

Get the Most out of your Data by the Power of R Graphics

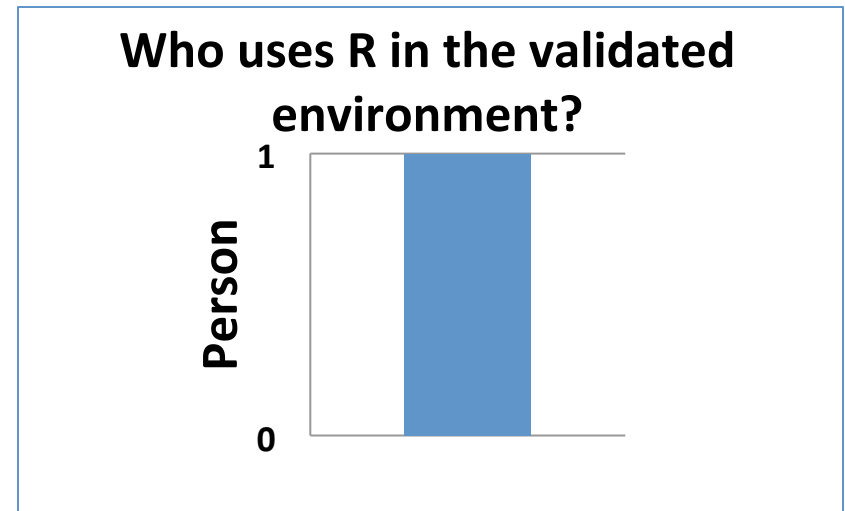
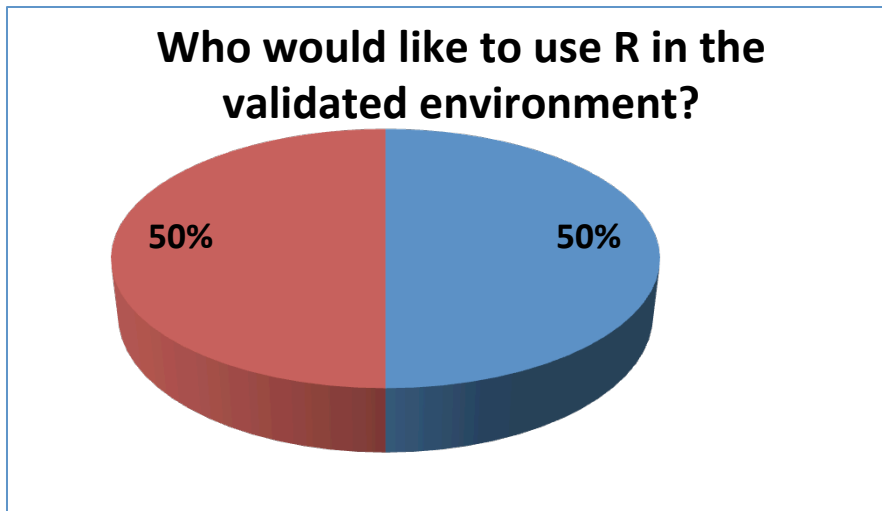
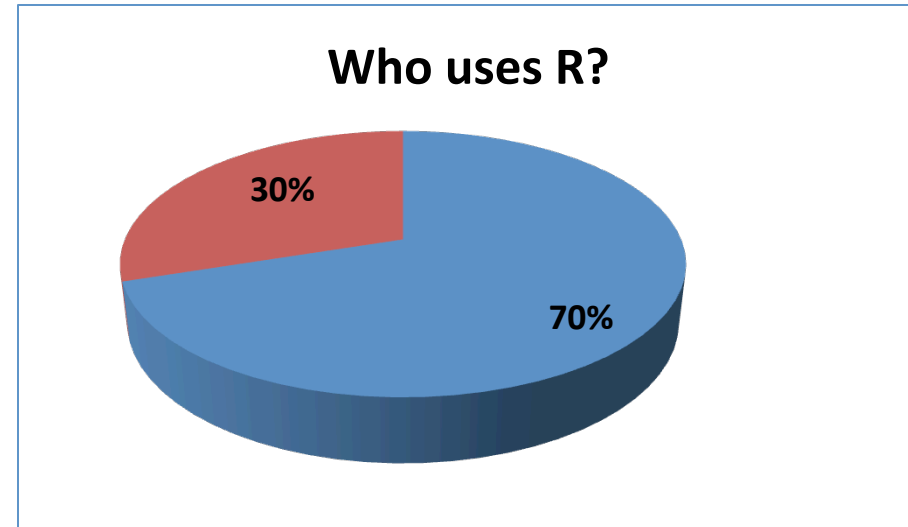
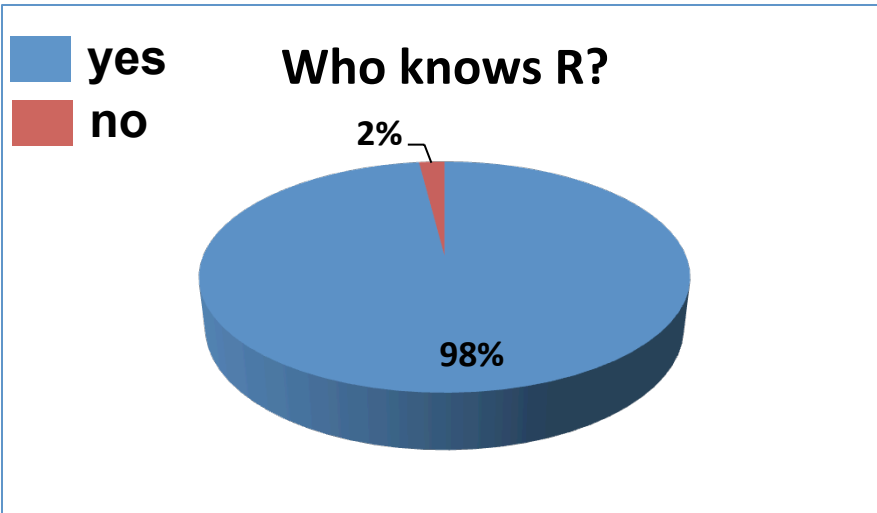
PhUSE-Conference 14.10.2013

