Welcome to: “Basic scientific computing for drug discovery”

6/11/2018

Universiteit Leiden
The Netherlands

The teaching team

Python/ Modeling

Dr M. M. Palm

(Margriet)

Python/ Cheminformatics

Brandon Bongers, Msc

Python/ Chemoinformatics

Dr G.J. P. van Westen

(Gerard)

R/Bioinformatics

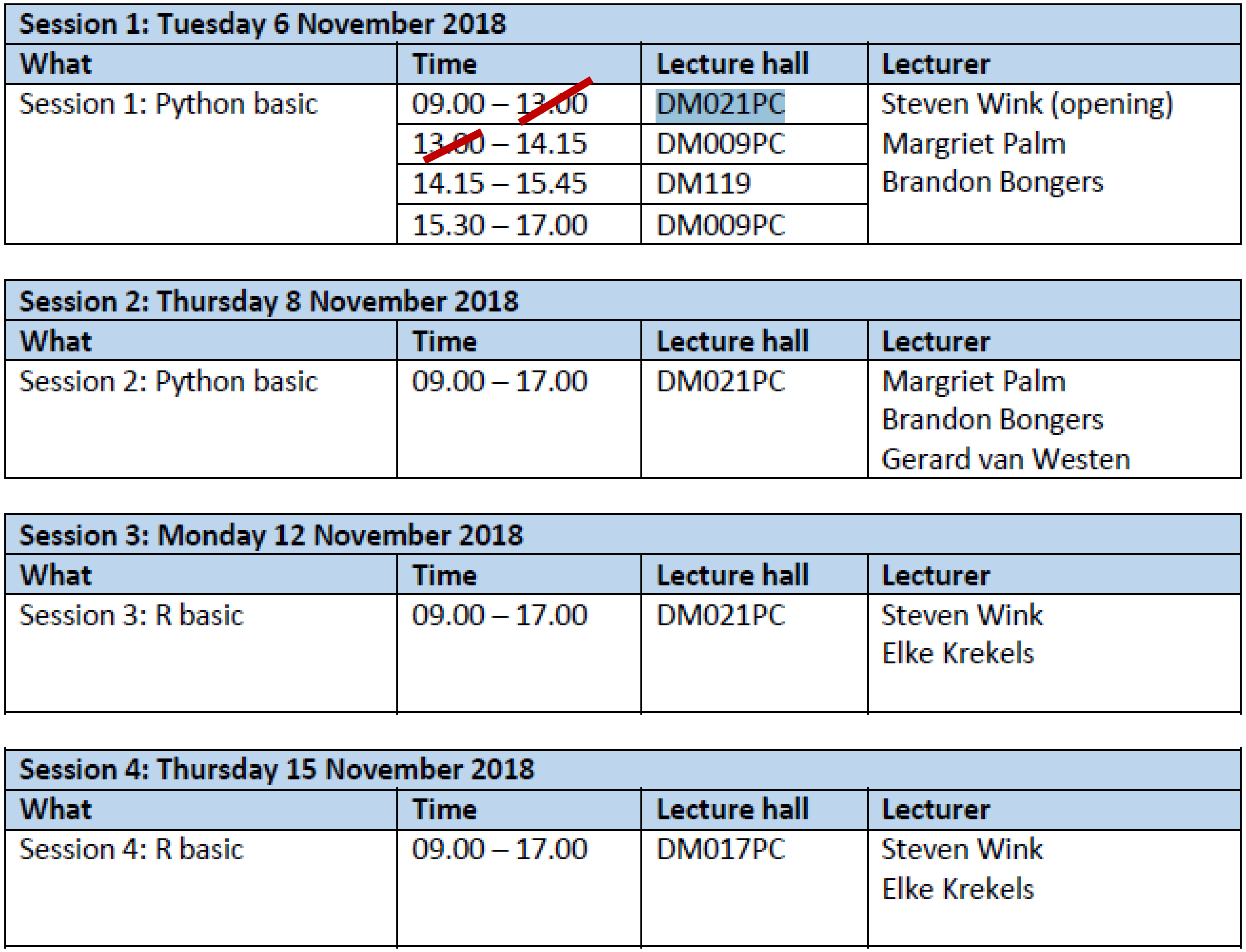
Dr S. Wink (Steven)

