

Streng Jacke!

Examples for sjPlotting functions, including correlations and proportional tables with ggplot #rstats

VERÖFFENTLICHT AM 18. APRIL 2013 VON DANIEL

2

~~Some times ago (http://en.wordpress.com/my-plotting)~~ functions I show here can be reproduced without having a SPSS data set (or at least, without having the data set I use because it's not public yet). So I started to write some examples that run "out of the box" and which I want to present you here. Furthermore, two new plotting functions are introduced: plotting correlations and plotting proportional tables on a percentage scale.

As always, you can find the latest version of my R scripts on my download page (<http://strengjacke.wordpress.com/r-scripts/>).

Following plotting functions will be described in this posting:

- Plotting proportional tables: sjPlotPropTable.R
- Plotting correlations: sjPlotCorr.R
- Plotting frequencies: sjPlotFrequencies.R
- Plotting grouped frequencies: sjPlotGroupFrequencies.R
- Plotting linear model: sjPlotLinreg.R
- Plotting generalized linear models: sjPlotOdds.R

Please note that I have changed function and parameter names in order to have consistent, logical names across all functions!

At the end of this posting you will find some explanation on the different parameters that allow you to fit the plotting results to your needs...

Plotting proportional tables: sjPlotPropTable.R

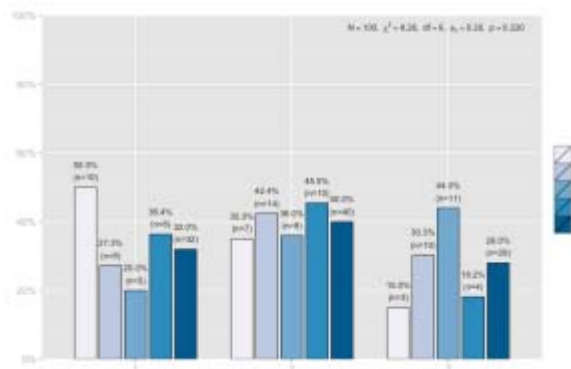
The idea for this function came up when I saw the distribution of categories (or factor levels) within one group or variable, that sum up to 100% – typically shown as stacked bars. So I wrote a script that shows the cross tabulation of two variables and either show column or row percentage calculations.

First, load the script and create two random variables:

```
1 | source("sjPlotPropTable.R")
2 | grp <- sample(1:4, 100, replace=T)
3 | y <- sample(1:3, 100, replace=T)
```

The simplest way to produce a plot is following (note that, due to random sampling, your plots may look different!):

```
1 | sjp.xtab(y, grp)
```



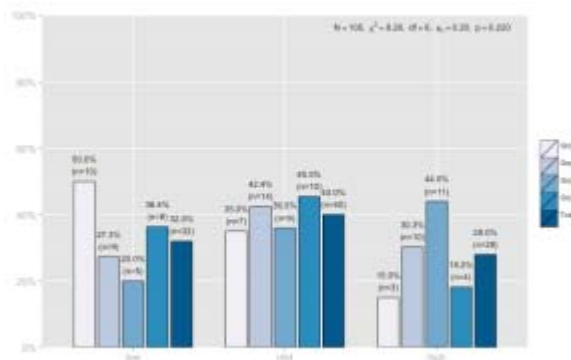
(<http://strengejacke.files.wordpress.com/2013/04>

/sjp_xtab_1.png)

Proportional table of two variables, column percentages, with "Total" column.

You can specify axis and legend labels:

```
1 | sjp.xtab(y, grp,  
2 |           axisLabels.x=c("low", "mid", "high"),  
3 |           legendLabels=c("Grp 1", "Grp 2", "Grp 3", "Grp 4"))
```



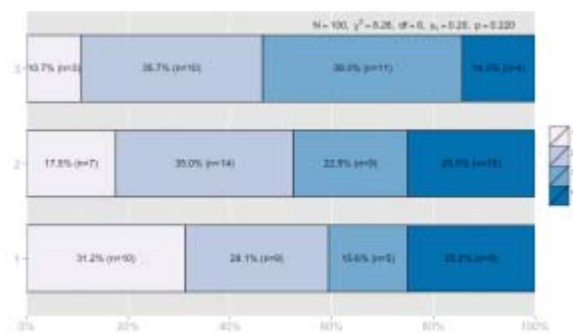
(<http://strengejacke.files.wordpress.com/2013/04>

/sjp_xtab_2.png)

Proportional table, column percentages, with assigned labels. The "Total" legend label is automatically added.

