Statistical analysis

[R]

Dani Arribas-Bel & Thomas De Graaff

September 5, 2014

Outline

Today

- Reproducible statistical analysis
- Reinhart & Rogoff: a textbook example of the power of replication
- `R`: what is it and why should I care?
- `R` overview
  - Libraries and help
  - Reading data
  - Exploring the `data.frame`
  - Manipulate a `data.frame`
  - Analyze data
  - Visualize data
  - Export results

Introduction

Reproducible statistical analysis

Open principles applied to the way you conduct statistical data analysis:

- Make the process explicit and transparent
- Provide every input required to reproduce the analysis carried out and obtain the same results, as reported in the final document published

This typically involves three levels:

- **Data** used for the study
- **Code** created to perform the analysis
- **Platform** required to run the code

Being fully open on the three is not always possible (e.g. proprietary data/software), but that should be goal to which to get as close as possible.

Getting halfway is better than not starting

In this session we will focus on the last two: **code** and **platform**

Reinhart & Rogoff

- In 2010, C. Reinhart and K. Rogoff put together a paper claiming to show how economic growth is seriously dampened once the ratio of debt to GDP goes above 90%
- The paper was very influential and became one of the most commonly cited ones to argue for austerity measures
- In 2013, **Thomas Herndon**, a PhD student at UMass, tried to replicate the results for a class assignment
- He could not, so finally he obtained from Reinhart the original (Excel) code and data only to find **results diverged** because of:
  - Selective exclusion of available data
  - Unconventional weighting of summary statistics
  - Coding errors
- The **replication** is posted online, together with the data and R code used for the paper

Reinhart & Rogoff

Lessons:

- **No one is free from mistakes** (even Harvard top economists!)
- **Posting your data and code** but, if you don't, sharing them honestly upon request is a good second best
- **Replication** should be much more widespread
- . . . you should not underestimate PhD students without a big name but with lots of time!

R

R: what is it?

R is a language and environment for statistical computing and graphics

- **language & environment**
- **statistical computing**
- **graphics**

Characteristics:

- It is a Free implementation of the S language created by **R**oss Ihaka and **R**obert Gentleman in 1993
- **Cross-platform**: runs on many *nix (included Linux) systems, Windows and MacOS.
- It is licensed under GPL, which makes it **free**...
  - ... as in **beer**
  - ... as in **speech**

Why should I care about R?

- Philosophy behind the project
- Convenience (once you get ahead the learning curve)

Some people who care about R:

- Many top universities use R in teaching and research
- Google and Facebook
- New York Times

The R Philosophy

. . . Then sit back, relax, and enjoy being part of something big. . .
[Tom Preston-Werner]

Being Free Software ("the users have the freedom to run, copy, distribute, study, change and improve the software") has enhanced:

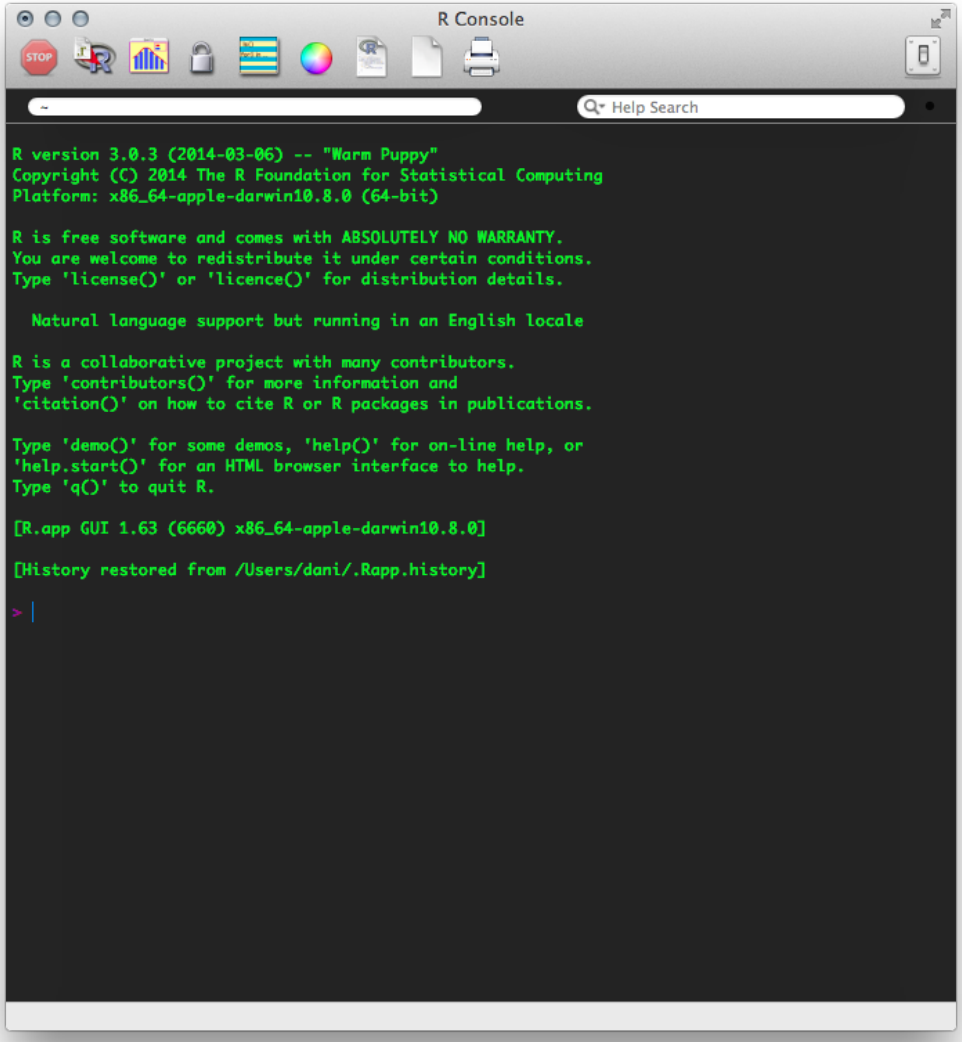
- **Worldwide community** of dedicated and enthusiastic users, contributors and developers that:
  - Lowers the entry barriers (mailing lists, blog posts, online tutorials, workshops. . .)
  - Continuously expands the capability and functionality
- Becoming an instrument for **democratization** of academic software and technology transfer
- Becoming the **lingua franca** in academia
- Facilitating reproductibility and Open Science

R as free beer

- The price is right
  - Education
  - Installation across multiple machines
- The *beer selection* is wide (CRAN hosts 3,669 available packages as of March 10th. 2012)
  - Makes R a good one stop-shop and a good investment of your time to learn it
  - No market profitability constraints put it at the cutting edge (research sandbox)
- Linus' Law: *"given enough eyeballs, all bugs are shallow"*
  - More reliable and stable