Pharmacokinetics

Dr E.H.J. Krekels (Elke)

Organizer

Ivonne Koomen



Session 1: Tuesday 6 November 2018			
What	**Time**	**Lecture hall**	**Lecturer**
Session 1: Python basic	09.00 – ~~13.00~~	DM021PC	Steven Wink (opening)
	~~13.00~~ – 14.15	DM009PC	Margriet Palm
	14.15 – 15.45	DM119	Brandon Bongers
	15.30 – 17.00	DM009PC	

Session 2: Thursday 8 November 2018			
What	**Time**	**Lecture hall**	**Lecturer**
Session 2: Python basic	09.00 – 17.00	DM021PC	Margriet Palm Brandon Bongers Gerard van Westen

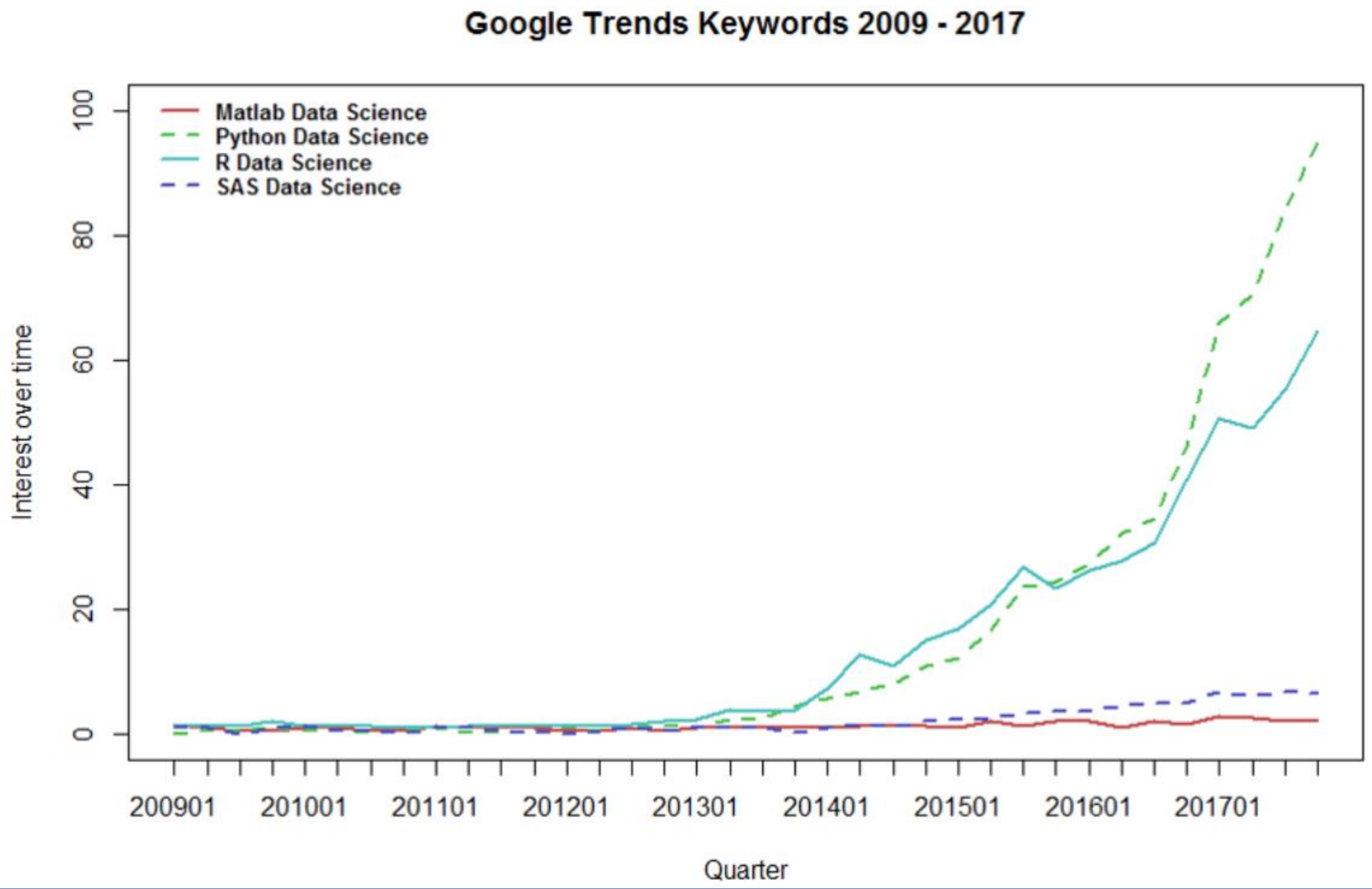
Session 3: Monday 12 November 2018			
What	**Time**	**Lecture hall**	**Lecturer**
Session 3: R basic	09.00 – 17.00	DM021PC	Steven Wink Elke Krekels