Company

- HMS Analytical Software is a specialist for Information Technology in the field of Data Analysis and Business Intelligence Systems
- Profile
 - 50 employees in Heidelberg, Germany
 - SAS Institute Partner for 15 years
 - Doing data oriented software projects for more than 20 years
 - Focus on life science industry
- Technologies
 - Analytics and Data Management: SAS, JMP, R, Microsoft SQL Server
 - Application Development: Microsoft .NET, Java, Open Source



Degree of Popularity of R since PhUSE 2010



Understand the Difference: Concepts Behind R

- 1993 first occurrence in a mailing list^[1]
- Open Source Software under the General Public License (GPL)
 - freely available
 - around 2 million users^[2]
- Focus:
 - statistical computing
 - graphics
- The R Foundation for Statistical Computing
 - official voice of the R project
 - non-profit organization seated in Vienna, Austria
 - consists of ordinary and supporting members, mainly the R Development Core-Team
 - holds and administers the copyrights of the R software and documentation.
- Huge amount of freely available additional packages
 - not developed by the R Development Core-Team but by programmers from throughout the world



What happened so far...

- *How much is it? Validation of Open-Source-Software using the example of R, PhUSE 2010, Berlin*
- *SAS and R - The odd Couple, PhUSE Single Day Event 2011, Frankfurt*
- *Workshop: Getting Started with R for SAS Programmers, PhUSE 2012, Budapest*



It's Open Source...



Thousands of Add-On packages

Why should one use R?

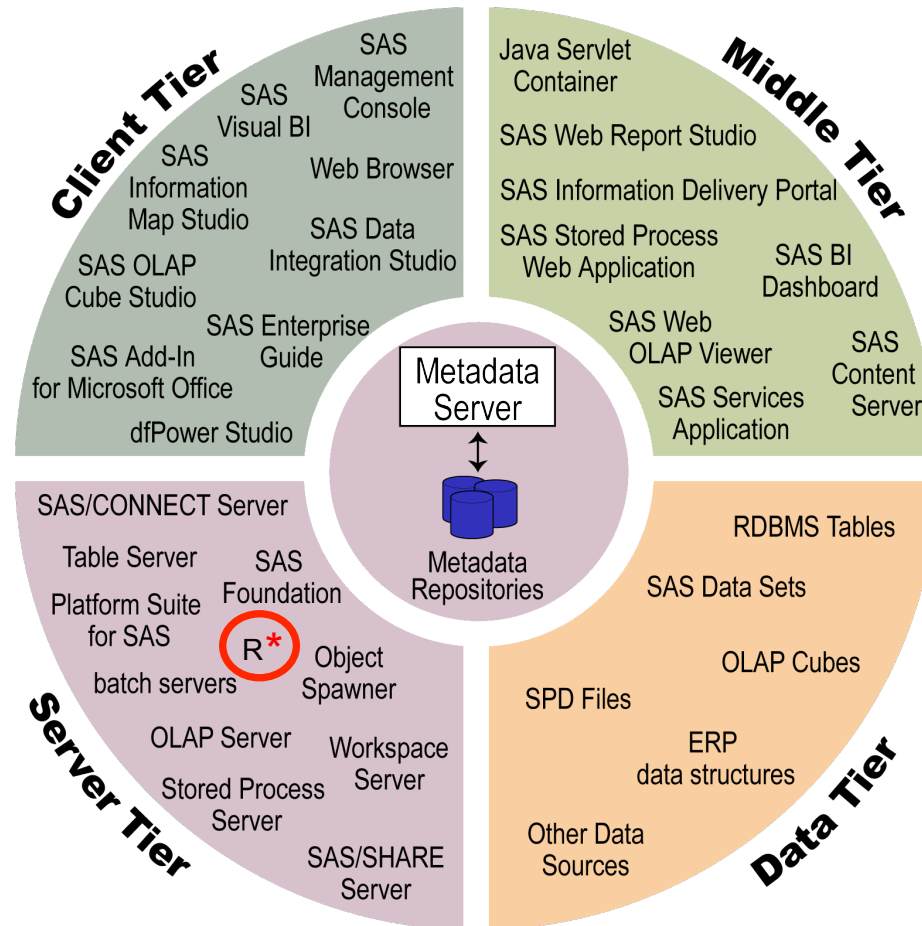
It's free... ääh...yes...it's absolutely FREE!



It has State-of-the-Art-Algorithms...



R as a SAS Product...