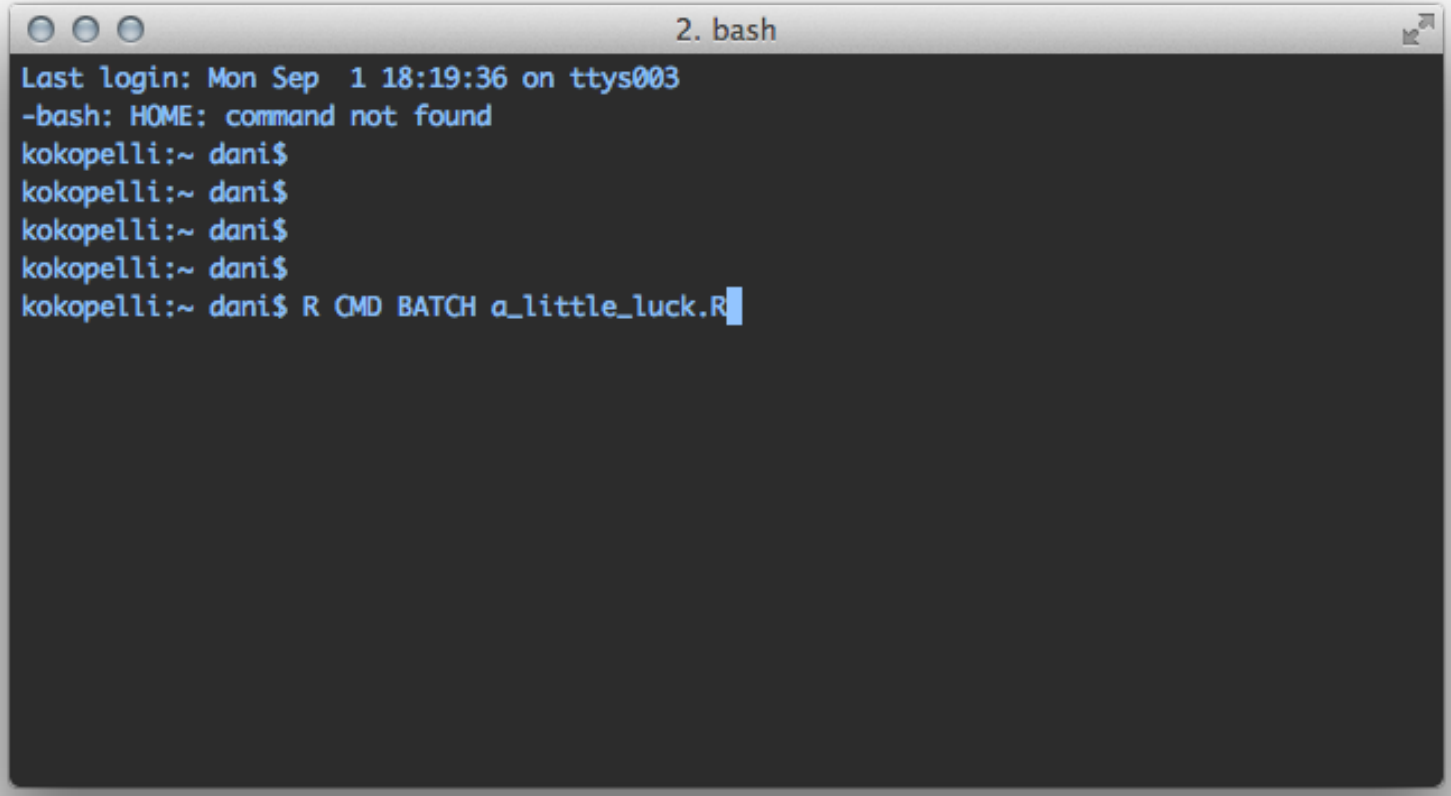
Ways to interact with R

- Interactive shell



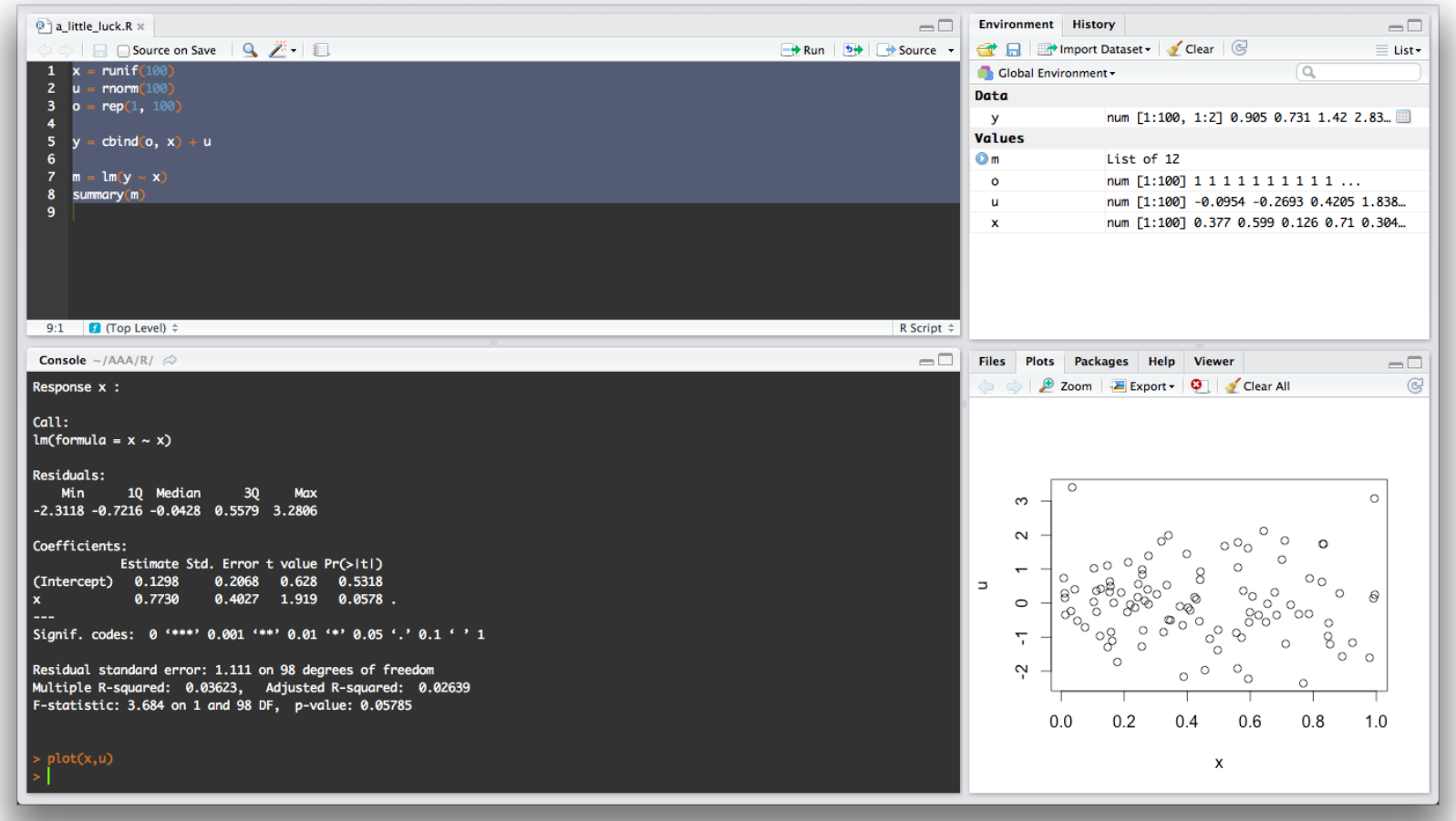
Ways to interact with R

- Batch mode from the command line



Ways to interact with R

- IDEs (e.g. RStudio)



R overview

Packages

Look for R info and packages

- Project website: `http://r-project.org`
- The Comprehensive R Archive Network (CRAN)
- The R-Journal (and JoSS)
- R bloggers
- Twitter: the `#rstats` hashtag
- Google (good luck on that)

Install and load packages

- Windows and MacOS GUIs have installers
- Command line with `instal.packages` function
- Command `library` (e.d. `library(maptools)` to load the package `maptools`)

Help

NEVER HAVE I FELT SO CLOSE TO ANOTHER SOUL

AND YET SO HELPLESSLY ALONE

AS WHEN I GOOGLE AN ERROR

AND THERE'S ONE RESULT

A THREAD BY SOMEONE WITH THE SAME PROBLEM

AND NO ANSWER

LAST POSTED TO IN 2003

Help and documentation

- R built-in search capability

Command	Function
`?read.csv`	Check local documentation for `read.csv` function
`spdep::moran.test`	Check local documentation in package `spdep` for `moran.test`
`help("read.csv")`	Check local documentation for `read.csv` function
`help.search("read.csv")`	Search for "read.csv" in all help files
`RSiteSearch("plot maps")`	Search for the term "plot maps" in the RSiteSearch website (requires connectivity)

