary = "means"`	For gaussian
`summary = "survival"`	For Cox regression
`summary = "proportions"`	For binomial regression



`heatmap`
`profile`

`box plot`
`means`

`survival`
`proportions`

Agnieszka Sitko, Przemysław Biecek (2017)

https://github.com/geneticsMiNIng/FactorMerger

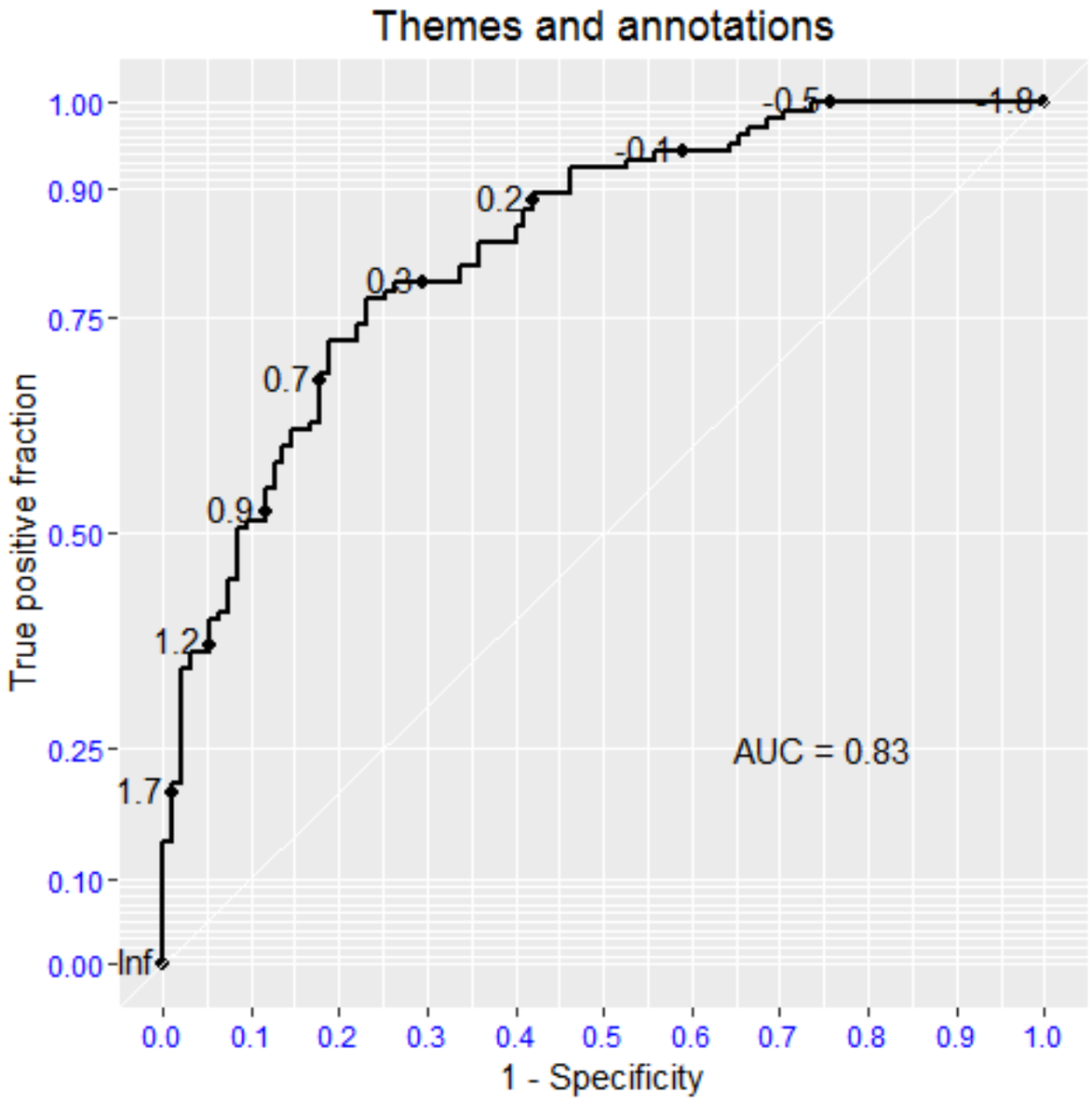
Package factorMerger

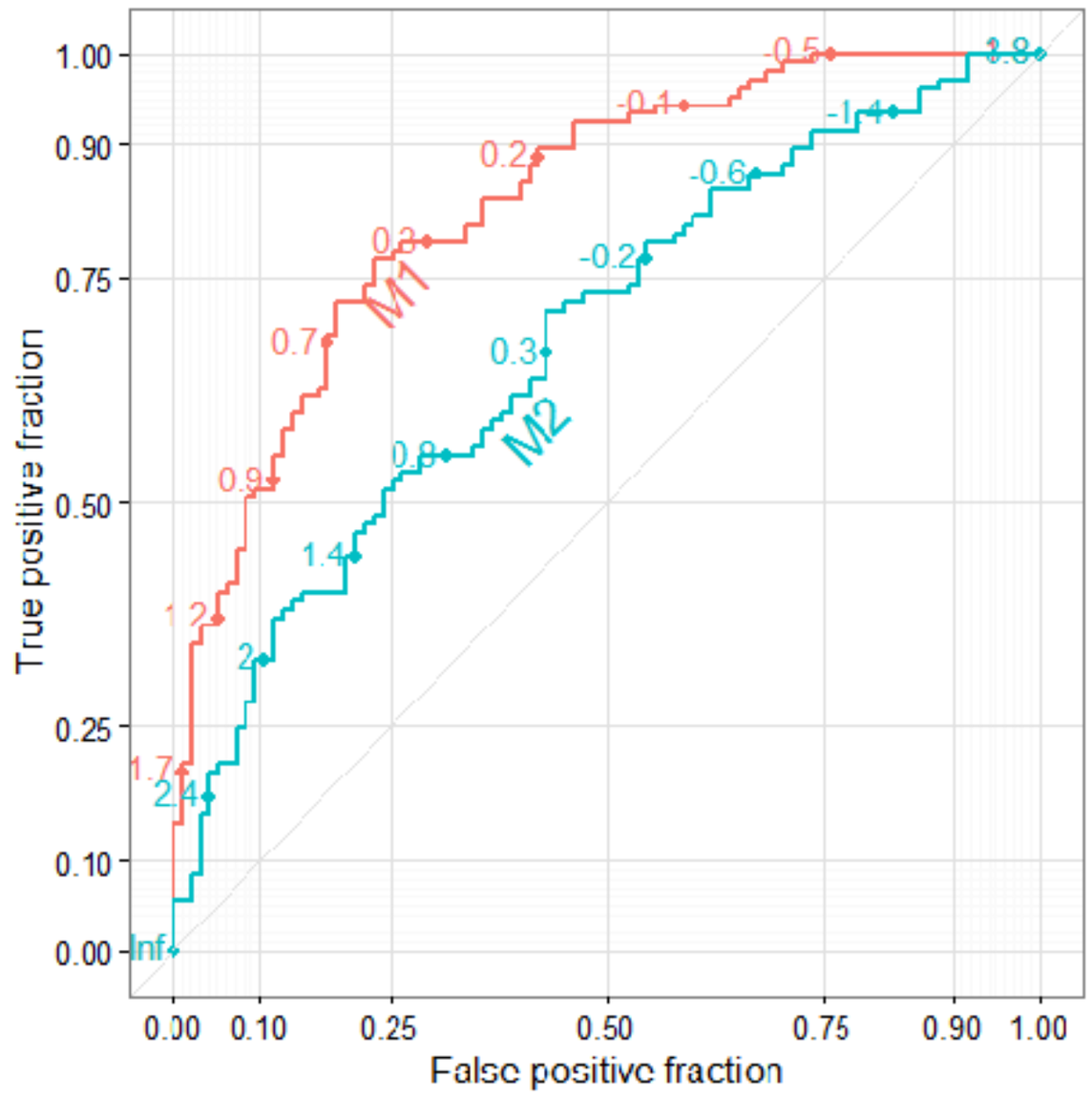
- The main function: mergeFactors()
- Works for likelihood-based regression models
- Plot a single summary - merging paths plots, rich in information about group's similarities
- Produces ggplot2 plots

Agnieszka Sitko, Przemysław Biecek (2017)
https://github.com/geneticsMiNIng/FactorMerger