*** This is just a joke**

One Volunteer is needed

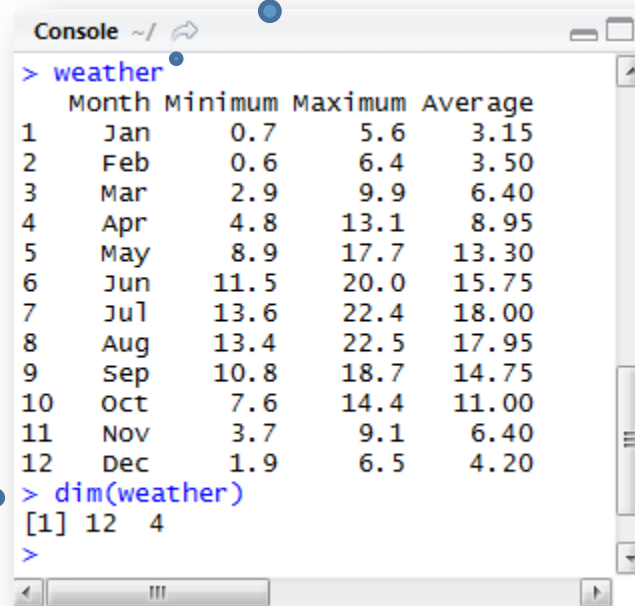


Draw the weather in Brussels 2012 by Hand



Draw the weather in Brussels 2012

it's an object of
type data.frame



```
> weather
  Month Minimum Maximum Average
1   Jan     0.7      5.6     3.15
2   Feb     0.6      6.4     3.50
3   Mar     2.9      9.9     6.40
4   Apr     4.8     13.1     8.95
5   May     8.9     17.7    13.30
6   Jun    11.5     20.0    15.75
7   Jul    13.6     22.4    18.00
8   Aug    13.4     22.5    17.95
9   Sep    10.8     18.7    14.75
10  Oct     7.6     14.4    11.00
11  Nov     3.7      9.1     6.40
12  Dec     1.9      6.5     4.20

> dim(weather)
[1] 12  4

>
```

The screenshot shows a R console window with the following content:

	Month	Minimum	Maximum	Average
1	Jan	0.7	5.6	3.15
2	Feb	0.6	6.4	3.50
3	Mar	2.9	9.9	6.40
4	Apr	4.8	13.1	8.95
5	May	8.9	17.7	13.30
6	Jun	11.5	20.0	15.75
7	Jul	13.6	22.4	18.00
8	Aug	13.4	22.5	17.95
9	Sep	10.8	18.7	14.75
10	Oct	7.6	14.4	11.00
11	Nov	3.7	9.1	6.40
12	Dec	1.9	6.5	4.20

Below the table, the dimensions of the data frame are shown:

```
> dim(weather)
[1] 12  4
```

has 12
columns and
4 rows

Draw the weather in Brussels 2012

```
PhUSE_2013.R* x
Source
6 plot(
7
8   x = weather$Average
9
10  , xaxt = 'n'
11
12  , yaxt = 'n'
13
14  , type = "p"
15
16  , main = "weather in Brussels"
17
18  , xlab = "Months"
19
20  , ylab = "Temperature [°C]"
21
22  , xlim = c(1, 12)
23
24  , ylim = c(0, 20)
25
26 )
27
28
29 axis(2, at = 1:12, labels= weather$Month)
30
31
```

plot function

data column

plot no x-axis labels

plot no y-axis labels

plot data as points

header

labels x-axis

labels y-axis

x-axis range

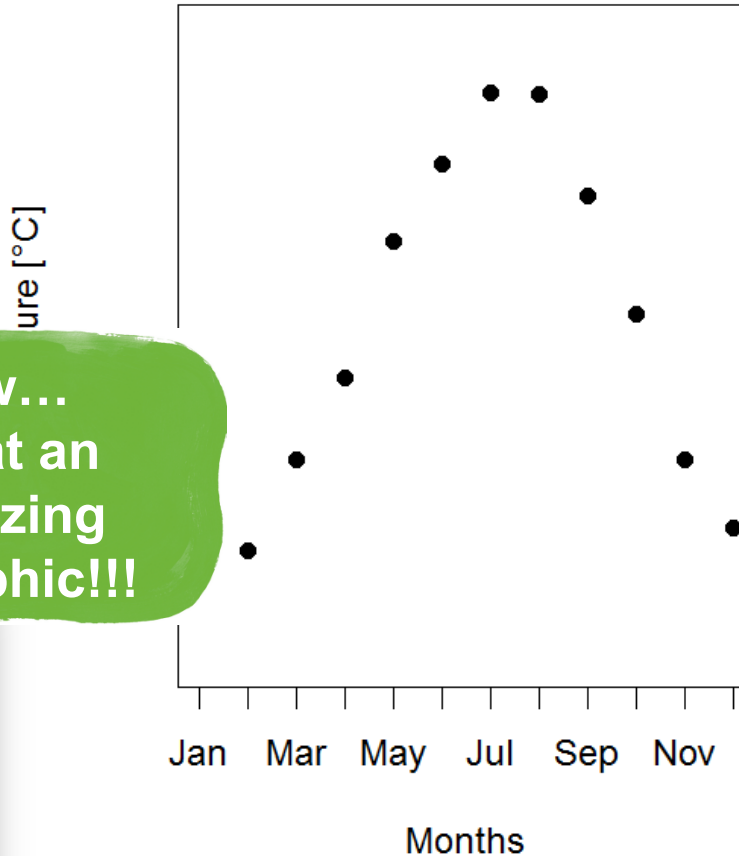
y-axis range

the end

add axis labels

Draw the weather in Brussels 2012

Weather in Brussels



Wow...
What an
amazing
graphic!!!