If you want row percentages, you can use stacked bars because each group sums up to 100%:

```
1 | sjp.xtab(y, grp,  
2 |           tableIndex="row",  
3 |           barPosition="stack",  
4 |           flipCoordinates=TRUE)
```



(<http://strengejacke.files.wordpress.com/2013/04>

/sjp_xtab_3.png)

Proportional table, stacked bars of row percentages,

Plotting correlations: sjPlotCorr.R

A very quick way of plotting a correlation heat map can be found [in this blog](http://martinsbioblogg.wordpress.com/2013/03/21/using-r-correlation-heatmap-with-ggplot2/) (<http://martinsbioblogg.wordpress.com/2013/03/21/using-r-correlation-heatmap-with-ggplot2/>). I had a similar idea in mind for some time and decided to write a small function that allows some tweaking of the produced plot like different colors indicating positive or negative correlations and so on.

Again, at first load the script and create a random sample:

```
1 | source("sjPlotCorr.R")
2 | df <- as.data.frame(cbind(rnorm(10),
3 |                             rnorm(10),
4 |                             rnorm(10),
5 |                             rnorm(10),
6 |                             rnorm(10)))
```

You can either pass a data frame as parameter or a computed correlation object as well. If you use a data frame, following correlation will be computed:

```
1 | cor(df, method="spearman"
2 |       use="pairwise.complete.obs")
```

The simple function call is:

```
1 | sjp.corr(df)
```



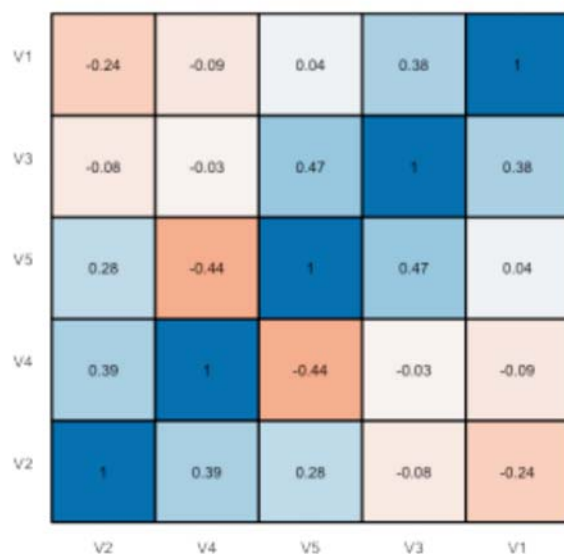
(<http://strengejacke.files.wordpress.com/2013/04>

/sjp_corr_1.png)

Correlation matrix of all variables in a data frame.

This gives you a correlation map with both circle size and color intensity indicating the strength of the correlations. You can also plot tiles, which looks more like a heat map, if you prefer:

```
1 | sjp.corr(df, type="tile", theme="none")
```



(<http://strengejacke.files.wordpress.com/2013/04>

/sjp_corr_2.png)

Tiled correlation matrix without background theme.

Plotting frequencies: sjPlotFrequencies.R

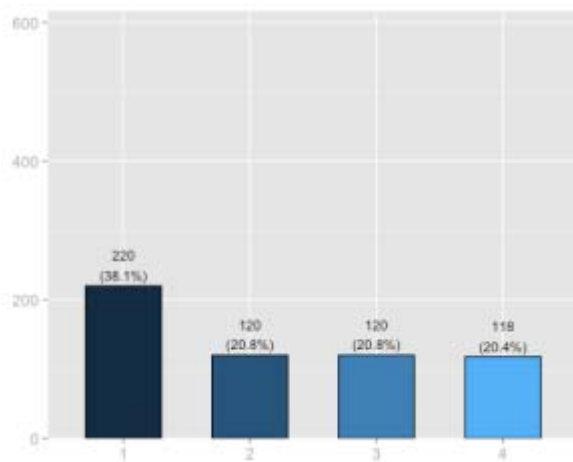
There is already a posting [which demonstrates this script \(http://strengejacke.wordpress.com/2013/02/25/simplify-frequency-plots-with-ggplot-in-r-rstats/\)](http://strengejacke.wordpress.com/2013/02/25/simplify-frequency-plots-with-ggplot-in-r-rstats/), however, since it uses a SPSS data set, I want to give short examples that run out of the box here.