Validation

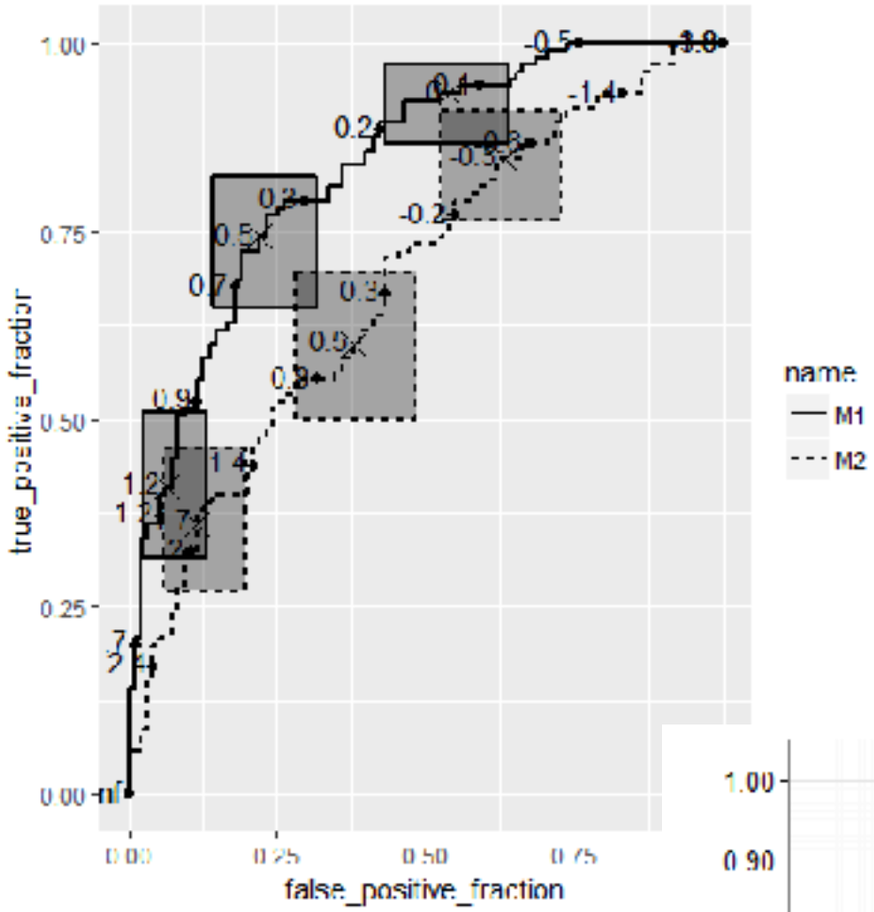
Package plotROC

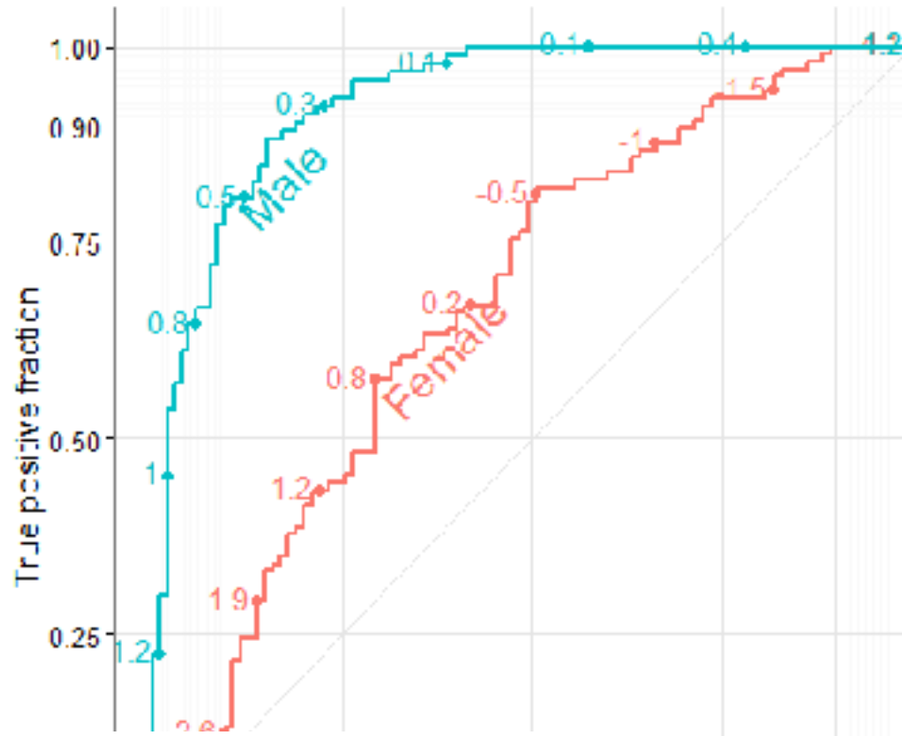


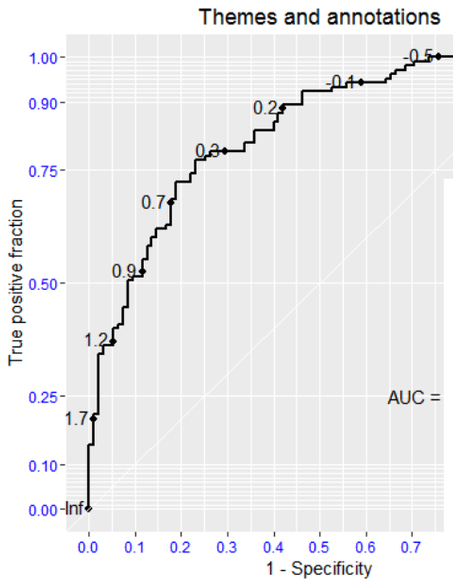


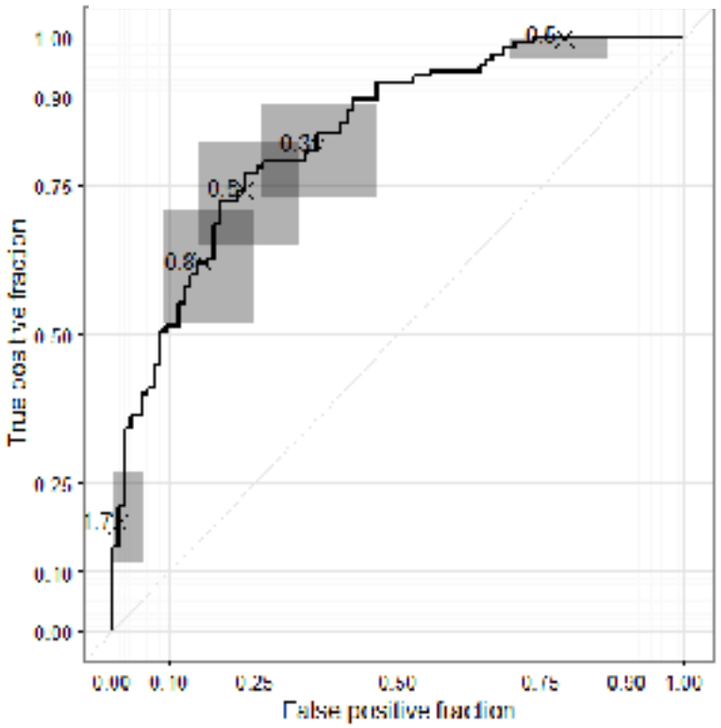
Michael Sachs (2016) http://sachsmc.github.io/plotROC

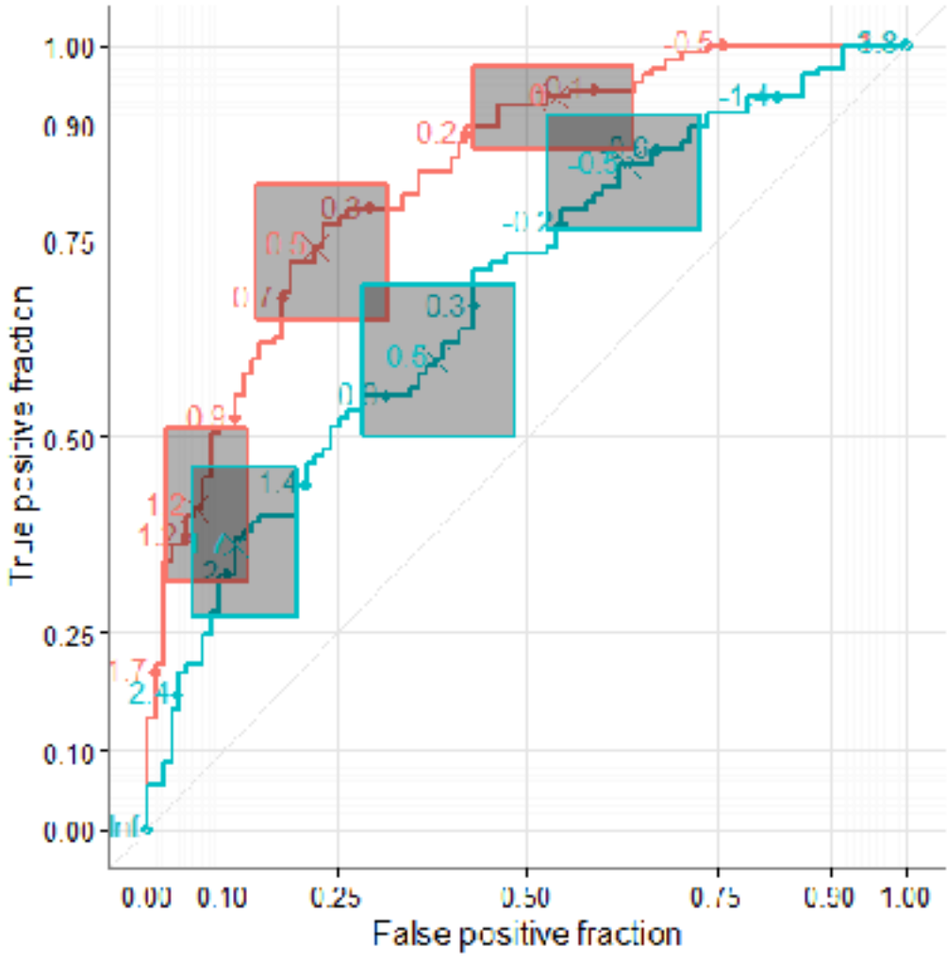
Package plotROC











Michael Sachs (2016) http://sachsmc.github.io/plotROC

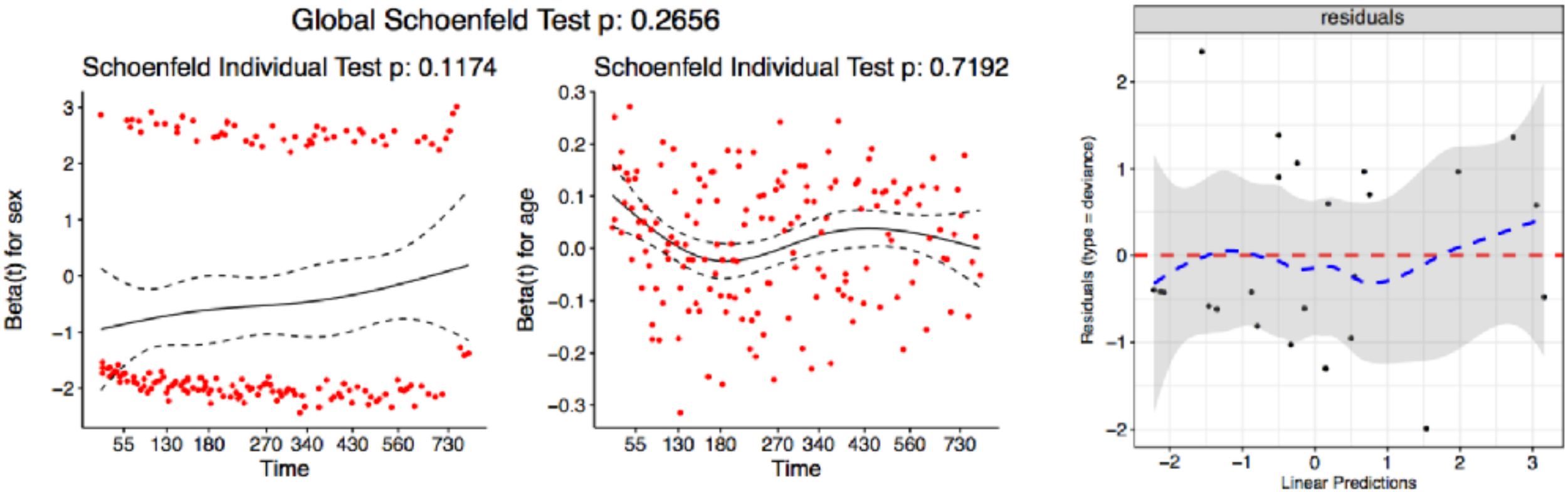
Package plotROC

- The main function: geom_roc()
- Works for multiple models
- For each model plots a ROC curve with confidence intervals (with geom_rocci())
- Produces ggplot2 plots