Session 4: Thursday 15 November 2018			
What	**Time**	**Lecture hall**	**Lecturer**
Session 4: R basic	09.00 – 17.00	DM017PC	Steven Wink Elke Krekels

R & Python popular data science languages.

For a nice article on what specific differences are:

https://www.digitalvidya.com/blog/r-vs-python/



R or Python for data science

Python

- Easier syntax
  - Bit easier learning curve
- Generic programming language
  - Web api's, database queries, pipeline development

R

- Developed by/for statisticians
  - Statistics books will often use R for implementation
- Tools for omics/biology (Bioconductor)
  - Microarray, annotation tools, RNA-seq analysis development is done in R

How to choose: look at your work environment.

A lot of biologists and statisticians tend to use R.

Computational scientists/ software developers tend to use Python over R.

Computational applications

- In your feedback: what topics would you be interested in?

  Some examples are introductions into:

  - ODE modeling (R)
  - Machine learning (python or R)
  - Pharmacokinetic modeling (R)
  - Omics analysis? (RNA-seq, for metabolomics we would have to check) (R)
  - Statistics using linear modeling approaches or generalized additive models (R)

Good luck and have fun!

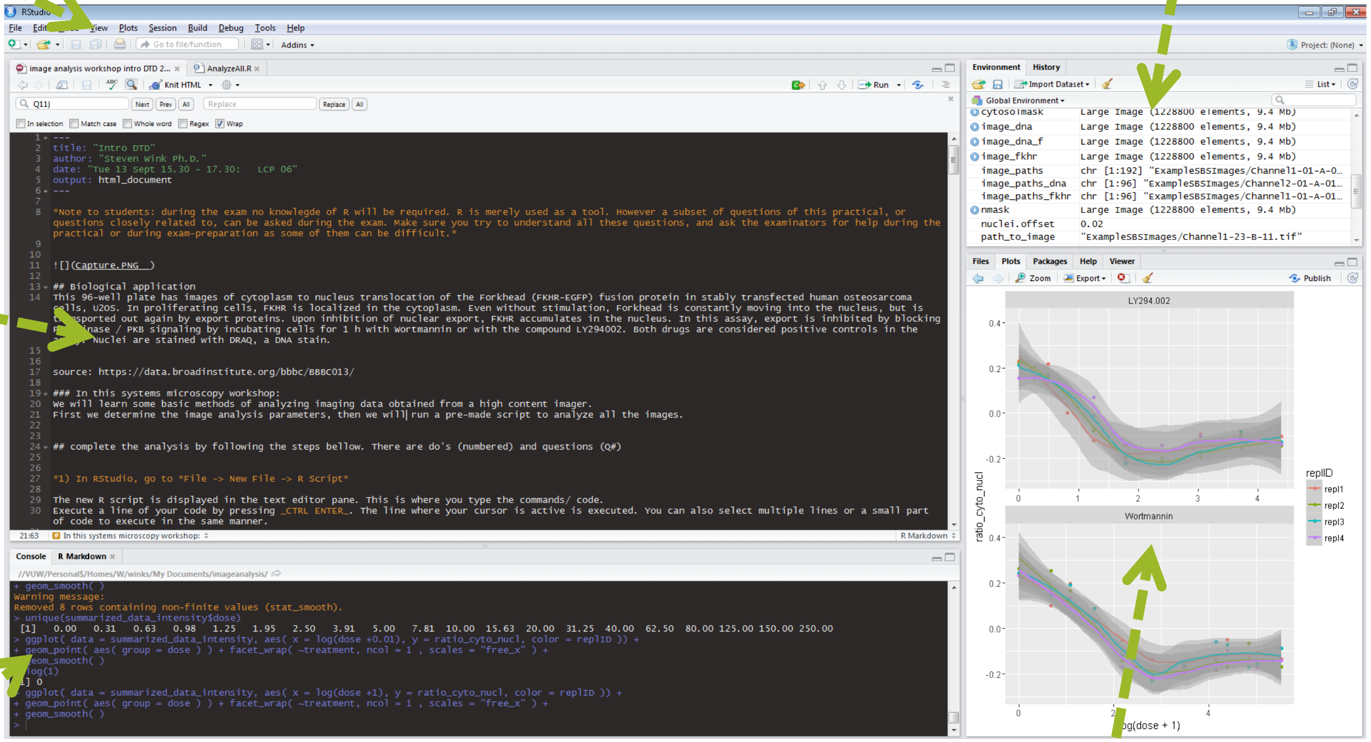
Rstudio (IDE)