Load the script:

```
1 | source("sjPlotFrequencies.R")
```

A simple bar chart:

```
1 | sjp.frq(ChickWeight$Diet)
```



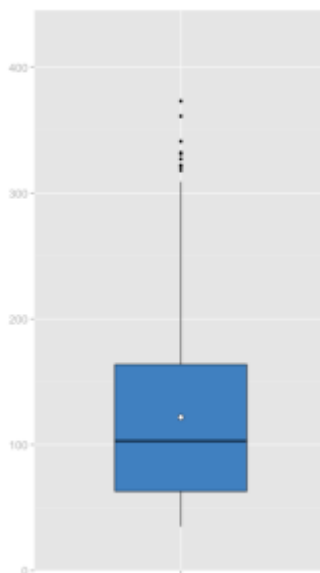
(http://strengejacke.files.wordpress.com/2013/04/sjp_freq_4.png)

/sjp_freq_4.png)

Simple bar chart of frequencies.

A box plot:

```
1 | sjp.frq(ChickWeight$weight, type="box")
```

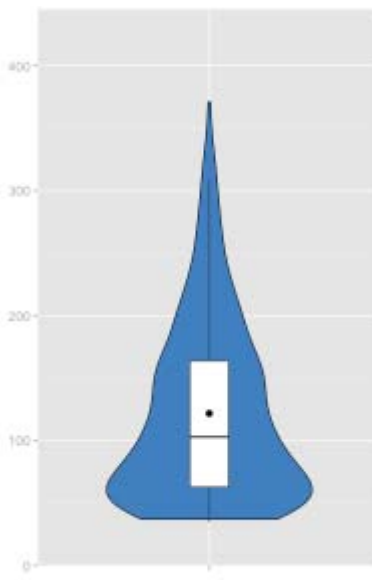


(http://strengejacke.files.wordpress.com/2013/04/sjp_freq_1.png)

A simple box plot with median and mean dot.

A violin plot:

```
1 | sjp.frq(ChickWeight$weight, type="v")
```

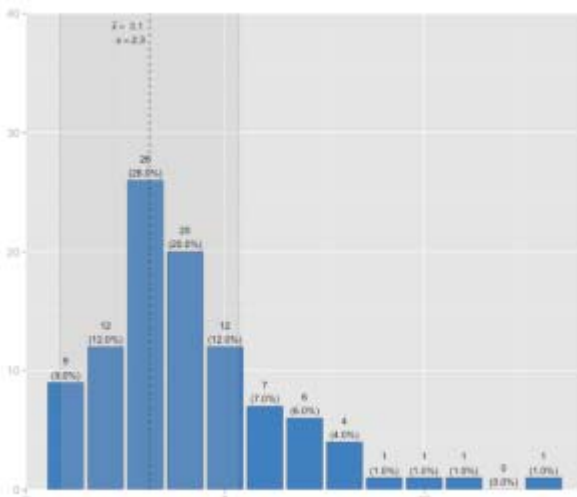


(http://strengejacke.files.wordpress.com/2013/04/sjp_freq_3.png)

A violin plot (density curve estimation) with box plot inside.

And finally, a histogram with mean and standard deviation:

```
1 | sjp.frq(discoveries, type="hist", showMeanIntercept=TRUE)
```



(<http://strengejacke.files.wordpress.com/2013/04>

[/sjp_freq_2.png](#))

Histogram with mean intercept and standard deviation range.

Plotting grouped frequencies: sjPlotGroupFrequencies.R

The grouped frequencies script has also been [described in a separate posting](#)

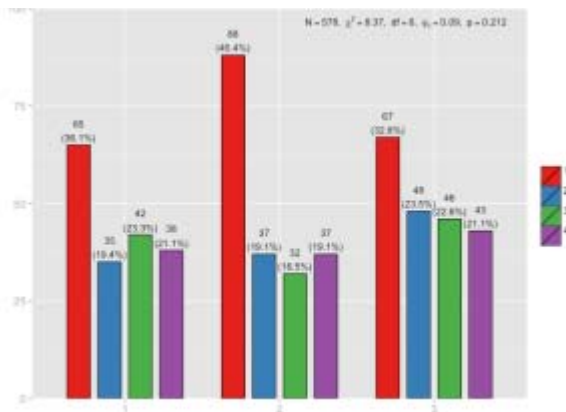
(<http://strengejacke.wordpress.com/2013/03/05/easily-plotting-grouped-bars-with-ggplot/>).

