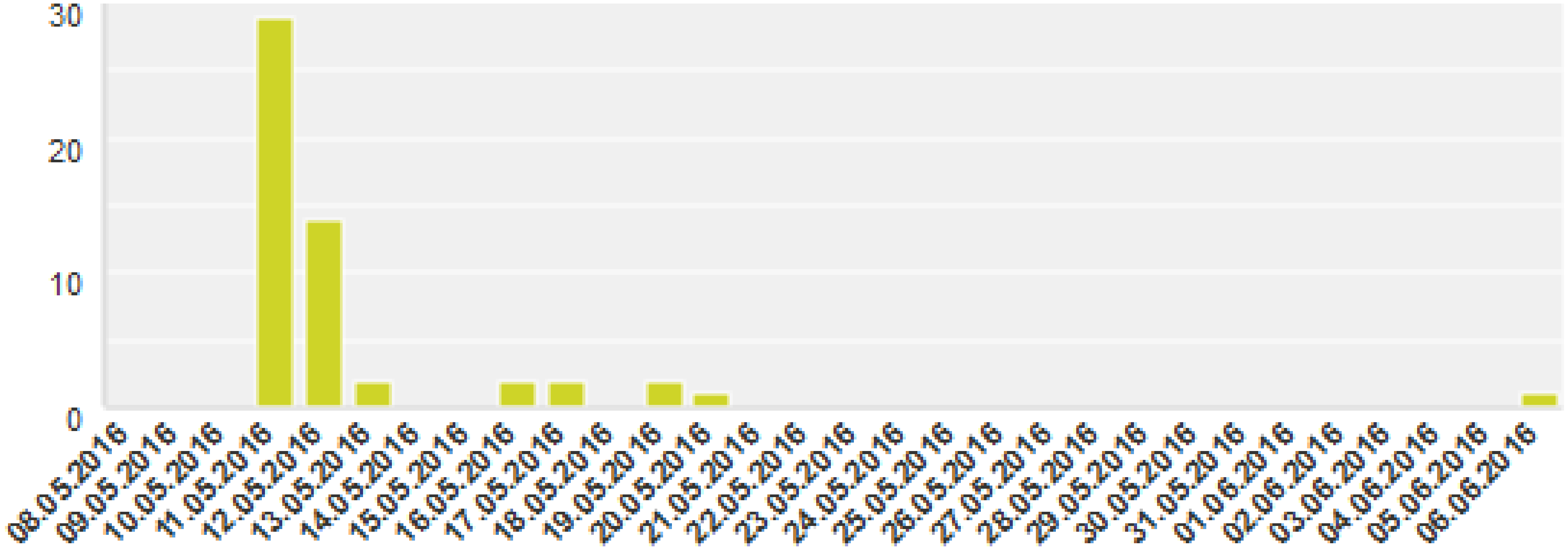
Berlin R User Group

Meetup
June 8, 2016

Berry Boessenkool
berry-b@gmx.de

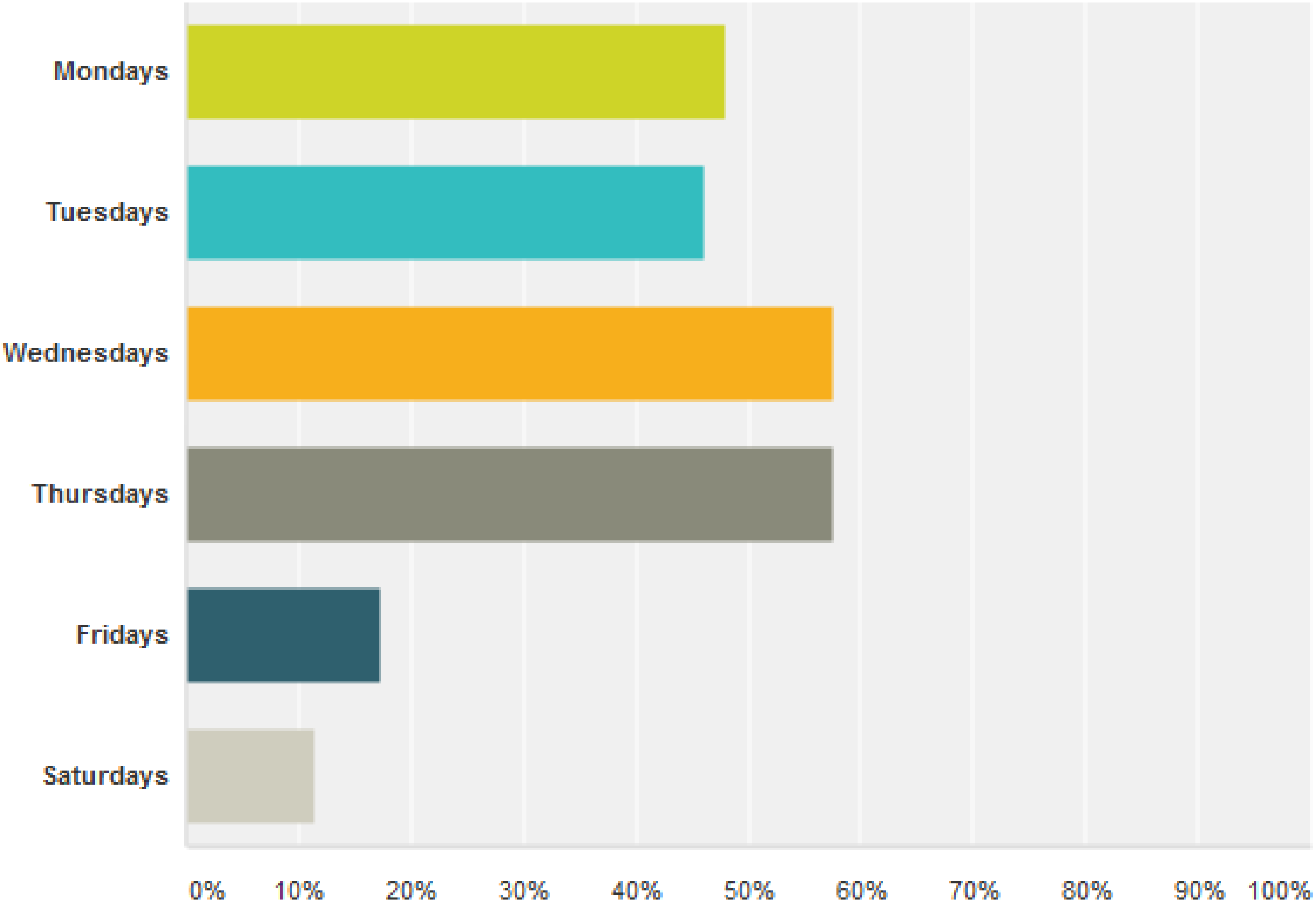
- Survey results
- Performant coding
  - for-loops
  - lapply
  - **pb**lapply, **mc**lapply
  - `Rcpp`