Run lines with 'Ctrl + enter'

Menu bar

Environment

Text editor



```
---
title: "Intro DTD"
author: "Steven Wink Ph.D."
date: "Tue 13 Sept 15.30 - 17.30:   LCP 06"
output: html_document
---

*Note to students: during the exam no knowlegde of R will be required. R is merely used as a tool. However a subset of questions of this practical, or questions closely related to, can be asked during the exam. Make sure you try to understand all these questions, and ask the examinators for help during the practical or during exam-preparation as some of them can be difficult.*


![](Capture.PNG  )

## Biological application
This 96-well plate has images of cytoplasm to nucleus translocation of the Forkhead (FKHR-EGFP) fusion protein in stably transfected human osteosarcoma ells, U2OS. In proliferating cells, FKHR is localized in the cytoplasm. Even without stimulation, Forkhead is constantly moving into the nucleus, but is t  sported out again by export proteins. Upon inhibition of nuclear export, FKHR accumulates in the nucleus. In this assay, export is inhibited by blocking P  inase / PKB signaling by incubating cells for 1 h with Wortmannin or with the compound LY294002. Both drugs are considered positive controls in the a    . Nuclei are stained with DRAQ, a DNA stain.


source: https://data.broadinstitute.org/bbbc/BBBC013/

### In this systems microscopy workshop:
We will learn some basic methods of analyzing imaging data obtained from a high content imager.
First we determine the image analysis parameters, then we will run a pre-made script to analyze all the images.


## complete the analysis by following the steps bellow. There are do's (numbered) and questions (Q#)


*1) In RStudio, go to *File -> New File -> R Script*

The new R script is displayed in the text editor pane. This is where you type the commands/ code.
Execute a line of your code by pressing _CTRL ENTER_. The line where your cursor is active is executed. You can also select multiple lines or a small part of code to execute in the same manner.
```

```
//VUW/Personal$/Homes/W/winks/My Documents/imageanalysis/
+ geom_smooth( )
Warning message:
Removed 8 rows containing non-finite values (stat_smooth).
> unique(summarized_data_intensity$dose)
 [1]   0.00   0.31   0.63   0.98   1.25   1.95   2.50   3.91   5.00   7.81  10.00  15.63  20.00  31.25  40.00  62.50  80.00 125.00 150.00 250.00
> ggplot( data = summarized_data_intensity, aes( x = log(dose +0.01), y = ratio_cyto_nucl, color = replID )) +
+ geom_point( aes( group = dose ) ) + facet_wrap( ~treatment, ncol = 1 , scales = "free_x" ) +
 eom_smooth( )
 log(1)
 ] 0
 ggplot( data = summarized_data_intensity, aes( x = log(dose +1), y = ratio_cyto_nucl, color = replID )) +
+ geom_point( aes( group = dose ) ) + facet_wrap( ~treatment, ncol = 1 , scales = "free_x" ) +
+ geom_smooth( )
> 
```

```
cytosolmask        Large Image (1228800 elements, 9.4 Mb)
image_dna          Large Image (1228800 elements, 9.4 Mb)
image_dna_f        Large Image (1228800 elements, 9.4 Mb)
image_fkhr         Large Image (1228800 elements, 9.4 Mb)
image_paths        chr [1:192] "ExampleSBSImages/Channel1-01-A-0…
image_paths_dna    chr [1:96] "ExampleSBSImages/Channel2-01-A-01…
image_paths_fkhr   chr [1:96] "ExampleSBSImages/Channel1-01-A-01…
nmask              Large Image (1228800 elements, 9.4 Mb)
nuclei.offset      0.02
path_to_image      "ExampleSBSImages/Channel1-23-B-11.tif"
```