But...
Keep basics in mind

```
plot(
  x = weather$Average, data column
  xaxt = "n", plot no x-axis labels
  yaxt = "n", plot no y-axis labels
  type = "p", plot data as points
  main = "weather in Brussels", header
  xlab = "Months", labels x-axis
  ylab = "Temperature [°C]", labels y-axis
  xlim = c(1, 12), x-axis range
  ylim = c(0, 20), y-axis range
  ) the end
axis(2, at = 1:12, labels= weather$Month) add axis labels
```

Combine Graphics and Statistics

```

57
58 pop.mod = nls( population ~ beta1/(1 + exp(beta2 + beta3*time)),
59               , start=list(beta1 = 350, beta2 = 4.5, beta3 = -0.3)
60               , trace=T)
61
62 summary(pop.mod)
63
64 plot(year, population, pch = 16)
65
66 lines(year, fitted.values(pop.mod), lwd=2, col = "red")
67
68
68:1 # (Untitled)
R Script

```

nonlinear least-squares
estimates of
a nonlinear model.

```

Console ~/
>
> summary(pop.mod)

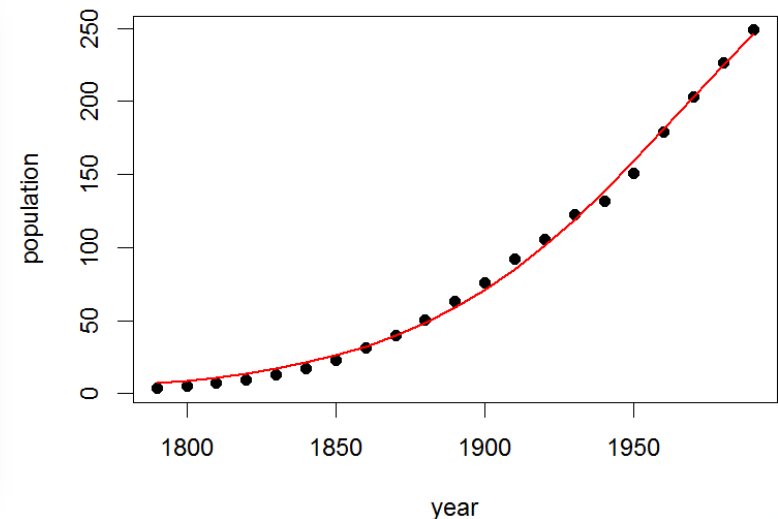
Formula: population ~ beta1/(1 + exp(beta2 + beta3 * time))

Parameters:
      Estimate Std. Error t value Pr(>|t|)
beta1  389.16551   30.81196   12.63  2.2e-10 ***
beta2    3.99035    0.07032   56.74  < 2e-16 ***
beta3   -0.22662    0.01086  -20.87  4.6e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.45 on 18 degrees of freedom

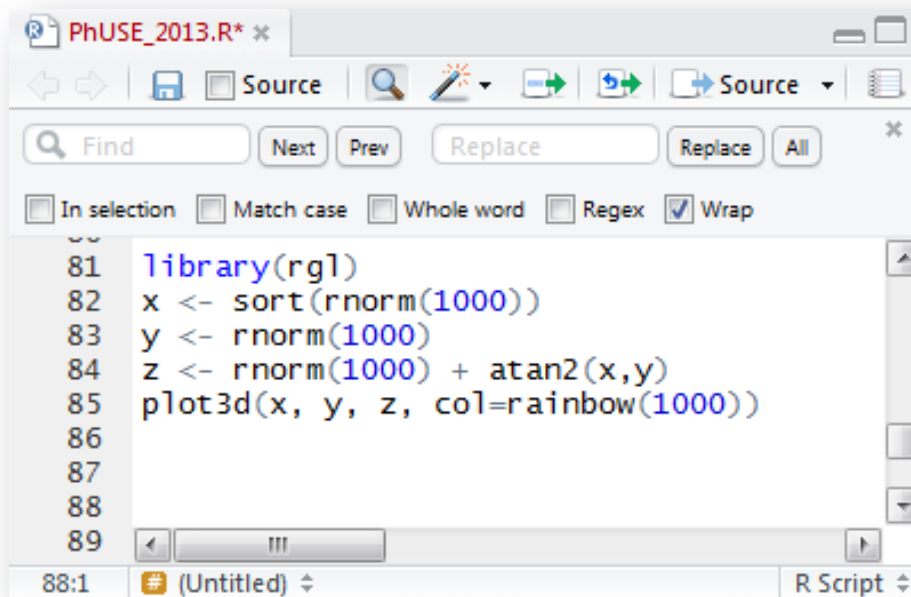
Number of iterations to convergence: 6
Achieved convergence tolerance: 1.504e-06