Member Survey May 2016: 53 Responses



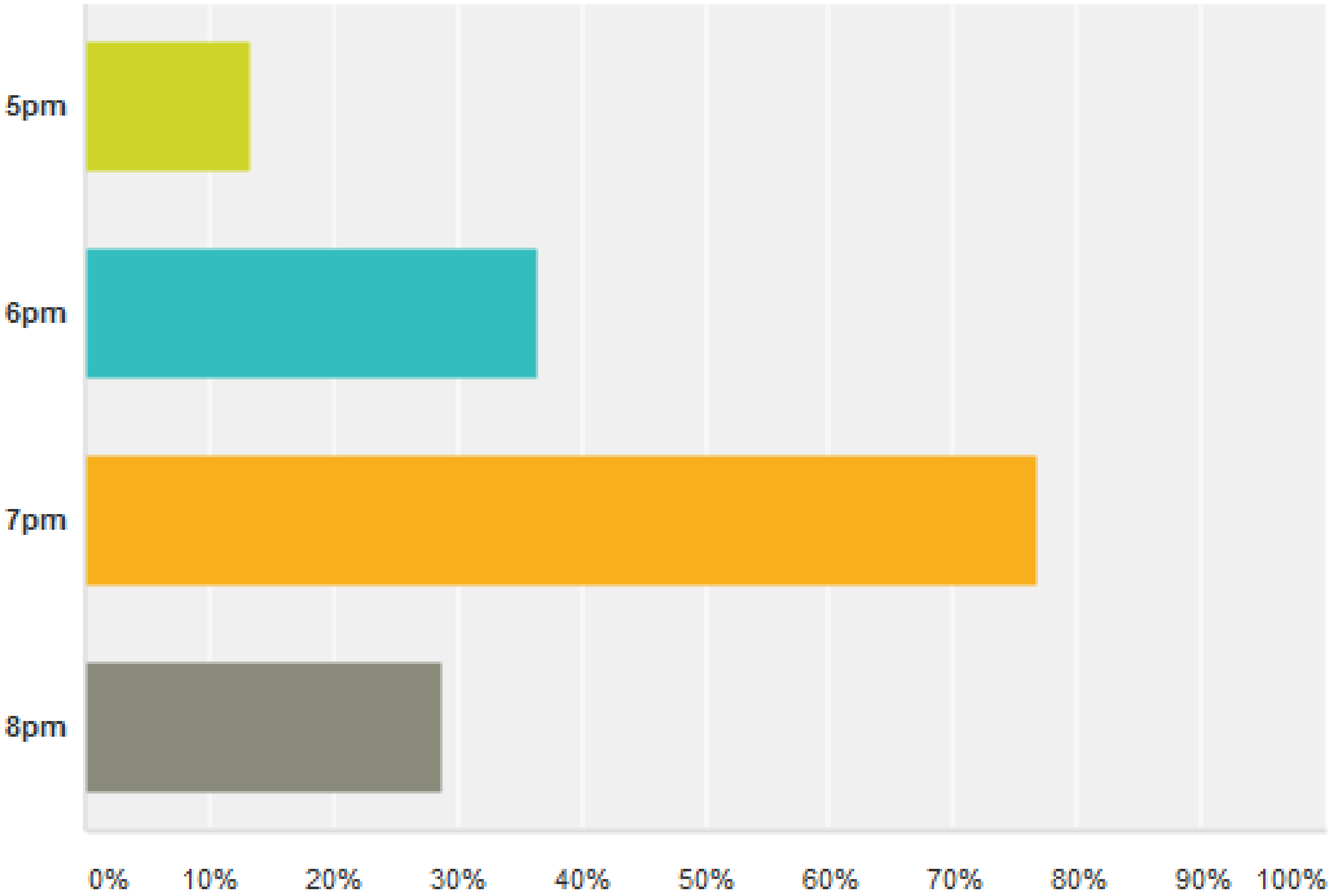
I would like to have our meetups on:

Beantwortet: 52 Übersprungen: 1



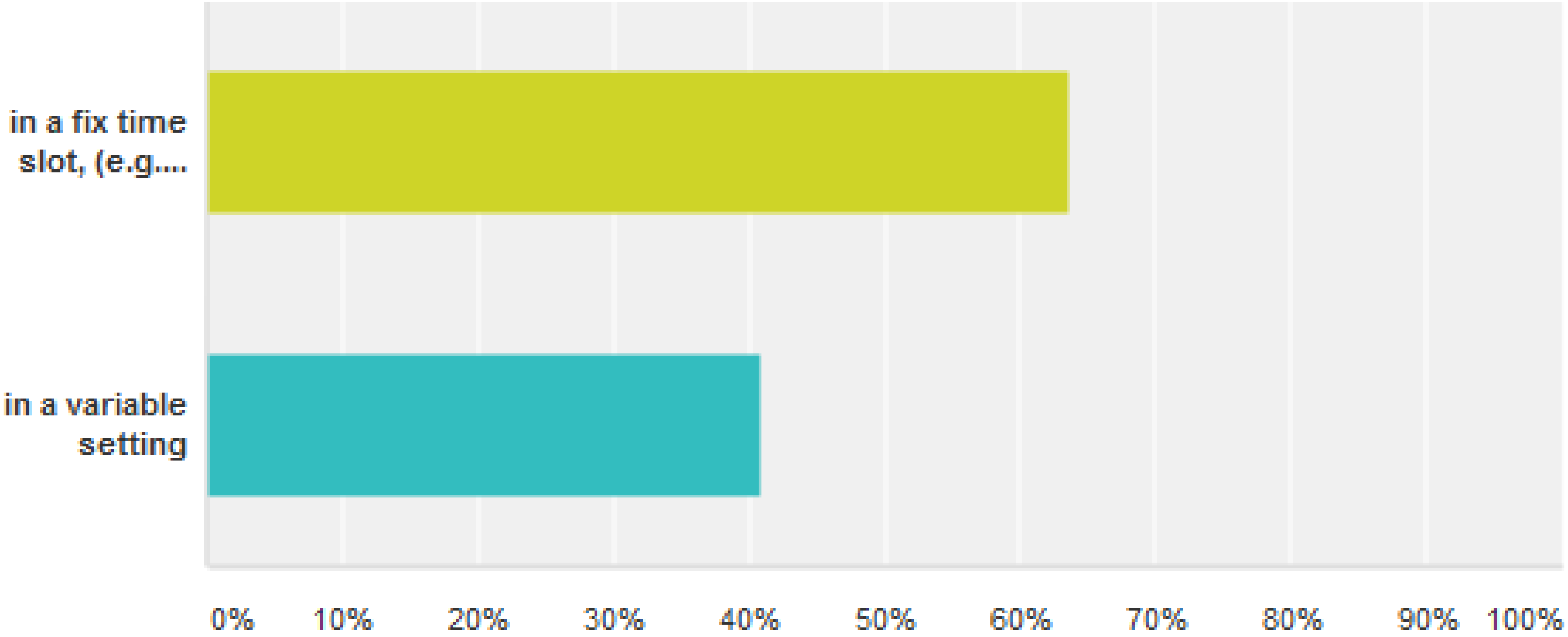
I would like to have our meetups start at:

Beantwortet: 52   Übersprungen: 1



I would rather have our meetups

Beantwortet: 44 Übersprungen: 9



I work / Research in

Science/Math	22
Engineering/computer science	14
Social sciences/psychology	12
Business / Finance / Marketing	12
Health sciences	11
Agriculture/Environmental studies	6
Communications/journalism	2
Education	2
Humanities	2
Performing and Fine Arts	1
Architecture	0
General studies	0
Law	0
Military/naval science	0

I am familiar with R possibilities in the following domains:

(non)linear regression	34
machine learning	29
hypothesis tests	28
classification algorithms	26
time series	20
distribution functions	17
bayesian stats	15
spatial analysis, mapping, geostatistics, GIS, etc	10
optimization	10
econometrics, finance	8

data processing reporting, dashboard (rmarkdown, shiny...) graphics
networks
item response theory
Data management
graph theory (igraph, RBGL)

I am familiar with R techniques related to:

programming conditions, loops, functions	42
computationally efficient programming (apply, tapply, lapply/sapply, replicate, etc)	40
graphics (plotting publication-ready visualisations)	33
character string operations	27
Rmd, knitR, sweave etc	25
package development	18
interactive plots, tcltk, shiny, etc	11

webserver integration
Not really, yet I would like to learn :-) Just have basic knowledge

I would like someone to present something on:

- computationally efficient programming, code optimization
- Simulation and Bayesian optimization
- Deep learning Mixed models
- starting understanding R
- interactive graphics
- Out of memory handling of large datasets
- Rccp
- Graphics, text mining, efficient programming, package development
- * graphics done right, presenting data in visually appealing way * extracting non-trivial information from series of events * integrating R with smth (preferably insane)
- bayesian statistics, mcmc
- time series analysis, (choice based) conjoint analysis, recommender systems, survival analysis, mixed effects models, neural networks
- Using R in production Shiny The "good parts" of R (à la Douglas Crockford)
- Bayesian stats in R, Probabilistic graphical models, time series, global optimization , efficient R usage and advanced programming techniques.
- Rcpp
- knitr, how to build reports
- time series analysis multilevel modeling visualisation computationally efficient programming machine learning
- Advanced classification, advanced machine learning, natural language processing, provenance (keeping data, code, and reports in sync)
- *S.P., per mail:* I'd really like someone to give a talk on MCMC in R starting from scratch. I'm interesting in learning how to perform MCMC AND to perform diagnostics on my chains (how do I know if they're well mixed, converged, etc.?)...

I could present on:

- R and Docker
- Developing SPSS extensions based on R-packages.
- graphics subsystems
- Caret, basic machine learning
- knitr (although rather basic stuff)
- Using R for environmetal/hydrological studies
- a website i built with r/shiny to compare data collected with different questionnaires
- efficient data wrangling with data.table - bayesian stats with rstan (or rstanarm) - choice modeling (= multiclass classification) with mlogit
- Psychological topics, Market research topics
- Packages I wrote. Package development. Programming in R: S3, S4, OOP, FP. Topics related to R in business and production systems.
- not sure if I am ready to present. I could do a very basic workshop on correlation and linear regression (what is it? what does it mean? and on how to do it in R obviously, what you have to pay attention to in the output, what the numbers mean etc.). If that's not too basic but I guess not everyone has a stats background, so maybe its useful. I have a PhD in psychology/neuroscience.
- creative data frame operations using Hadley's packages (dplyr, tidyr, tibble), visualizations (ggplot2, shiny), creating reports with Rmarkdown, R package development, microarray analysis and other packages from Bioconductor