- StackOverflow

Reading data

Point to the folder

```
#setwd('~/code/WooWii/slides/')
getwd()
```

```
## [1] "/Users/tomba/Dropbox/Thomas/Colleges/Workflow/Woow
```

Native csv reading

```
nl <- read.csv("../Paper/Final/Data/RR - Netherlands.csv")
```

Foreign formats supported

```
library(foreign)
proc <- read.dta("../Paper/Final/Data/RR-processed.dta")
```

Many other formats supported (dbf, xls, sql-like databases...)

Exploring a data.frame

head/tail for the top/bottom of the table

```
head(nl)
```

```
##         Country Year Debt  GDP1 GDP2 RGDP1 RGDP2 GDPI1 GDP
## 1 Netherlands 1807   NA 490.3   NA 381.9    NA 128.4
## 2 Netherlands 1808   NA 436.2   NA 339.3    NA 128.6
## 3 Netherlands 1809   NA 407.9   NA 334.8    NA 121.8
## 4 Netherlands 1810   NA    NA   NA    NA    NA    NA
## 5 Netherlands 1811   NA    NA   NA    NA    NA    NA
## 6 Netherlands 1812   NA    NA   NA    NA    NA    NA
```

```
nl[1, ]
```

```
##         Country Year Debt  GDP1 GDP2 RGDP1 RGDP2 GDPI1 GDP
## 1 Netherlands 1807   NA 490.3   NA 381.9    NA 128.4
```

Exploring a data.frame

```
max(nl$GDP1, na.rm=TRUE)
```

```
## [1] 6489
```

```
min(nl$Debt, na.rm=TRUE)
```

```
## [1] 6.6
```

Create new variables

```
nl['dtg'] = nl$Debt / nl$GDP1
```

Exploring a data.frame

summary for basic statistics

```
summary(nl)
```

```
##          Country          Year          Debt                (
##  Netherlands:204   Min.   :1807   Min.   :    7   Min.
##                    1st Qu.:1858   1st Qu.:   26   1st Qu
##                    Median :1908   Median :  620   Median
##                    Mean   :1908   Mean   : 1263   Mean
##                    3rd Qu.:1959   3rd Qu.: 1158   3rd Qu
##                    Max.   :2010   Max.   :12619   Max.
##                                   NA's   :74      NA's
##       GDP2             RGDP1            RGDP2          GDP
##  Min.   : 14.5   Min.   :   335   Min.   :243   Min.
##  1st Qu.: 53.4   1st Qu.:   772   1st Qu.:279   1st Qu.
##  Median :178.8   Median :  2078   Median :343   Median
##  Mean   :212.3   Mean   : 61539   Mean   :356   Mean
##  3rd Qu.:316.1   3rd Qu.: 83826   3rd Qu.:427   3rd Qu.
```

Querying a data.frame

A data.frame has fancy query features

```
with_debt <- nl[!is.na(nl$Debt), ]
head(with_debt, 3)
```

```
##         Country Year Debt GDP1 GDP2 RGDP1 RGDP2  GDPI1 GI
## 74 Netherlands 1880  942 1120   NA  1139    NA  98.36
## 75 Netherlands 1881  941 1134   NA  1160    NA  97.80
## 76 Netherlands 1882  999 1191   NA  1191    NA 100.00
```

```
nl_clean <- nl[!is.na(nl$GDP1), ]
mean_gdp <- mean(nl_clean$GDP1)
high_gdp <- nl_clean[nl_clean$GDP1 > mean_gdp, ]
head(high_gdp, 3)
```

```
##          Country Year Debt GDP1 GDP2 RGDP1 RGDP2 GDPI1 GI
## 99  Netherlands 1905 1106 1711   NA  1931    NA 88.62
## 100 Netherlands 1906 1145 1823   NA  1971    NA 92.50
```