Michael Sachs (2016) http://sachsmc.github.io/plotROC

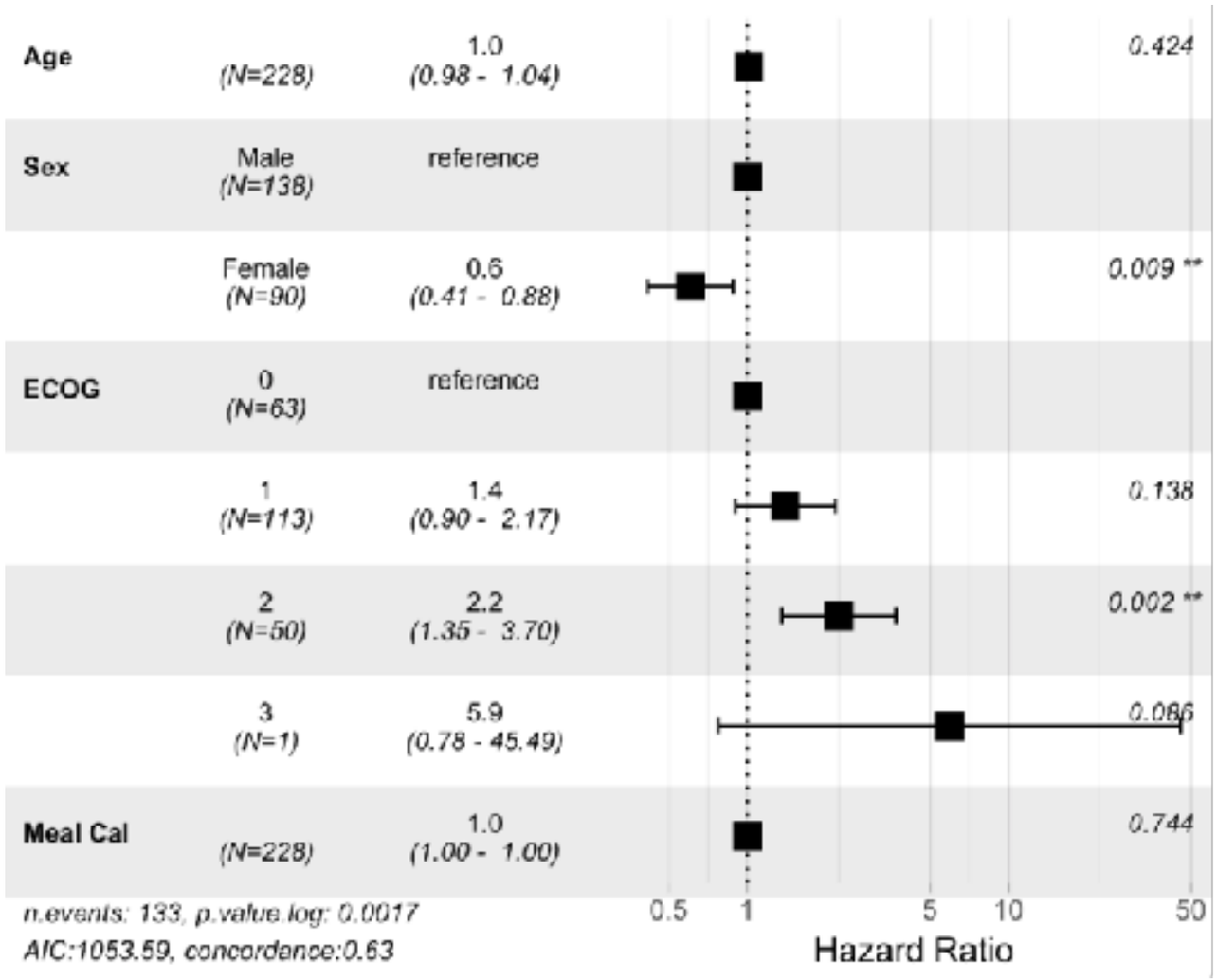
Package survminer

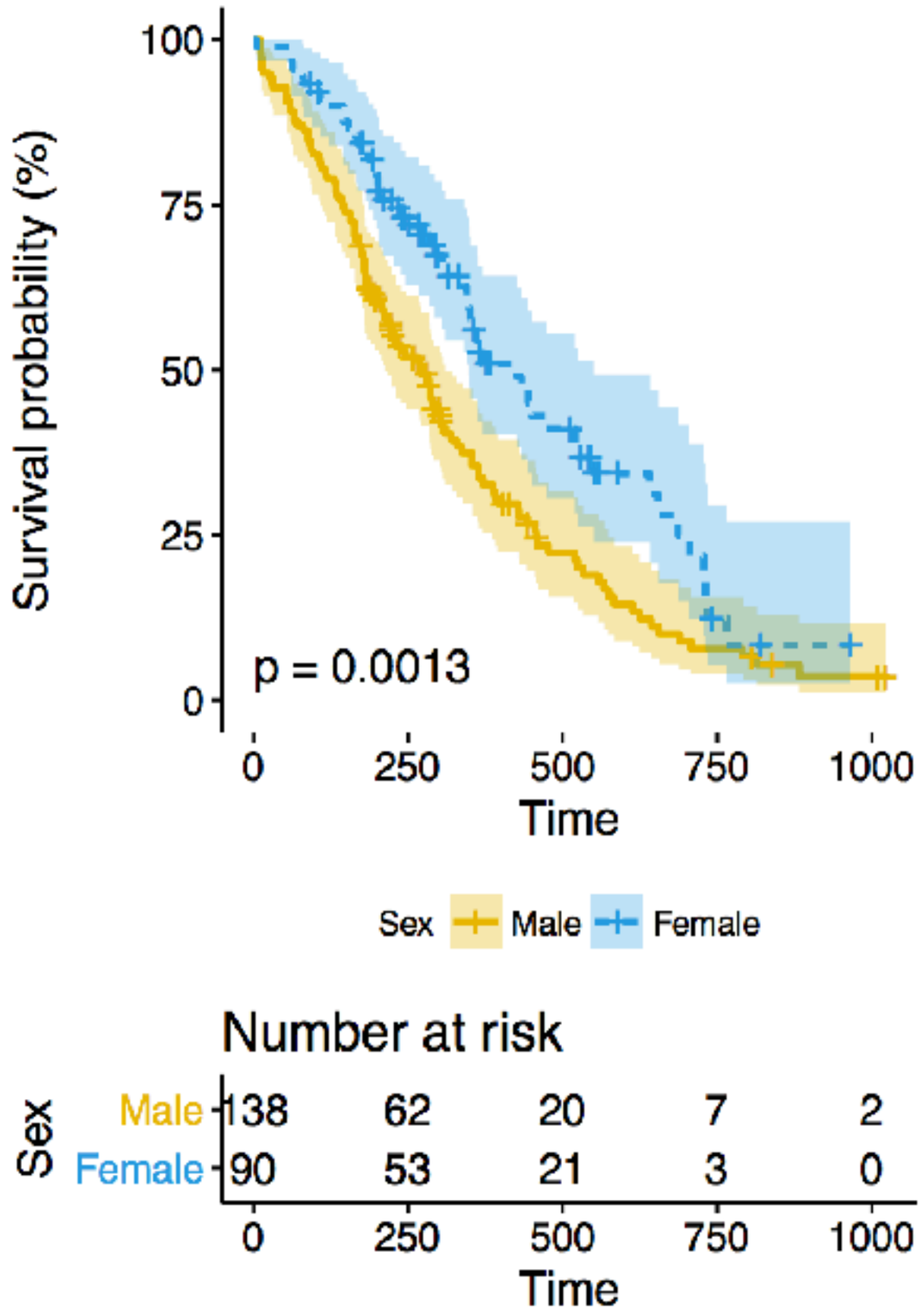
Diagnostic plots for various residuals: "martingale", "deviance", "score", "schoenfeld", "dfbeta", "dfbetas" and "scaledsch"



Alboukadel Kassambara, Marcin Kosiński (2016) https://github.com/kassambara/survminer

Package survminer





Alboukadel Kassambara, Marcin Kosiński (2016) https://github.com/kassambara/survminer

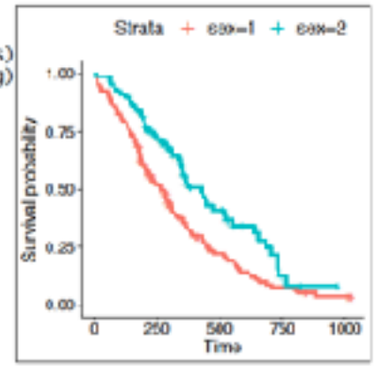
Creating Survival Plots

Informative and Elegant with *survminer*

Survival Curves

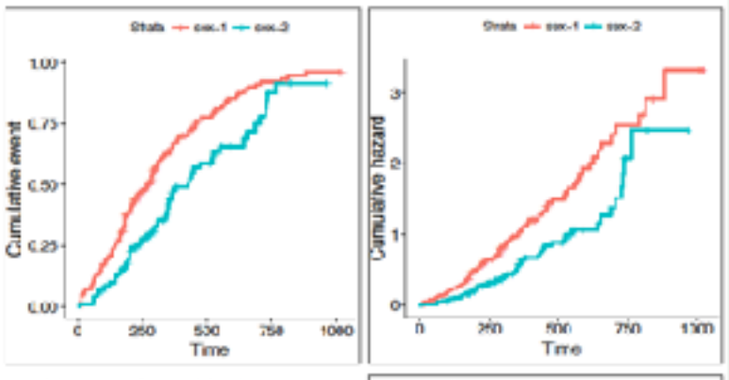
The **ggsurvplot()** function creates **ggplot2** plots from **survfit** objects.

```
library("survival")
fit <- survfit(Surv(time, status)
               ~ sex, data = lung)

class(fit)
## [1] "survfit"

library("survminer")
ggsurvplot(fit, data = lung)
```



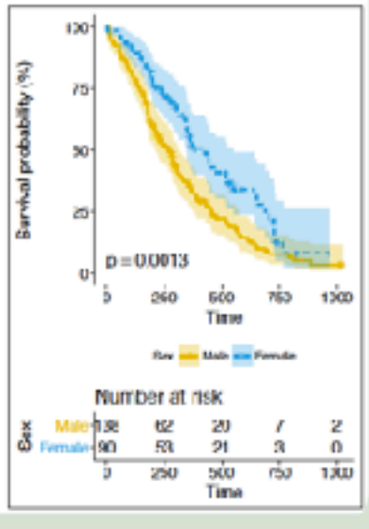
Use the **fun** argument to set the transformation of the survival curve. E.g. **"event"** for cumulative events, **"cumhaz"** for the cumulative hazard function or **"pct"** for survival probability in percentage.

```
ggsurvplot(fit, data = lung, fun = "event")
ggsurvplot(fit, data = lung, fun = "cumhaz")
```



With lots of graphical parameters you have full control over look and feel of the survival plots; position and content of the legend; additional annotations like p-value, title, subtitle.