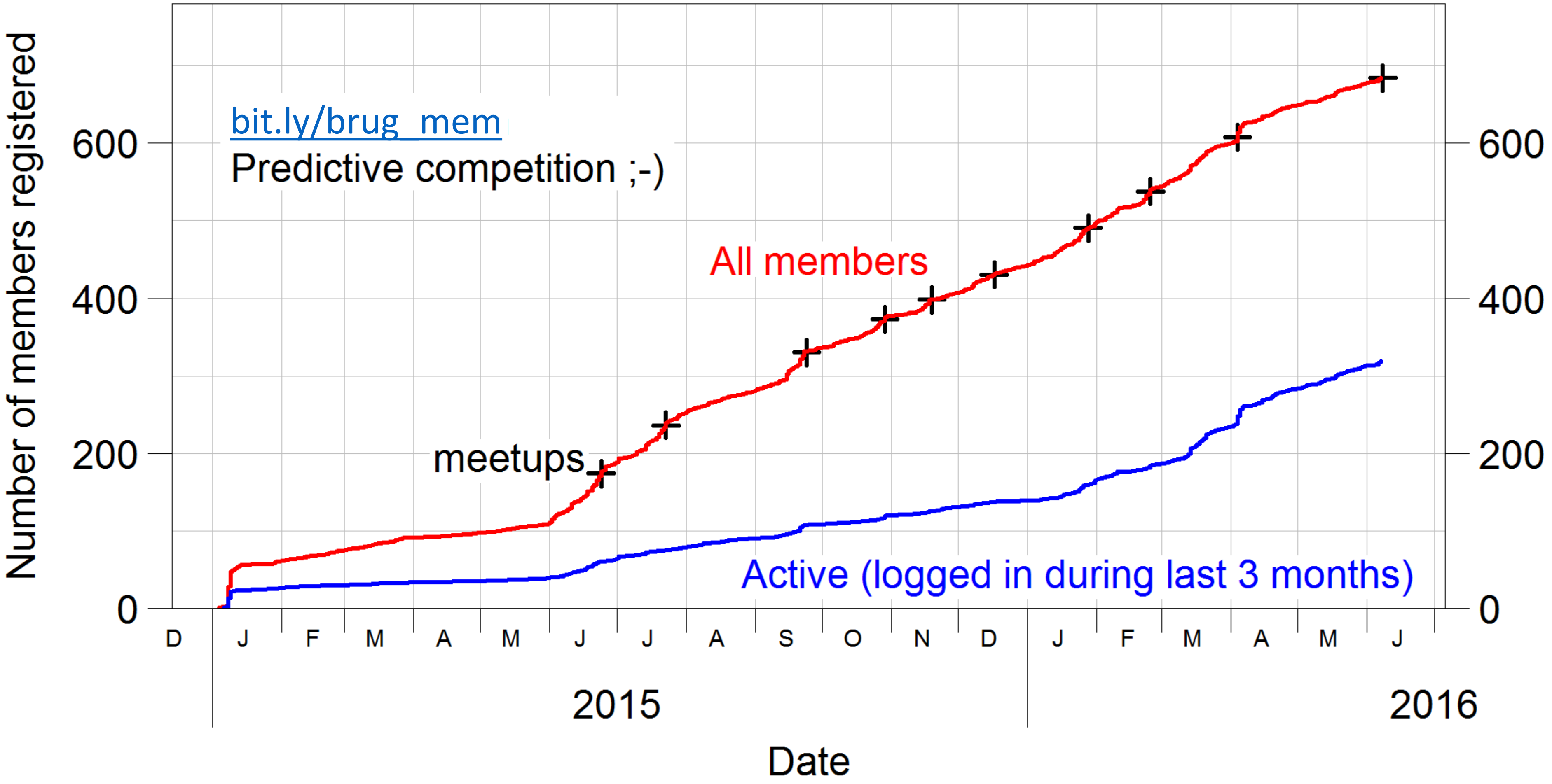
Other comments:

- Some of the Meetups I've attended had a very poor speaker quality. Also, it appeared to be an excuse for many companies/one-man persons self-promoting.
- keep up the good job, fellows!
- It would be great to offer the formation of learning groups
- Thanks for organising!!
- Since I work a lot with Python, too, I can also talk a bit about the Jupyter Notebook, rpy2, and I could prepare something about the new data exchange format "feather".