Querying a data.frame

Which you can combine:

```
super_clean <- nl[(!is.na(nl$GDP1)) & (!is.na(nl$Debt)), ]
ratio <- super_clean$Debt / super_clean$GDP1
good_years <- super_clean[(ratio < 0.9) & (super_clean$GDP1
head(good_years, 3)
```

```
##          Country Year Debt GDP1 GDP2 RGDP1 RGDP2 GDPI1 GI
## 99  Netherlands 1905 1106 1711   NA  1931    NA 88.62
## 100 Netherlands 1906 1145 1823   NA  1971    NA 92.50
## 101 Netherlands 1907 1140 1812   NA  1935    NA 93.61
```

Hands on!

In `proc`:

- In what country and year the GDP is largest?
- Show a country in which the Debt/GDP ratio has never been beyond 90%

Answer

- In what country and year the GDP is largest?

```
clean_gdp <- proc[!is.na(proc$GDP), ]
max_gdp <- max(clean_gdp$GDP)
top_gdp <- clean_gdp[clean_gdp$GDP == max_gdp, ]
top_gdp[, c('Country', 'Year', 'GDP')]
```

```
##      Country Year     GDP
## 1107      UK 2008 1446110
```

- Show a country in which the Debt/GDP ratio has never been beyond 90%

```
proc['dtg'] <- 100 * proc$Debt / proc$GDP
debt_to_gdp_clean <- proc[!is.na(proc$dtg), ]
good_boys <- debt_to_gdp_clean[debt_to_gdp_clean$dtg < 0.9
good_boys[1:5, c('Country', 'Year', 'dtg')]
```