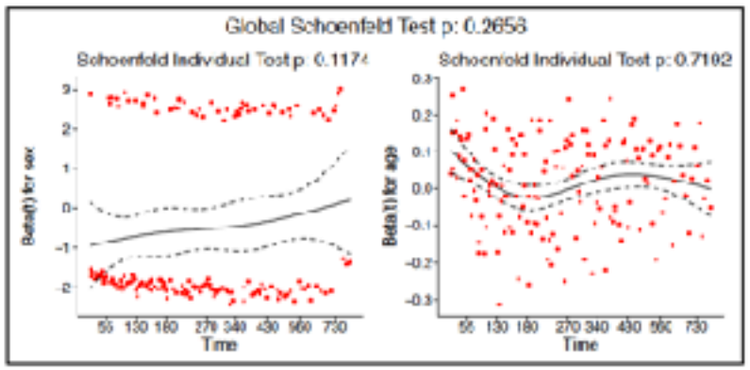
```
ggsurvplot(fit, data = lung,
    conf.int = TRUE,
    pval = TRUE,
    fun = "pct",
    risk.table = TRUE,
    size = 1,
    linetype = "strata",
    palette = c("#E7B800",
                "#2E9FDF"),
    legend = "bottom",
    legend.title = "Sex",
    legend.labs = c("Male",
                    "Female"))
```



Diagnostics of Cox Model

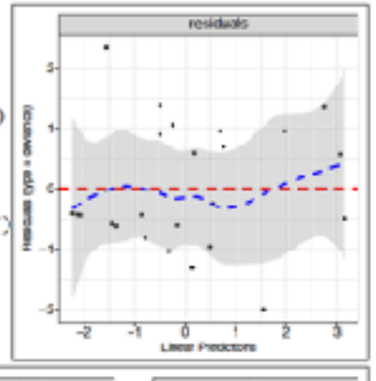
The function **cox.zph()** from **survival** package may be used to test the proportional hazards assumption for a Cox regression model fit. The graphical verification of this assumption may be performed with the function **ggcoxzph()** from the **survminer** package. For each covariate it produces plots with scaled Schoenfeld residuals against the time.

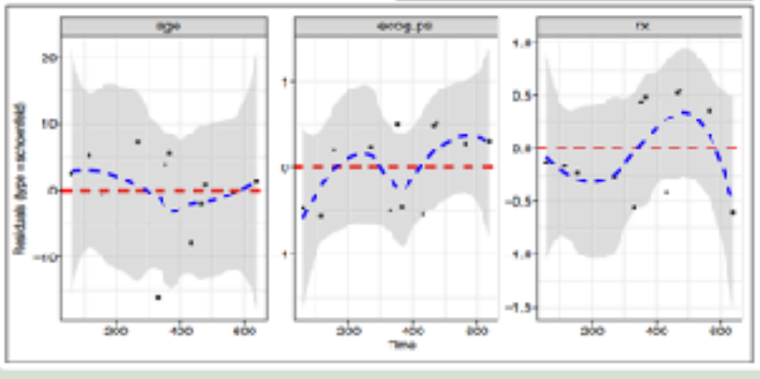
```
library("survival")
fit <- coxph(Surv(time, status) ~ sex + age, data = lung)
ftest <- cox.zph(fit)
ftest
##         rho chisq     p
## sex    0.1236 2.452 0.117
## age   -0.0275 0.129 0.719
## GLOBAL      NA 2.651 0.266
library("survminer")
ggcoxzph(ftest)
```



The function **ggcoxdiagnostics()** plots different types of residuals as a function of time, linear predictor or observation id. The type of resicual is selected with **type** argument. Possible values are "martingale", "deviance", "score", "schoenfeld", "dfbeta", "dfbetas", and "scaledsch".
The **ox.scale** argument defines what shall be plotted on the OX axis. Possible values are "linear.predictions", "observation.id", "time".
Logical arguments **hline** and **sline** may be used to add horizontal line or smooth line to the plot.

```
library("survival")
library("survminer")
fit <- coxph(Surv(time, status) ~
             sex + age, data = lung)

ggcoxdiagnostics(fit,
    type = "deviance",
    ox.scale = "linear.predictions")

ggcoxdiagnostics(fit,
    type = "schoenfeld",
    ox.scale = "time")
```

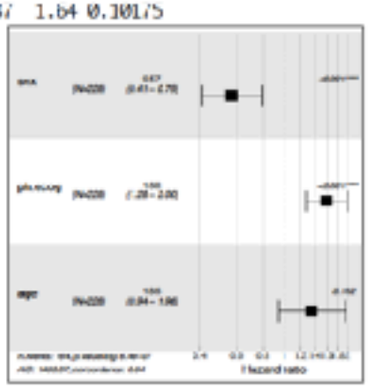




Summary of Cox Model

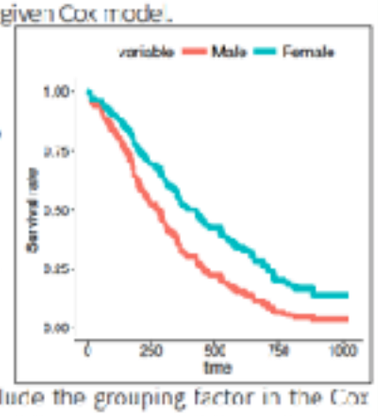
The function **ggforest()** from the **survminer** package creates a forest plot for a Cox regression model fit. Hazard ratio estimates along with confidence intervals and p values are plotted for each variable.

```
library("survival")
library("survminer")
lung$age <- ifelse(lung$age > 70, ">70","<= 70")
fit <- coxph( Surv(time, status) ~ sex + ph.ecog + age, data = lung)
fit

## Call:
## coxph(formula = Surv(time, status) ~ sex+ph.ecog+age, data=lung)
##
##              coef exp(coef) se(coef)     z       p
## sex       -0.567     0.567    0.158 -3.37 0.00075
## ph.ecog    0.470     1.600    0.113  4.16 3.1e-05
## age>70     0.307     1.359    0.157  1.64 0.10175
##
## Likelihood ratio test=31.6  on
## n= 227, number of events= 164

ggforest(fit)
```



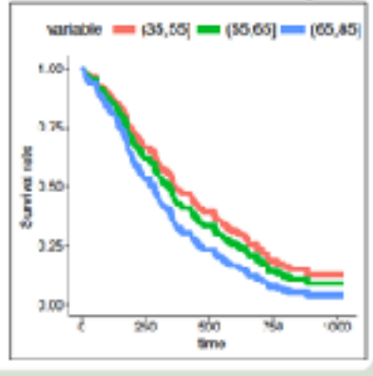
The function **ggcoxadjustedcurves()** from the **survminer** package plots Adjusted Survival Curves for Cox Proportional Hazards Model. Adjusted Survival Curves show how a selected factor influences survival estimated from a Cox model.
Note that these curves differ from Kaplan Meier estimates since they present expected survival based on given Cox model.

```
library("survival")
library("survminer")

lung$sex <- ifelse(lung$sex == 1,
                   "Male", "Female")

fit <- coxph(Surv(time, status) ~
             sex + ph.ecog + age,
             data = lung)

ggcoxadjustedcurves(fit, data=lung,
                    variable=lung$sex)
```



Note that it is not necessary to include the grouping factor in the Cox model. Survival curves are estimated from Cox model for each group defined by the factor independently.

```
lung$age3 <- cut(lung$age,
                 c(35,55,65,85))

ggcoxadjustedcurves(fit, data=lung,
                    variable=lung$age3)
```



This onepager presents the **survminer** package [Alboukadel Kassambara, Marcin Kosinski 2017] in version 0.3.1
See *https://github.com/kassambara/survminer/* for more details.