```

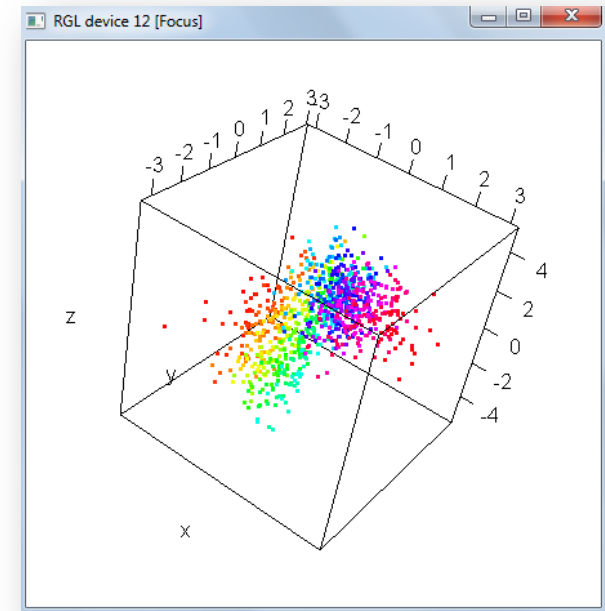


Three Dimensional Plots

- `install.packages("rgl")`

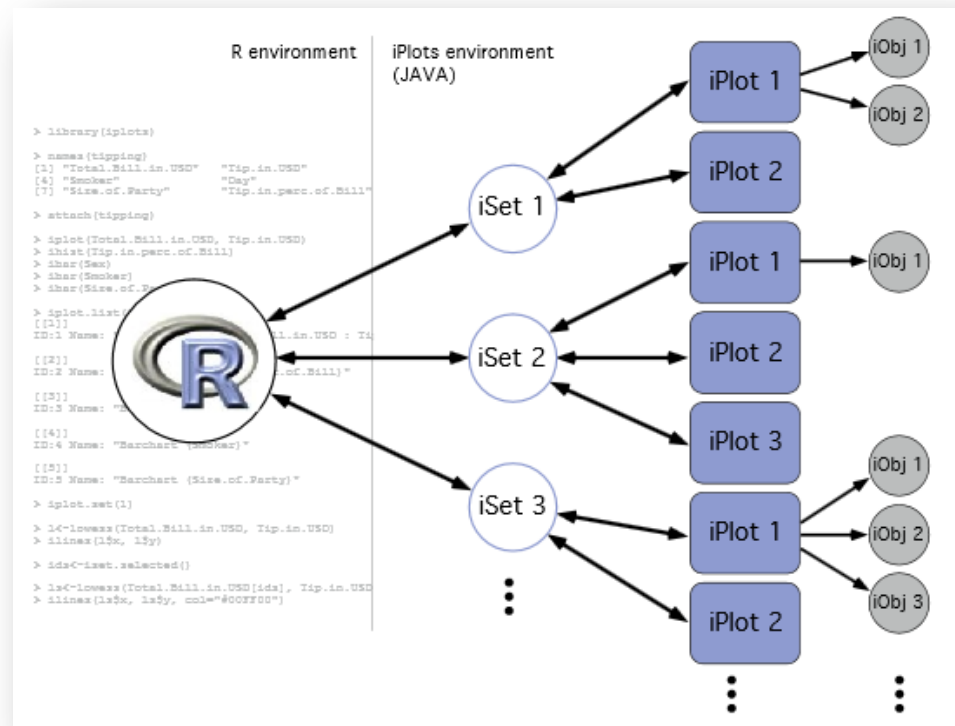


```
81 library(rgl)
82 x <- sort(rnorm(1000))
83 y <- rnorm(1000)
84 z <- rnorm(1000) + atan2(x,y)
85 plot3d(x, y, z, col=rainbow(1000))
86
87
88
89
```



Interactive Plots

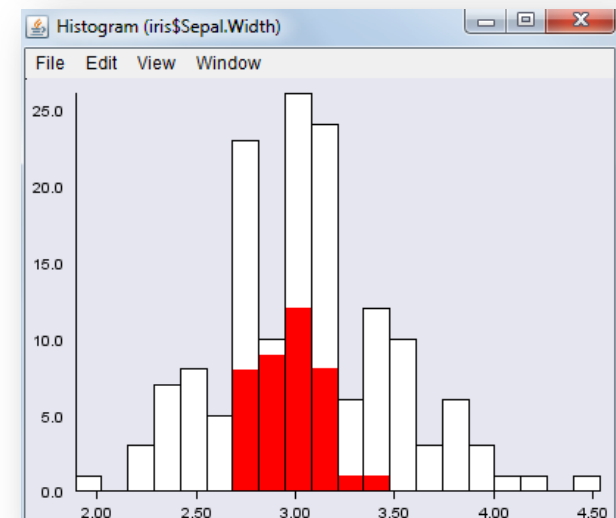
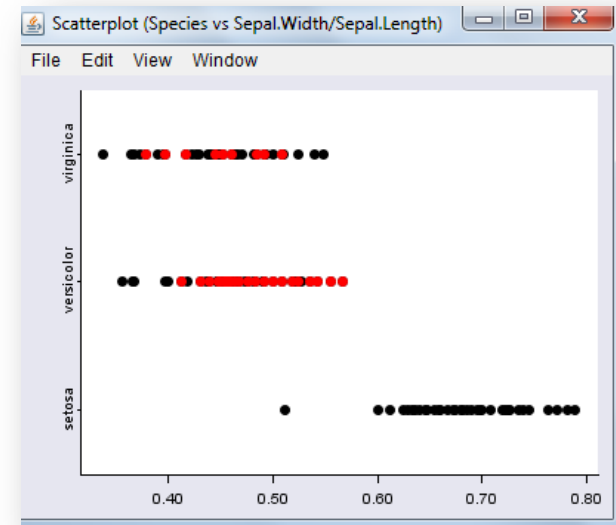
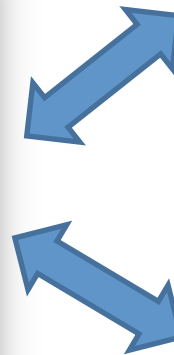
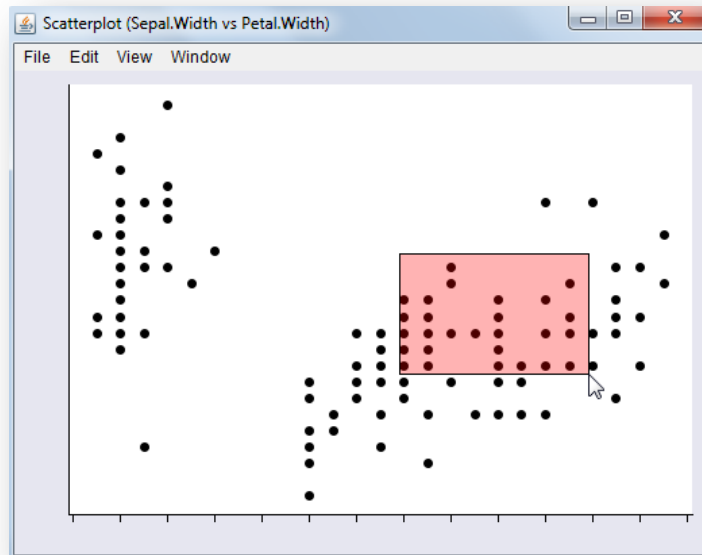
- `install.packages("iplots")`