Load the script:

```
1 | source("sjPlotGroupFrequencies.R")
```

Grouped bars using the ChickenWeight data set. Note that due to random sampling, your figure may look different:

```
1 | sjp.grpfrq(sample(1:3, length(ChickWeight$Diet), replace=T),
2 |             as.numeric(ChickWeight$Diet),
3 |             barSpace=0.2)
```



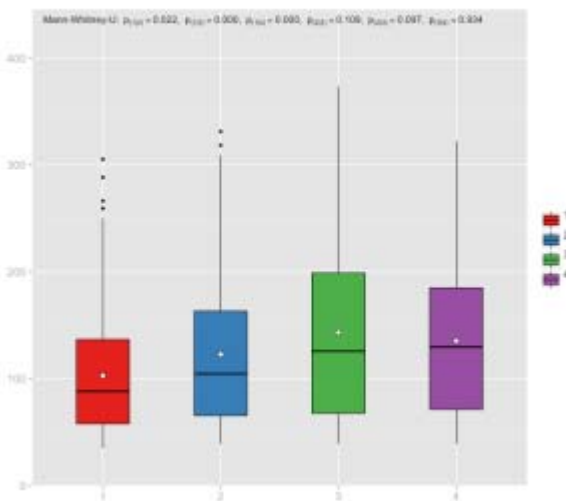
(<http://strengejacke.files.wordpress.com/2013/04>

/sjp_grfq_4.png)

Grouped bars with little bar spacing.

Grouped box plots. Note that this plot automatically computes the Mann-Whitney-U-test for the relation of each two subgroups. The tested groups are indicated by the subscripted numbers after the "p":

```
1 | sjp.grpfreq(ChickWeight$weight,
2 |             as.numeric(ChickWeight$Diet),
3 |             type="box")
```



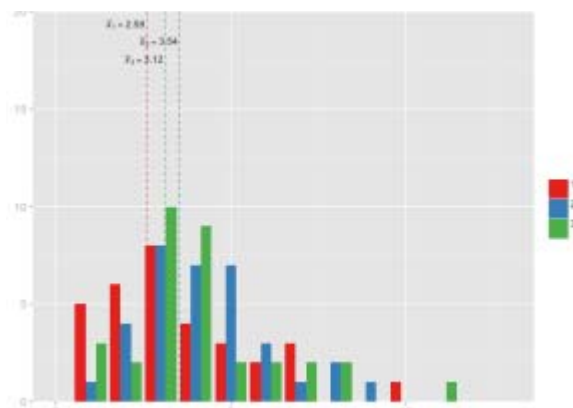
(<http://strengejacke.files.wordpress.com/2013/04>

/sjp_grfq_2.png)

Grouped box plots, showing the Weight distribution, divided into 4 random groups.

Grouped histogram:

```
1 | sjp.grpfreq(discoveries,
2 |             sample(1:3, length(discoveries), replace=T),
3 |             type="hist",
4 |             showValueLabels=FALSE,
5 |             showMeanIntercept=TRUE)
```



/sjp_gfrq_1.png)

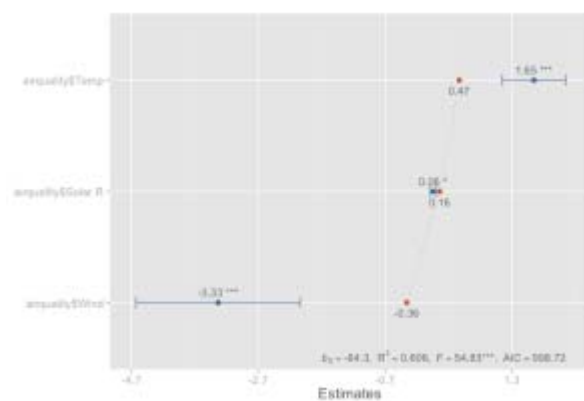
(<http://strengejacke.files.wordpress.com/2013/04>

Grouped histogram of "Discoveries", divided into three random subgroups, including mean intercepts for each group. Value labels for each bar are hidden.