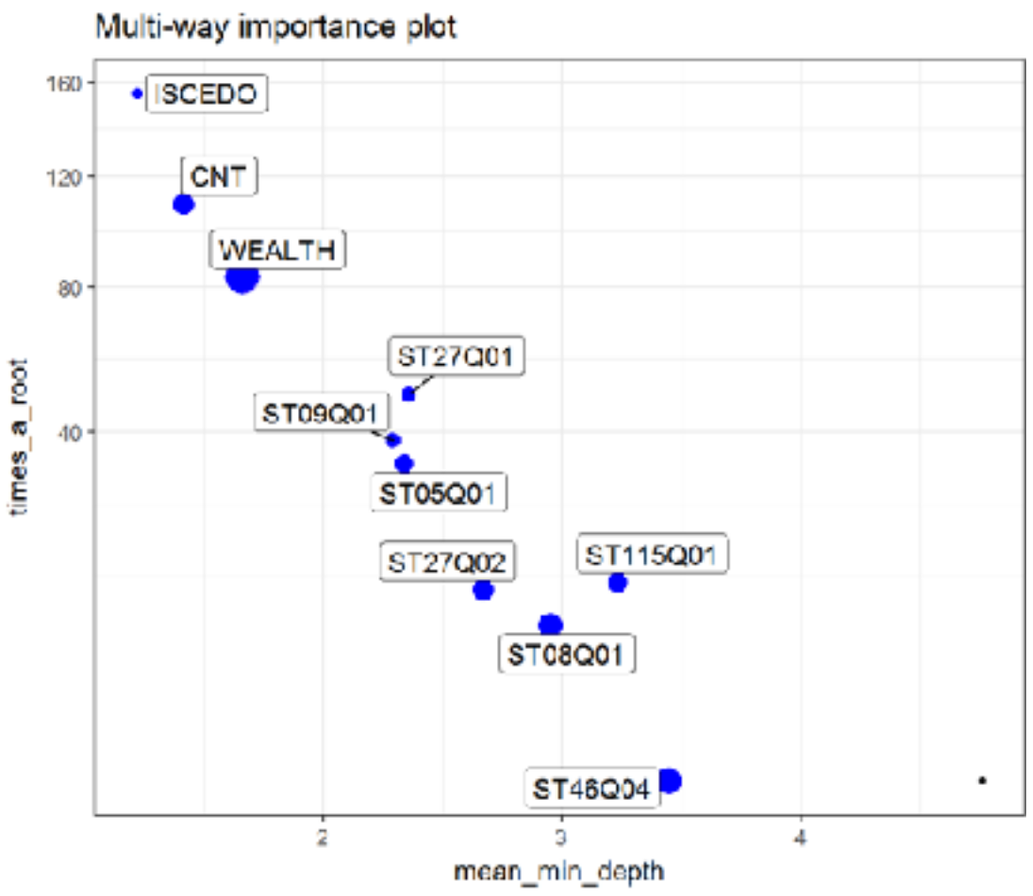
Package survminer

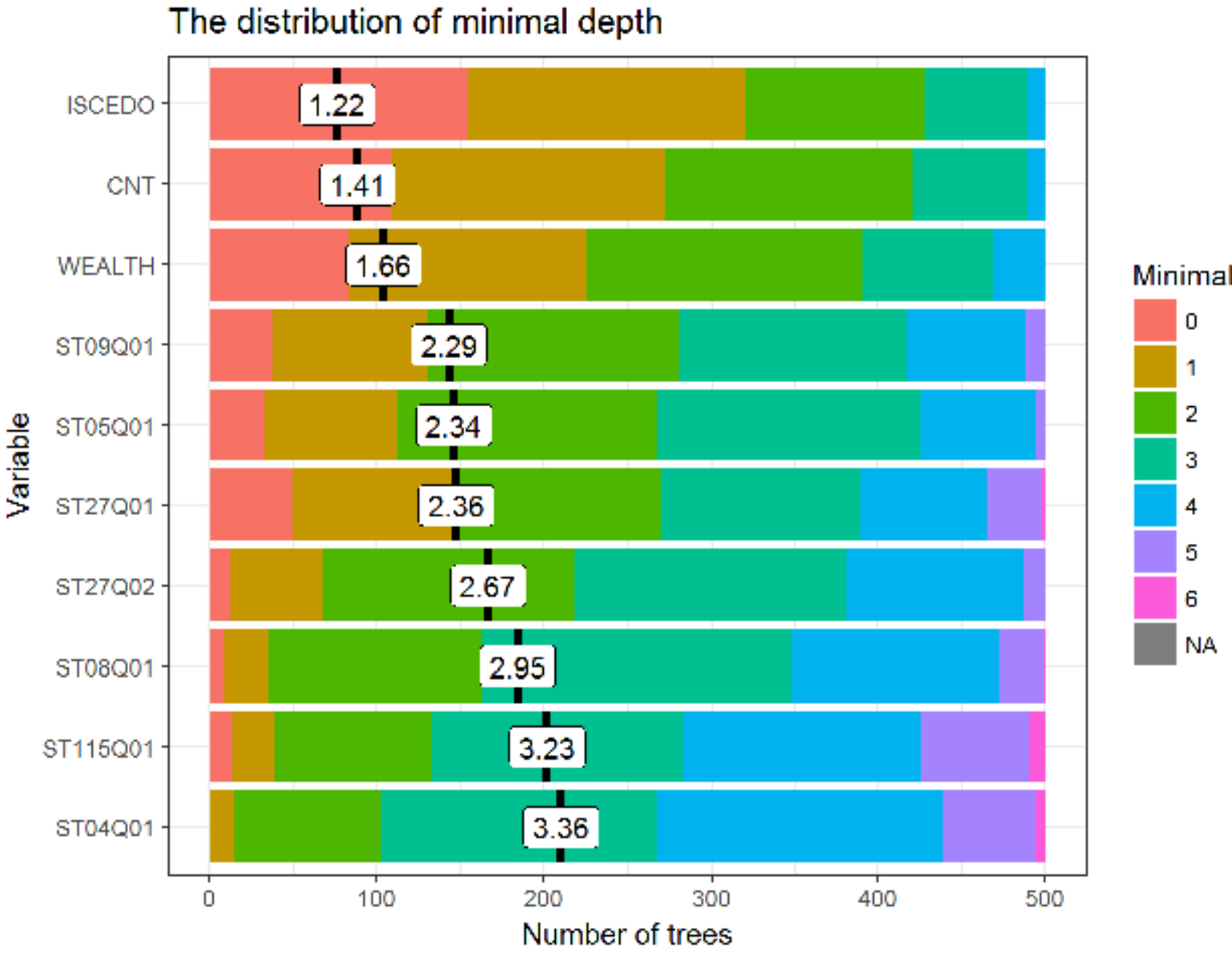
- Set of functions with many arguments
- Works for selected survival models (e.g. CoxPH, survfit, cmprisk)
- Plots many (all popular) graphical summaries for given models
- Produces ggplot2 plots

Alboukadel Kassambara, Marcin Kosiński (2016) https://github.com/kassambara/survminer

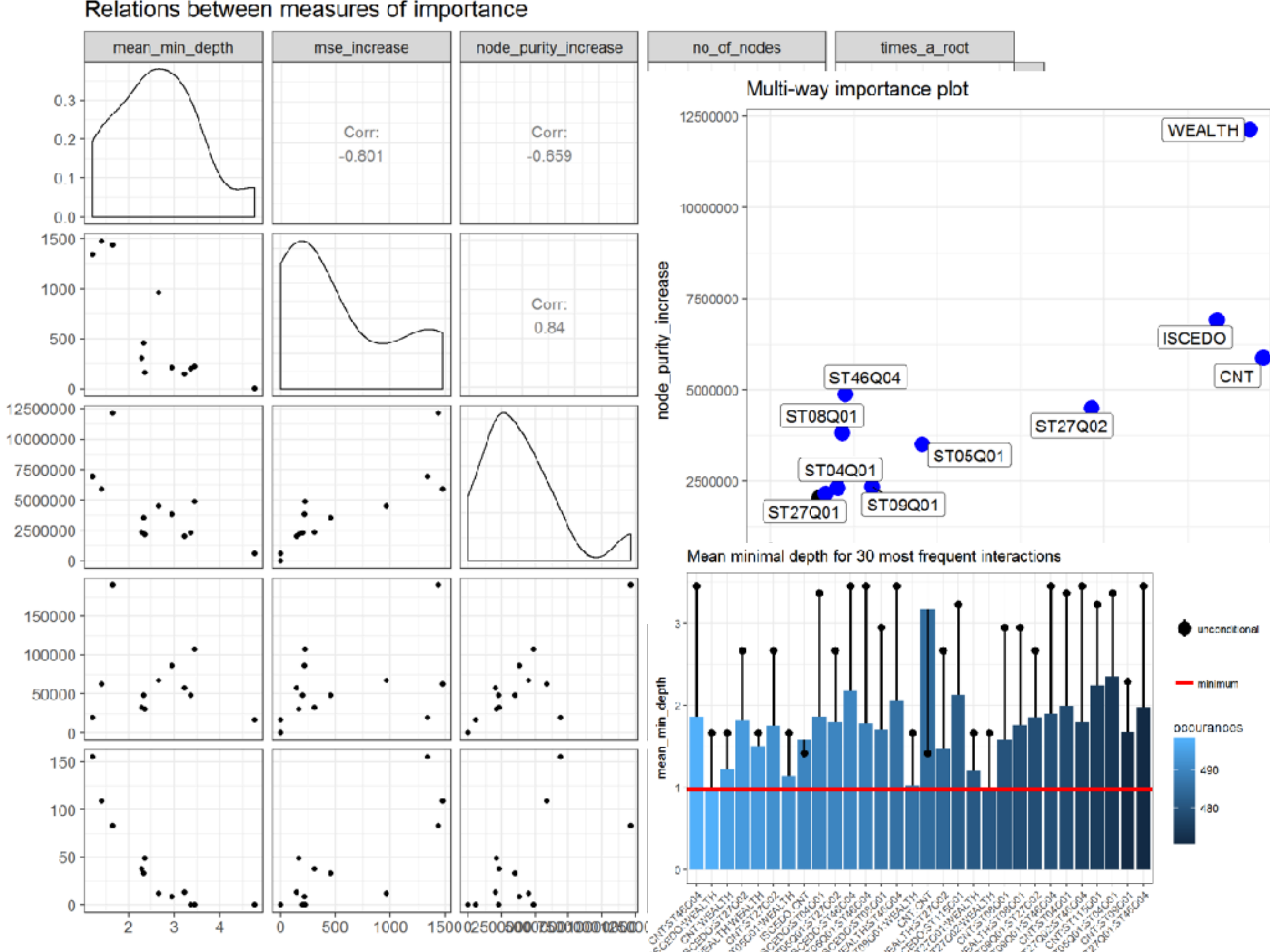
Prediction

Package randomForestExplainer
What is in a random forest?





Aleksandra Paluszynska, Przemyslaw Biecek (2017)
https://github.com/geneticsMiNIng/BlackBoxOpener
John Ehrlinger (2015) ggRandomForests: Random Forests for Regression

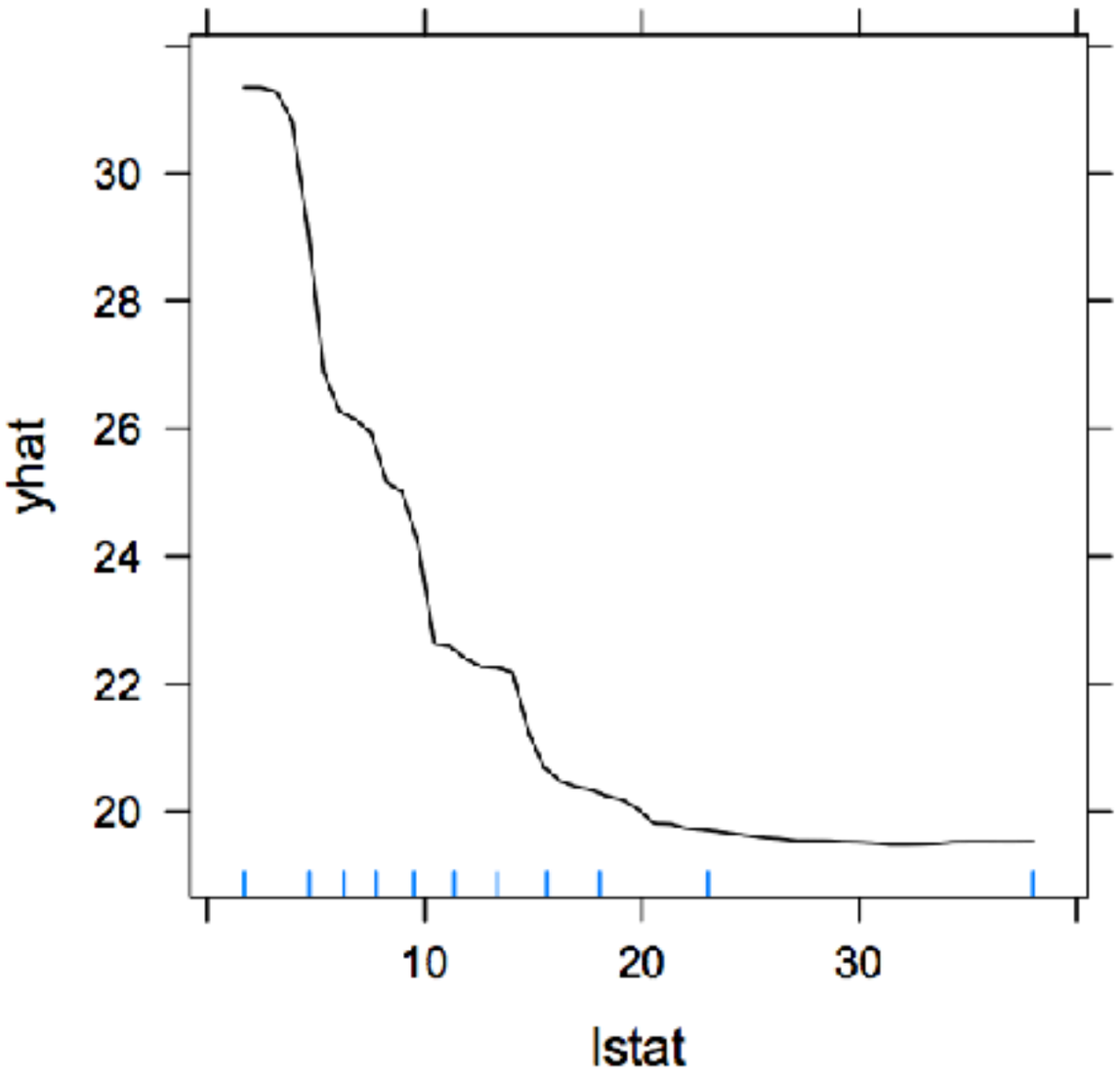

Relations between measures of importance
mean_min_depth
mse_increase
node_purity_increase
no_of_nodes
times_a_root
Corr:
-0.801
Corr:
-0.659
Corr:
0.84
Multi-way importance plot
node_purity_increase
WEALTH
ISCEDO
CNT
ST46Q04
ST08Q01
ST27Q02
ST05Q01
ST04Q01
ST09Q01
ST27Q01
Mean minimal depth for 30 most frequent interactions
mean_min_depth
unconditional
minimum
occurances