[iPlots. High Interaction Graphics for R.](#) Simon Urbanek, Martin Theus

Interactive Plots

- `install.packages("iplots")`



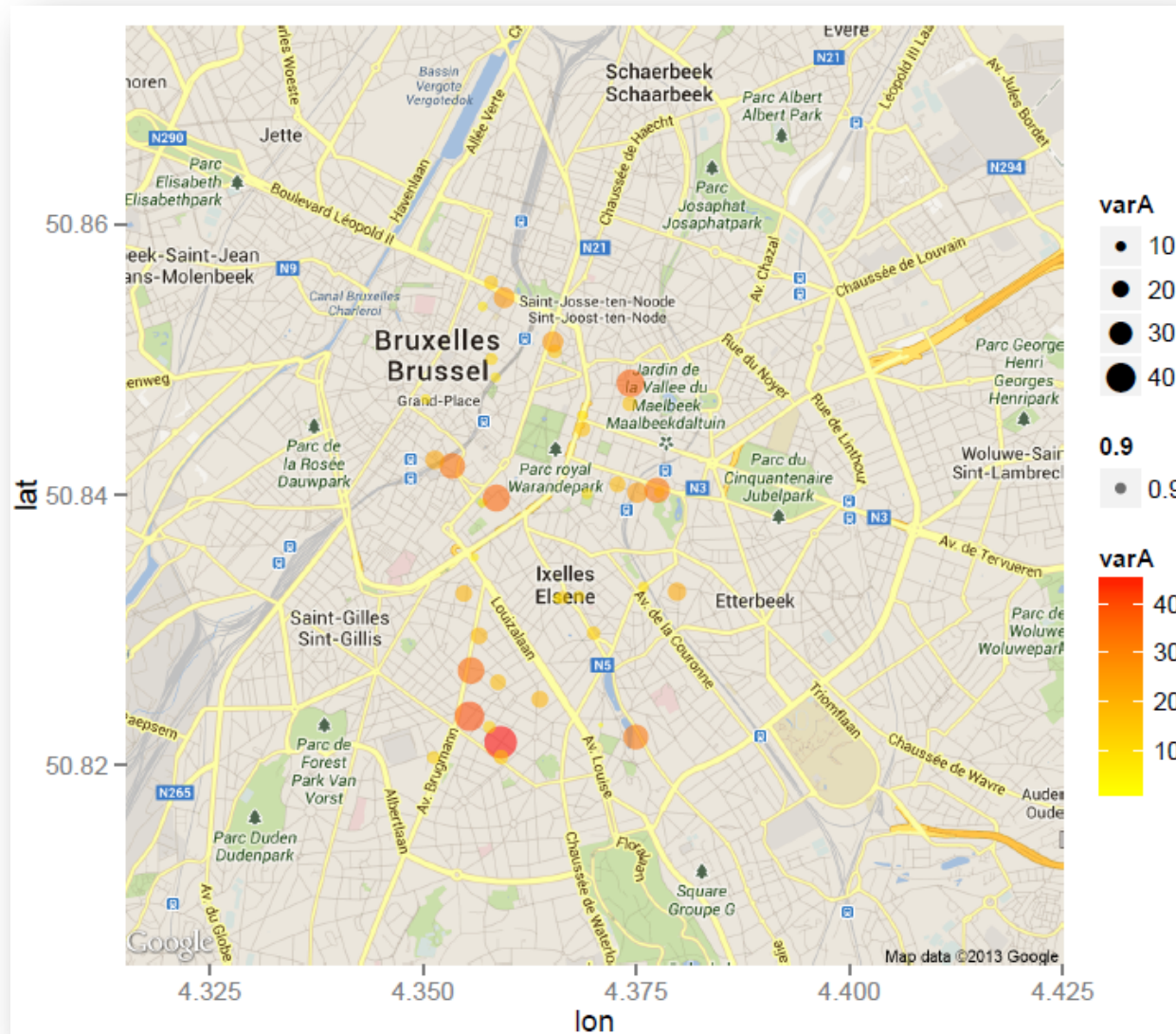
Saving Graphics

```
134 pdf(file = "c:/temp/Brussel_Google.pdf", bg = "transparent")
135
136
137 library(grid)
138 library(ggmap)
139
140 set.seed(1234)
141 lon <- runif (40, 4.35, 4.38)
142 lat <- runif (40, 50.82, 50.86)
143 varA = rnorm (40, 20, 10)
144 myd <- data.frame (lon, lat, varA)
145 brusselsMap = get_map(location = c(lon = 4.37 , lat = 50.84 )
146                      , zoom = 13, maptype = 'roadmap'
147                      , source = "google")
148 brusselsMap = ggmap(brusselsMap)
149 brusselsMap + geom_point(aes(x = lon, y = lat, colour = varA
150                             , size = varA, alpha = 0.9)
151                          , data = myd) + scale_colour_gradient(low="yellow", high="red")
152
153 dev.off()
154
155
```