Future Topics (distilled):

- Rcpp
- Interactive graphics (`shiny`)
- MCMC (chains well mixed? converged?)
- Bayesian stats in R (`rstan`)
- Report / slides generation (`knitr, rmarkdown`)
- Time series analysis
- Large dataframes (`data.table vs dplyr, tidyr, tibble`)
- Jupyter notebooks: Julia, Python, R (`rpy2`)
- Programming: S3, S4, OOP, FP, R6
- Machine learning (`caret`)
- Package development (`devtools, roxygen2`)
- Spark, Hadoop, Amazone Azure, Docker (envisioned: July 6th)
- …

BERLIN-RUG member development 2016-06-08



bit.ly/brug_mem

Predictive competition ;-)

for-loops, lapply, Rcpp – motivational example

```
files <- dir(pattern="*.csv")

# bad and slow way:
dfs <- list() # initiate empty list
for(i in 1:length(files))
    dfs[[i]] <- read.csv(files[i], as.is=TRUE)

# much better way: apply function to each file
dfs <- lapply(X=files, FUN=read.csv, as.is=TRUE)

# single data.frame if all files have n columns:
df <- do.call(rbind, dfs)


# PS: much faster in this example could be
library("data.table")
dfs <- lapply(X=files, FUN=fread, sep=",")
df <- rbindlist(dfs)
```

for-loops, lapply, Rcpp – lapply is easy to expand

```
library("pbapply")  # progress bar with remaining time
library("parallel") # for multicore parallel execution

nc <- detectCores()-1
dfs <-        lapply(X=files, FUN=read.csv, as.is=TRUE)
dfs <-      pblapply(X=files, FUN=read.csv, as.is=TRUE)
dfs <-      mclapply(X=files, FUN=read.csv, as.is=TRUE,
                     mc.cores=nc) # easy on linux
# more code needed on windows:
cl <- makePSOCKcluster(nc)
dfs <- parLapply(cl, X=files, fun=read.csv, as.is=TRUE)
stopCluster(cl)
# sometimes needed before parLapply call:
clusterExport(cl, c("files","otherObjects"))
clusterEvalQ(cl, library("somePackage"))

# time your code:
begintime <- Sys.time(); begintime
parLapply(cl, X, fun)
Sys.time() - begintime ; rm(cl, begintime)
```

for-loops, lapply, Rcpp – better code timing

```
#install.packages("microbenchmark")
library(microbenchmark)
forbad <- function(n)  # fibonacci
  {
  fibvals <- c(1,1)
  for (i in 3:n) fibvals[i] <- fibvals[i-1]+fibvals[i-2]
  fibvals
  }
forgood <- function(n)
  {
  fibvals <- rep(1,n)
  for (i in 3:n) fibvals[i] <- fibvals[i-1]+fibvals[i-2]
  fibvals
  }

mb <- microbenchmark(forbad(2000), forgood(2000))
mb
Unit: milliseconds
          expr      min       lq     mean   median       uq       max neval
  forbad(2000) 5.186815 5.500281 6.251113 5.662076 6.319942 40.908350   100
 forgood(2000) 2.302748 2.432611 2.533551 2.489203 2.587277  3.323405   100
```

for-loops, lapply, Rcpp – loop unavoidable, but slow?

Rcpp saves the day!

Loops are fast in C++, so outsource that part of your code into C.
The package Rcpp will compile it for you and make it available as a normally accessible R function.

Start learning at
http://adv-r.had.co.nz/Rcpp.html

Speed gain is highly variable, of course, but you might be able to reduce your computing time from 2 hours to 20 seconds!

To find the slow parts of your code, you perform a process called "profiling":
www.r-bloggers.com/profiling-r-code
https://stat.ethz.ch/R-manual/R-devel/library/utils/html/Rprof.html
http://adv-r.had.co.nz/Profiling.html
New and very promising:
https://blog.rstudio.org/2016/05/23/profiling-with-rstudio-and-profvis
https://support.rstudio.com/hc/en-us/articles/218221837-Profiling-with-RStudio