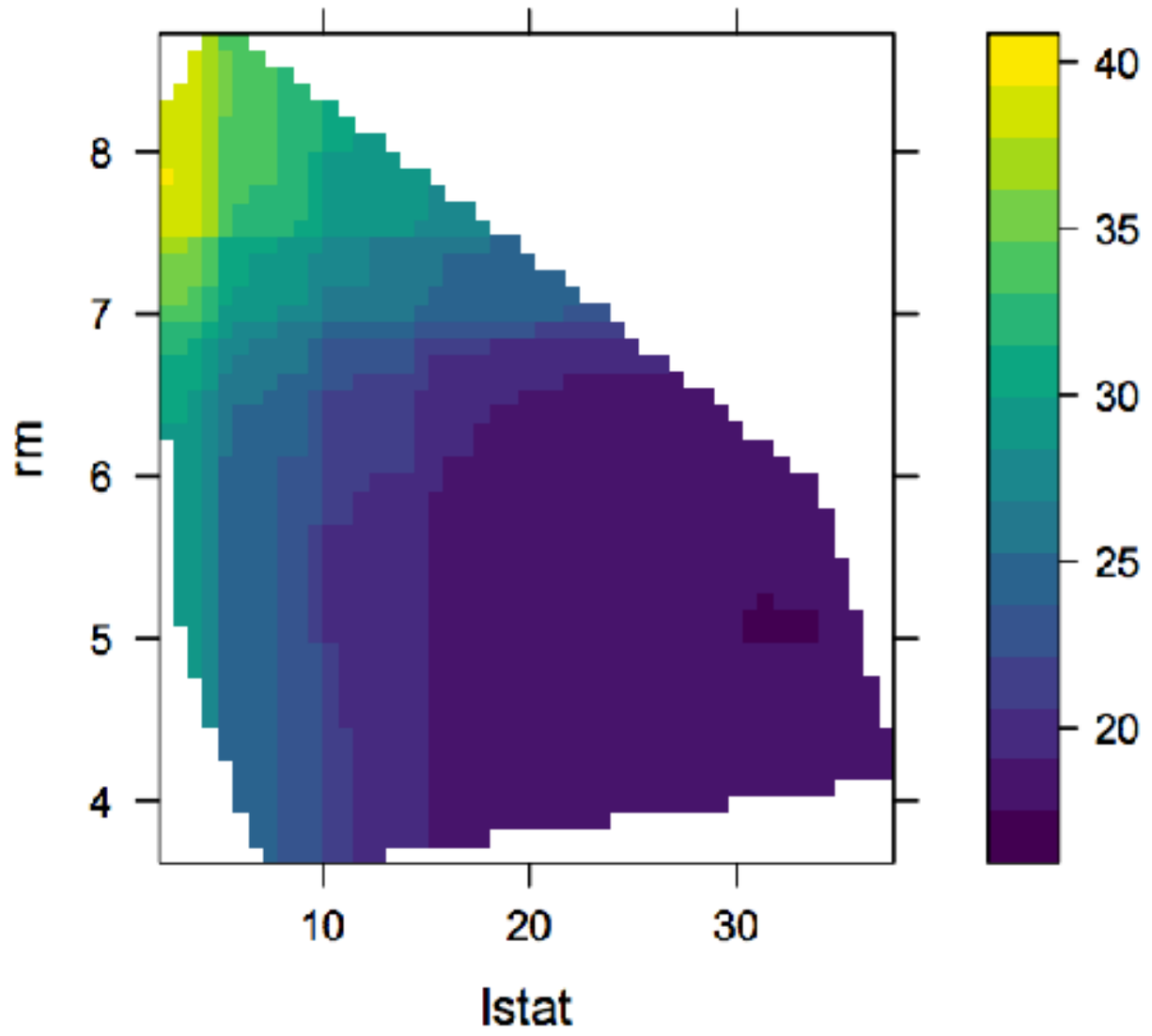
Package randomForestExplainer

- The main function: summarize_forest()
- Works for randomForest objects
- Calculates and plots various measures of variable importance
- Produces ggplot2 plots

Aleksandra Paluszynska, Przemyslaw Biecek (2017)
https://github.com/geneticsMiNIng/BlackBoxOpener
John Ehrlinger (2015) ggRandomForests: Random Forests for Regression

Package pdp - Partial Dependence Plots



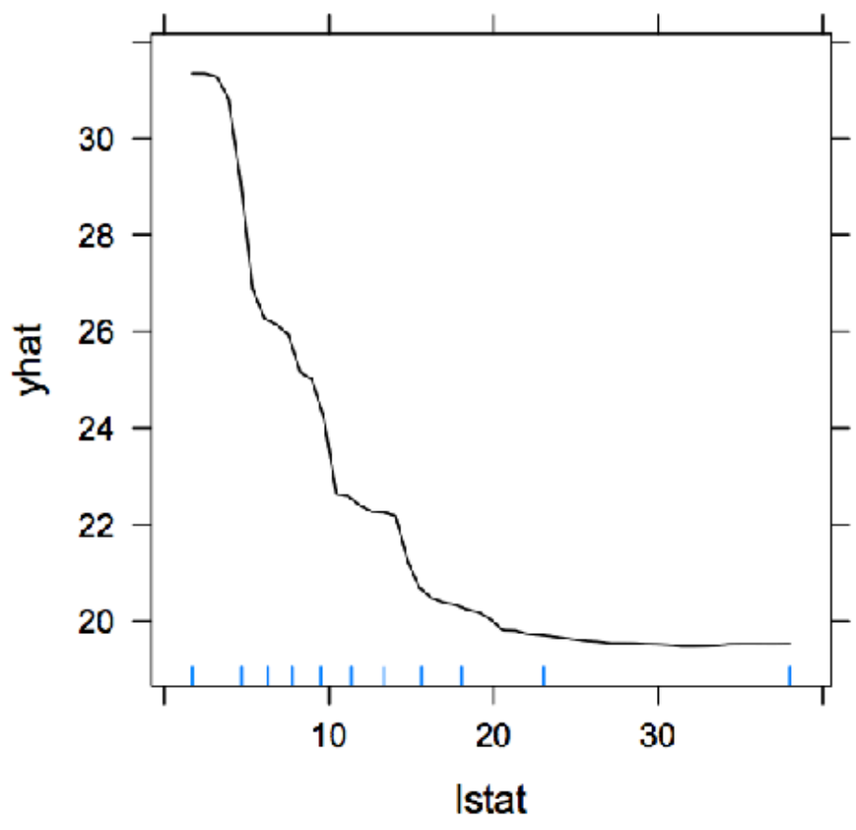


pdp: An R Package for Constructing Partial Dependence Plots. Brandon M. Greenwell (2017)
https://journal.r-project.org/archive/2017/RJ-2017-016/index.html

Package pdp - Partial Dependence Plots

Type of model	R package	Object class
Decision tree	C50 (Kuhn et al., 2015)	"C5.0"
	party	"BinaryTree"
	partykit	"party"
	rpart (Therneau et al., 2015)	"rpart"
Bagged decision trees	adabag (Alfaro et al., 2013)	"bagging"
	ipred (Peters and Hothorn, 2015)	"classbagg", "regbagg"
Boosted decision trees	adabag (Alfaro et al., 2013)	"boosting"
	gbm	"gbm"
	xgboost	"xgb.Booster
Cubist	Cubist (Kuhn et al., 2014)	"cubist"
Discriminant analysis	MASS (Venables and Ripley, 2002)	"lda", "qda"
Generalized linear model	**stats**	"glm", "lm"
Linear model	**stats**	"lm"
Nonlinear least squares	**stats**	"nls"
Multivariate adaptive regression splines (MARS)	earth (Milborrow, 2016)	"earth"
	mda (Leisch et al., 2016)	"mars"
Projection pursuit regression	**stats**	"ppr"
Random forest	**randomForest**	"randomForest"
	party	"RandomForest"
	partykit	"cforest"
	ranger (Wright, 2016)	"ranger"
Support vector machine	e1071 (Meyer et al., 2015)	"svm"
	kernlab (Karatzoglo	

Table 1: Models specifically supported by the **pdp** pacl
may still need to supply additional arguments in the cal



pdp: An R Package for Constructing Partial Dependence Plots
Brandon M. Greenwell (2017)
https://journal.r-project.org/archive/2017/RJ-2017-016/index.html