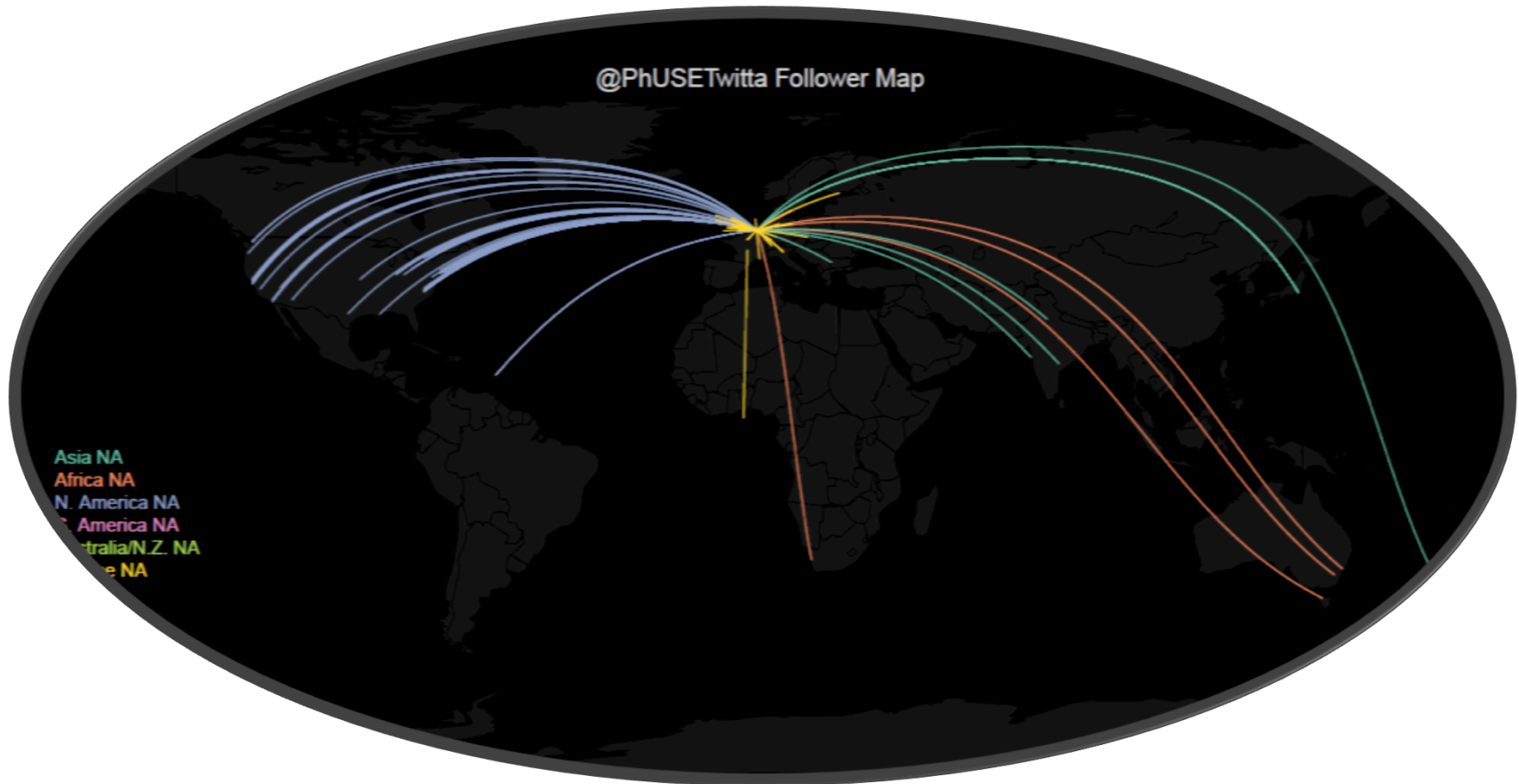
starts the graphics device drive

closes the graphics device drive

Saving Graphics

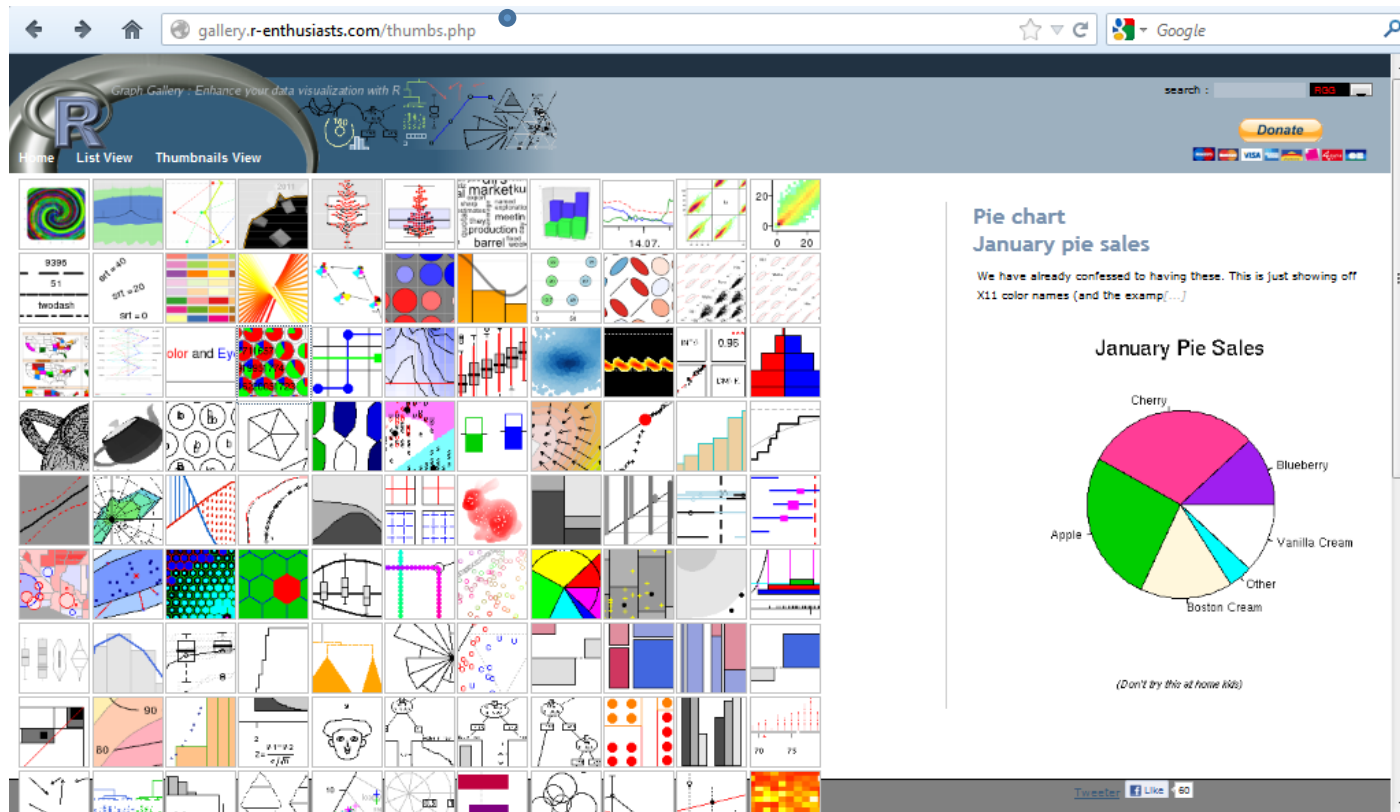


Who follows Theo PhUSE@Twitter?



R graphics gallery

*Learn from
the experts*



<http://gallery.r-enthusiasts.com/thumbs.php>

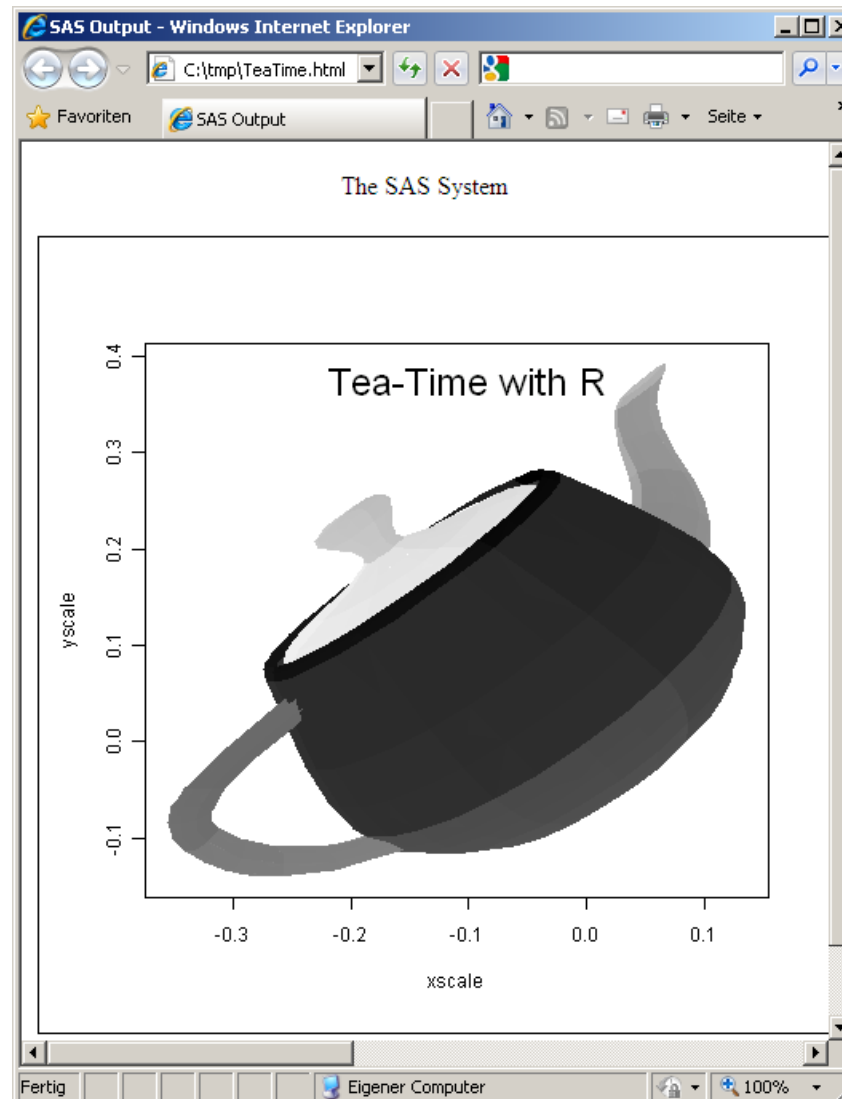
Get Graphics from R to SAS

```
PROC IML;
    SUBMIT / R;
        png(file ="c:/tmp/TeaTime.png"
            ,bg ="transparent")
            <R Code for a graphic>
        dev.off()
    ENDSUBMIT;
QUIT;

ODS ESCAPECHAR='^';
ODS HTML FILE='c:\tmp\TeaTime.html' STYLE=minimal
GPATH='c:\temp\' GTITLE GFOOTNOTE;

DATA _NULL_;
    FILE PRINT;
    PUT "<IMG SRC='c:\tmp\TeaTime.png' BORDER='0'>";
RUN;
```

Get Graphics from R to SAS



Literature



- Books:
 - [Introductory Statistics with R](#)
 - [R for SAS and SPSS Users](#)
 - [Lattice - Multivariate data visualization with R](#)
 - Blocks
 - <http://sas-and-r.blogspot.de/>
 - <http://www.r-bloggers.com/>
- Others:
- <http://r4stats.com/>, Robert A. Muenchen
 - [SAS 9.3 Interface to R](#)

Thank you for your Attention

Dr. Peter Bowerunge

Project Leader and Software Engineer

HMS Analytical Software GmbH

Rohrbacher Str. 26 • 69115 Heidelberg

Telefon +49 6221 6051-0

Peter.Bowerunge@analytical-software.de

www.analytical-software.de