Plotting linear model: sjPlotLinreg.R

Plotting (generalized) linear models have also already been described in a posting (<http://strengejacke.wordpress.com/2013/03/22/plotting-lm-and-glm-models-with-ggplot-rstats/>), so I will keep it short here and just give a running example:

```
1 source("sjPlotLinreg.R")
2 fit <- lm(airquality$Ozone ~ airquality$Wind +
3         airquality$Temp +
4         airquality$Solar.R)
5 sjp.lm(fit, gridBreaksAt=2)
```



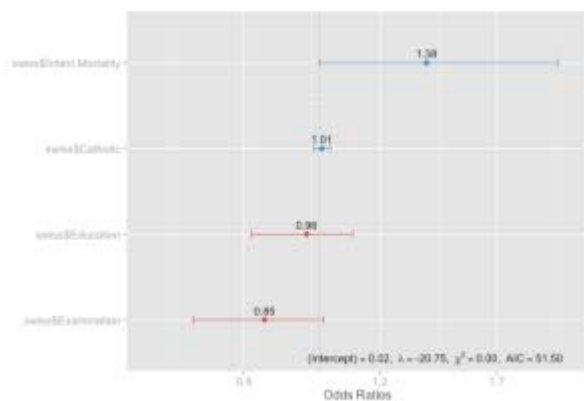
(<http://strengejacke.files.wordpress.com/2013/04>

/sjp_lm1.png)

Beta coefficients (blue) and standardized beta coefficients (red) from a linear model.

Plotting generalized linear models: sjPlotOdds.R

```
1 source("sjPlotOdds.R")
2 y <- ifelse(swiss$Fertility < median(swiss$Fertility), 0, 1)
3 fitOR <- glm(y ~ swiss$Examination +
4             swiss$Education +
5             swiss$Catholic +
6             swiss$Infant.Mortality,
7             family=binomial(link="logit"))
8 sjp.glm(fitOR, transformTicks=TRUE)
```

[/sjp_glm1.png](#)

(<http://strengejacke.files.wordpress.com/2013/04>

Odds ratios.

Which parameters can be changed?

There is, depending on the function, a long list of parameters that can be changed to tweak the figure you want to produce. If you use editors like **RStudio** (<http://www.rstudio.com/>), you can press ctrl+space inside a function call to access a list of all available parameters of a function. All available parameters are documented at the beginning of each script (and if not, please let me know so I can complete the documentation).

Three examples of what you can modify in your plot:

Labels

Axis labels can be changed with the `axisLabel.x` or `axisLabel.y` parameter, depending on where labels appear (for instance, if you have frequencies, you use the `.x`, if you plot linear models, you use `.y` to change the labels). The size and color of labels can be changed with `axisLabelSize` and `axisLabelColor`. **Value labels** (labels inside the diagram), however, are manipulated with `valueLabels`, `valueLabelSize` and `valueLabelColor`. The same pattern applies to **legend labels**.

Showing / hiding elements

Many labels, values or graphical elements can be shown or hidden. `showAxisLabels.x` shows/hides the variable labels on the x-axis. `showValueLabels` shows/hides the value labels inside a diagram etc.

Diagram type

With the `type` parameter you can specify the type of diagram. E.g. the `sjPlotFrequencies` offers histograms, bars, box plots etc. Just specify the desired type with this parameter.

Last remarks

In case you want to apply the above shown functions on your (imported) data sets, you also may find this posting helpful (<http://strengejacke.wordpress.com/2013/03/27/getting-the-data-for-the-sjplotting-functions-into-shape-rstats/>).

VERÖFFENTLICHT IN WE DO WIE DU | GETAGGT MIT DATA VISUALIZATION,
GGPLOT, R, RSTATS, STATISTIK

2 Gedanken zu “Examples for sjPlotting functions,

including correlations and proportional tables with ggplot #rstats”

1. *Joint_Posterior*

18. APRIL 2013 UM 14:49

Wow, great stuff! Now integrate something like colorbrewer or

<http://tools.medialab.sciences-po.fr/iwanthue/> and package it up.

Antwort

2. *Daniel*

18. APRIL 2013 UM 17:05

Well, color brewer is already “semi-implemented”. If you use

```
1 | barColor="brewer", colorPalette="<inserBrewerPalette>", ...
```

as parameter, you can use any of the available color brewer palettes.

And yes, I already thought about creating a package and did some tries, but failed to compile a running package... :-/ Perhaps I'll manage to create a package somewhen in the (hopefully near) future.

Antwort

Bloggen Sie auf WordPress.com. Theme: Forever von Automattic.