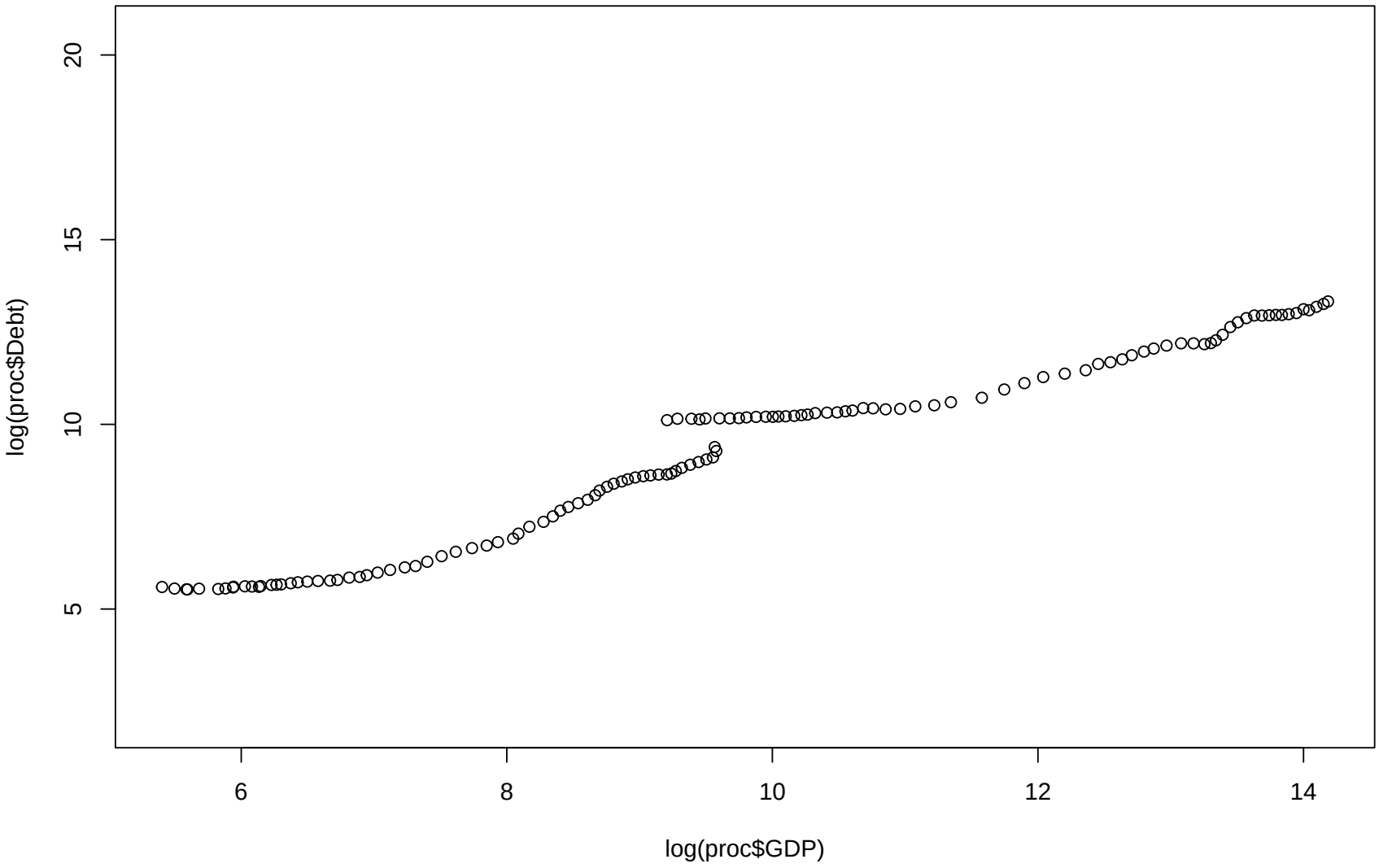
Analyze data: regression

```
ols <- lm('log(Debt) ~ log(GDP)', data=proc)
summary(ols)
```

```
##
## Call:
## lm(formula = "log(Debt) ~ log(GDP)", data = proc)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -0.698 -0.259 -0.121  0.120  1.421
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.2086     0.1640   -1.27     0.21
## log(GDP)      0.9668     0.0164   59.08   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
##
```

Visualization

```
plot(log(proc$GDP), log(proc$Debt))
```



Try also `plot(ols)` in RStudio!

Advanced manipulations

Corrected Reinhart & Rogoff Table 1 (as shown by Herndon et al.)

```
library(xtable)
proc$dgcat.lm <- cut(proc$debtgdp, breaks=c(0,30,60,90,Inf)
proc$dgcat <- factor(proc$dgcat.lm, labels = c("0-30%","30-
(RR.correct.mean <- with(proc, tapply( dRGDP, dgcat, mean,
```

```
##      0-30%     30-60%     60-90% Above 90%
##      4.174      3.116      3.222      2.168
```

Converted to a data.frame

```
table_df <- data.frame(RR.correct.mean, dgcat=names(RR.cor
table_df
```

```
##              RR.correct.mean      dgcat
## 0-30%                  4.174      0-30%
## 30-60%                 3.116     30-60%
## 60-90%                 3.222     60-90%
```

Export results

Simply write to a csv

```
write.csv(table_df, '~/Desktop/table.csv')
```

Or to LaTeX

```
library(xtable)
xtable(table_df)
```

```
## % latex table generated in R 3.1.1 by xtable 1.7-3 packa
## % Fri Sep  5 18:10:51 2014
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrl}
##   \hline
##  & RR.correct.mean & dgcat \\
##   \hline
## 0-30\% & 4.17 & 0-30\% \\
##   30-60\% & 3.12 & 30-60\% \\
```

Export results

If we wanted to write it out to a file

```
sink('~/Desktop/table.tex')
xtable(table_df)
sink()
```

For this session, we have borrowed important amounts of inspiration and material from **Software Carpentry**'s session on git and the freely available book Pro Git