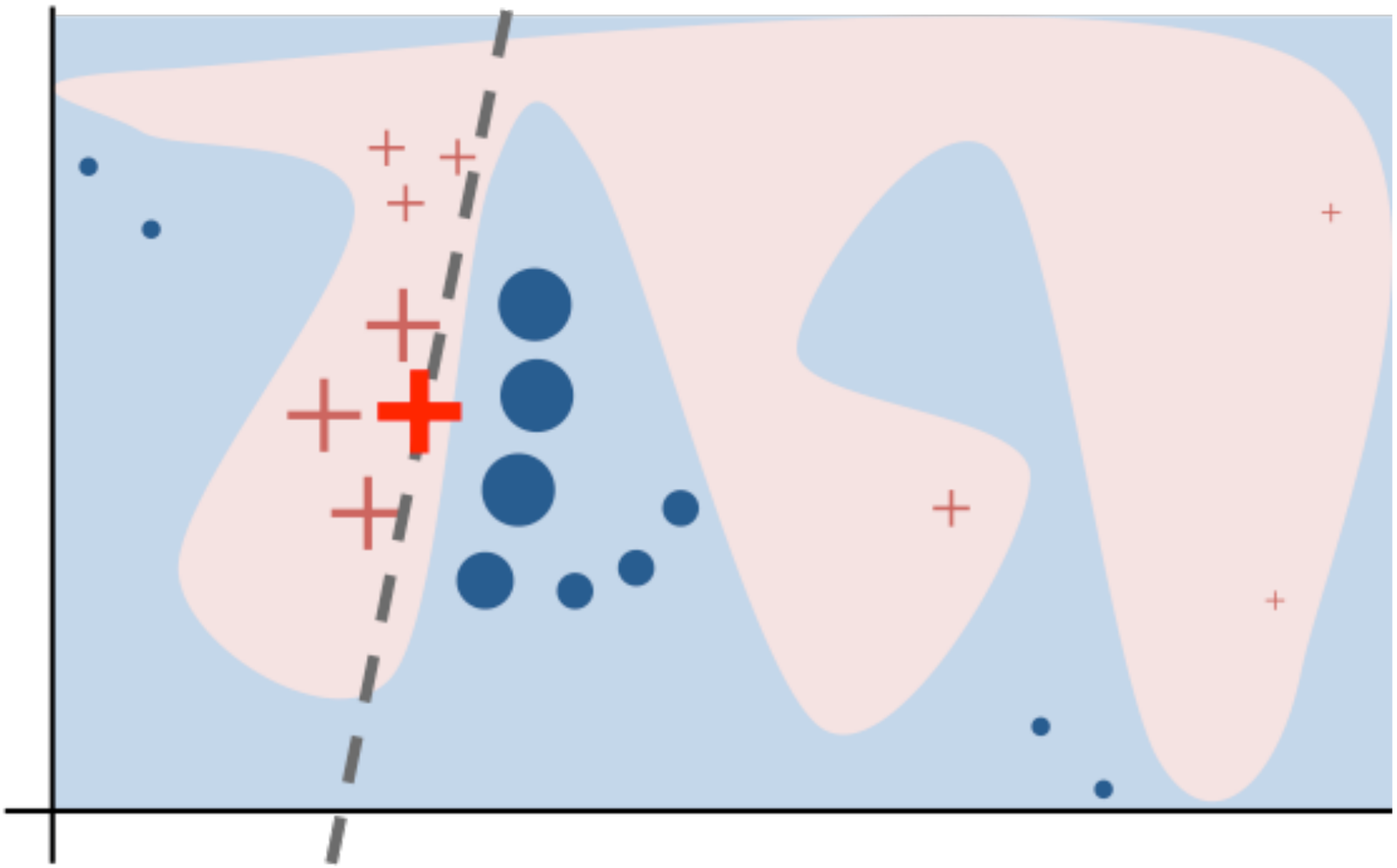
Package pdp - Partial Dependence Plots

- The main function: partial()
- Works for various models
- Calculates and plots partial dependence functions
- Produces ggplot2 plots and lattice plots

pdp: An R Package for Constructing Partial Dependence Plots. Brandon M. Greenwell (2017)
https://journal.r-project.org/archive/2017/RJ-2017-016/index.html

LIME: Local Interpretable Model-agnostic Explanations

1. Generate a fake dataset around x.
2. Use black-box estimator to get target values y.
3. Train a new white-box estimator for (y,x).
4. Check prediction quality of a white-box classifier.
5. Use white-box estimator as an explanation of black-box model.



"Why Should I Trust You?" Explaining the Predictions of Any Classifier.
Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin (2016). https://arxiv.org/pdf/1602.04938.pdf

Port to R: Thomas Lin Pedersen (2017) https://github.com/thomasp85/lime

LIME: Local Interpretable Model-agnostic Explanations

(a) Original Image

(b) Explaining *Electric guitar*

What went wrong
and what went OK

(a) Husky classified as wolf

(b) Explanation

An R port https://github.com/thomasp85/lime

"Why Should I Trust You?" Explaining the Predictions of Any Classifier.
Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin (2016). https://arxiv.org/pdf/1602.04938.pdf

Package lime - Local Interpretable Model-agnostic Explanations

- R port of a python package https://github.com/marcotcr/lime
- Originated from text-analysis / image-analysis
- Model agnostic
- Fits different ,,white box'' models

"Why Should I Trust You?" Explaining the Predictions of Any Classifier.
Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin (2016). https://arxiv.org/pdf/1602.04938.pdf

Reproducibility

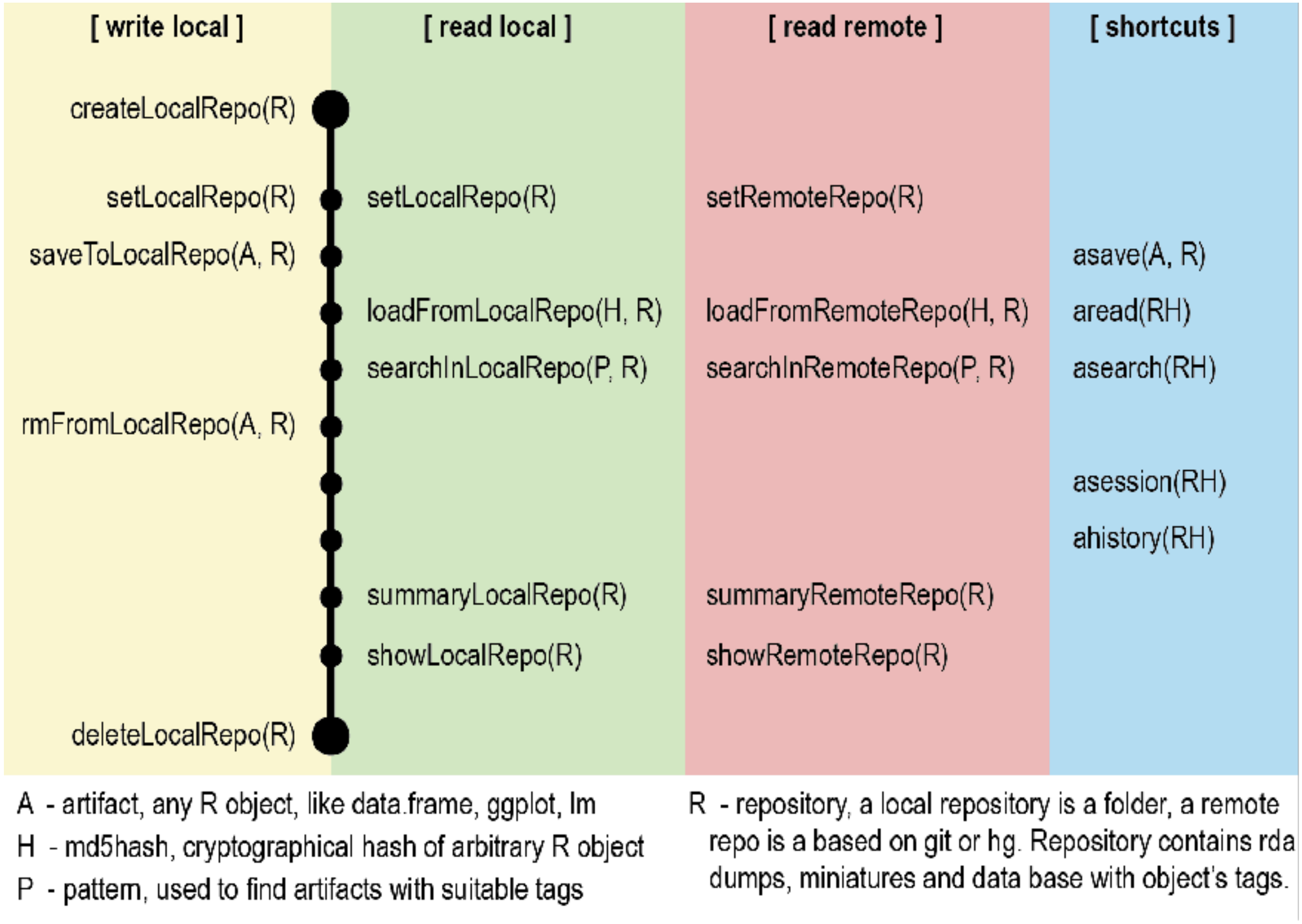
archivist - reproducible and recordable research

```
library("archivist")
model <- lm(Sepal.Length ~ Sepal.Width, data=iris)
saveToLocalRepo(model)

models <- asearch("pbiecek/graphGallery", patterns = "class:lm")
modelsBIC <- sapply(models, BIC)
sort(modelsBIC)
```

```
## 990861c7c27812ee959f10e5f76fe2c3 2a6e492cb6982f230e48cf46023e2e4f
##                         39.05577                         67.52735
## 0a82efeb8250a47718cea9d7f64e5ae7 378237103bb60c58600fe69bed6c7f11
##                        189.73593                        189.73593
## 7f11e03539d48d35f7e7fe7780527ba7 c1b1ef7bcddefb181f79176015bc3931
##                        189.73593                        189.73593
```

Przemysław Biecek, Marcin Kosiński (2015) https://github.com/pbiecek/archivist

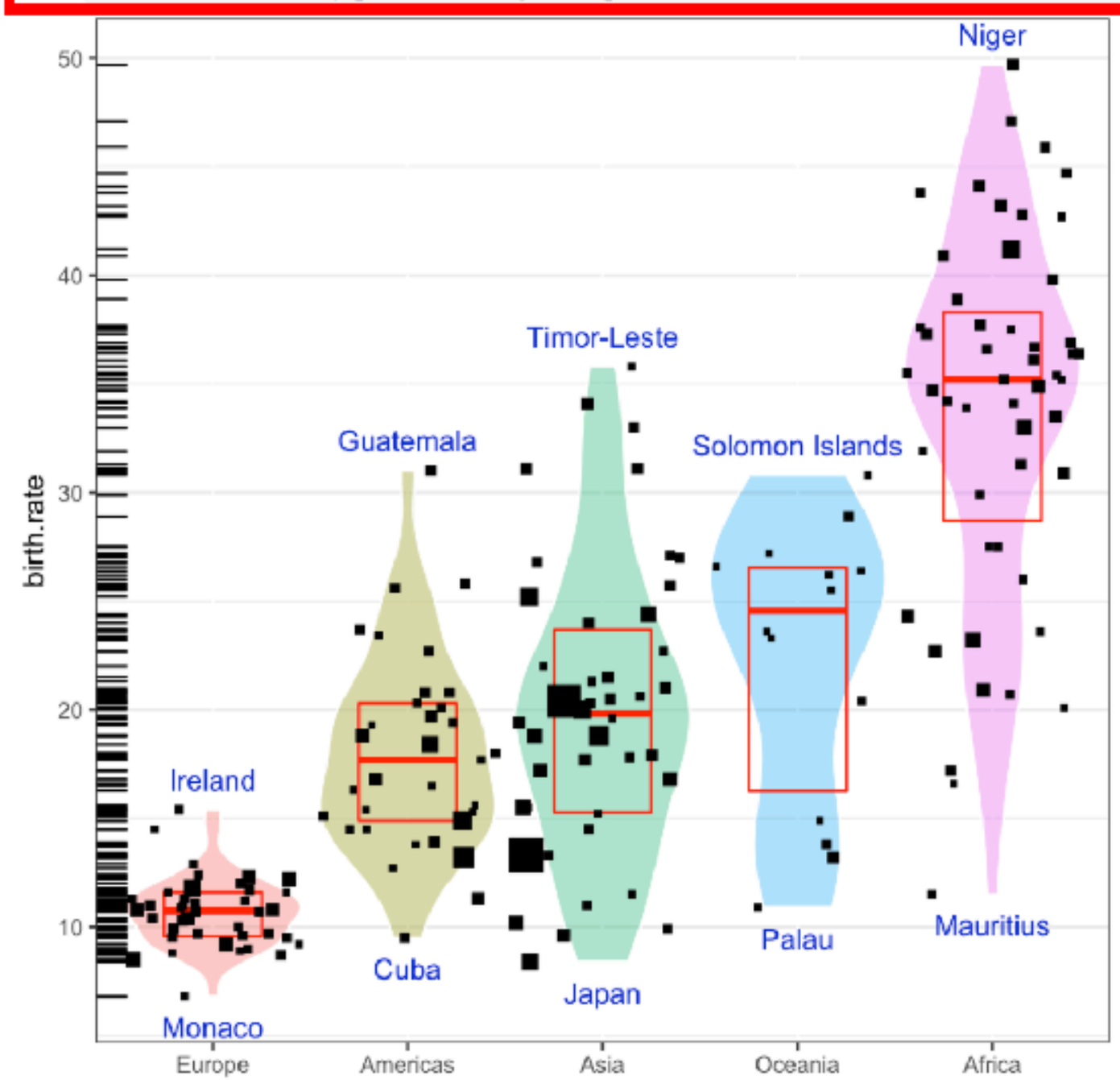
archivist - reproducible and recordable research



Przemysław Biecek, Marcin Kosiński (2015) https://github.com/pbiecek/archivist

```
ggplot(countries, aes(x=continent, y=birth.rate, label=country)) +
  geom_violin(scale="width", aes(fill=continent), color="white", alpha=0.4) +
  stat_summary(fun.data = "q3", geom = "crossbar",
               colour = "red", width = 0.5) +
  geom_jitter(aes(size=(population)^0.9),position=position_jitter(width = .45, height = 0),
              shape=15) +
  geom_rug(sides = "l") +
  geom_text(data=countriesMin, vjust=2, color="blue3") +
  geom_text(data=countriesMax, vjust=-1, color="blue3") +
  theme_bw() + xlab("") + theme(legend.position="none", panel.grid.major.x = element_line(color="whi
te"))
```

```
Load: archivist::aread('pbiecek/Eseje/arepo/ba7f58fafe7373420e3ddce039558140')
```

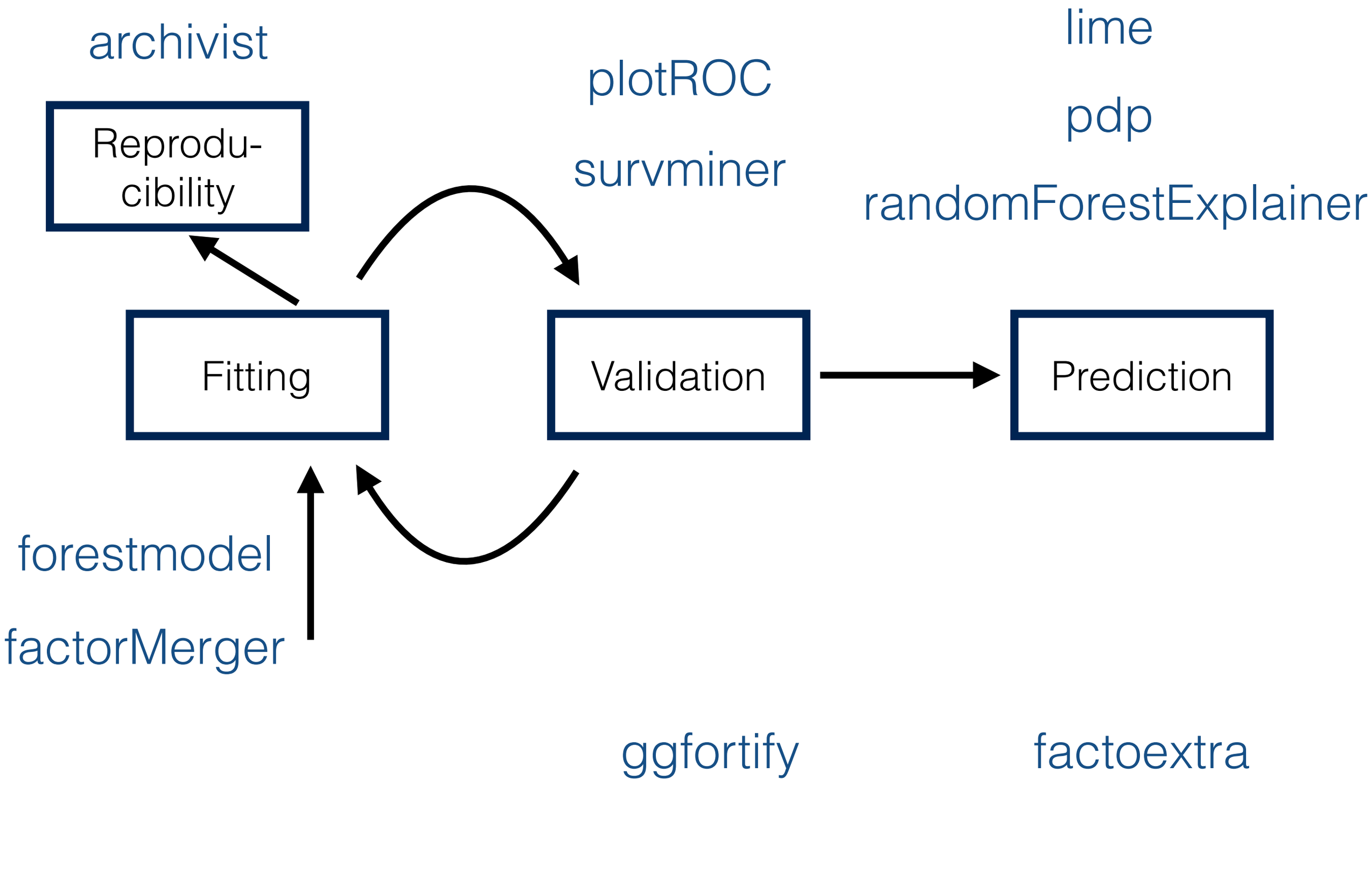


archivist - reproducible and recordable research

- Set of functions for storing, searching for, and recovering binary copies of R objects
- Repository of R objects that can be hosted as a local foley or as a git/mercurial folder
- Generates a md5 checksum to verify object's identity

Przemysław Biecek, Marcin Kosiński (2015) https://github.com/pbiecek/archivist

Life-cycle of a typical prognostic model



Further reading

ELI5 is a Python library which allows to visualize and debug various Machine Learning models
http://eli5.readthedocs.io/en/latest/index.html

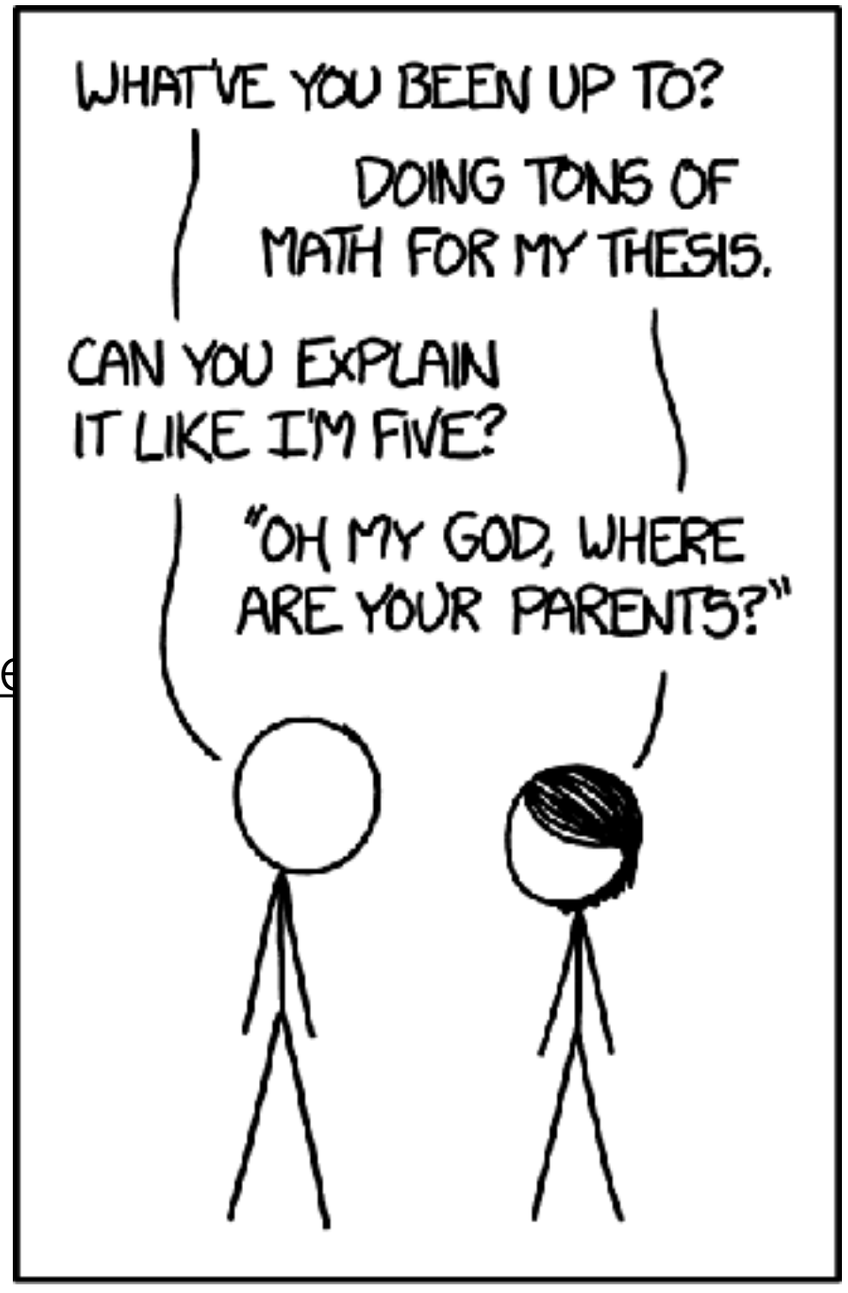
Ideas on interpreting machine learning.
Patrick Hall, Wen Phan, SriSatish Ambati (2017)
https://www.oreilly.com/ideas/ideas-on-interpreting-machine-le

ggRandomForests: Random Forests for Regression
John Ehrlinger (2015)
https://arxiv.org/pdf/1501.07196.pdf

rms: Regression Modeling Strategies.
Frank E Harrell (2009-2017) CRAN
https://cran.r-project.org/web/packages/rms/index.html

Visualizing statistical models: Removing the blindfold.
Hadley Wickham, Dianne Cook, Heike Hofmann (2015)
Statistical Analysis and Data Mining
http://had.co.nz/stat645/model-vis.pdf

Thank you for your attention!



https://imgs.xkcd.com/comics/like_im_five.png