R console

Graphics & help

Documentation

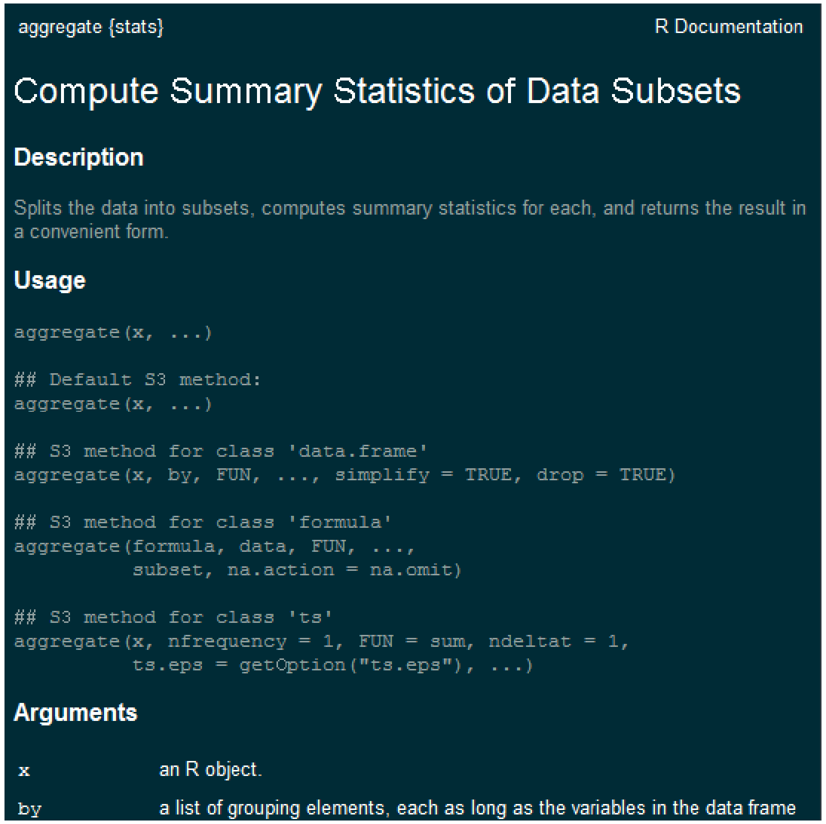
- ?plot
  - R documentation, for arguments and examples

- help("plot")
  - Same as ?
- ?`+`
  - For special symbols use backticks

- ??Mean
  - Fuzzy matching, for if you don't remember the exact function name

- www.stackoverflow.com
  - For finding answers to your coding problems, usually you are not the first!
  - After an extensive search for existing answers, you are encouraged to post your own question.

Documentation



aggregate {stats} R Documentation

Compute Summary Statistics of Data Subsets

Description

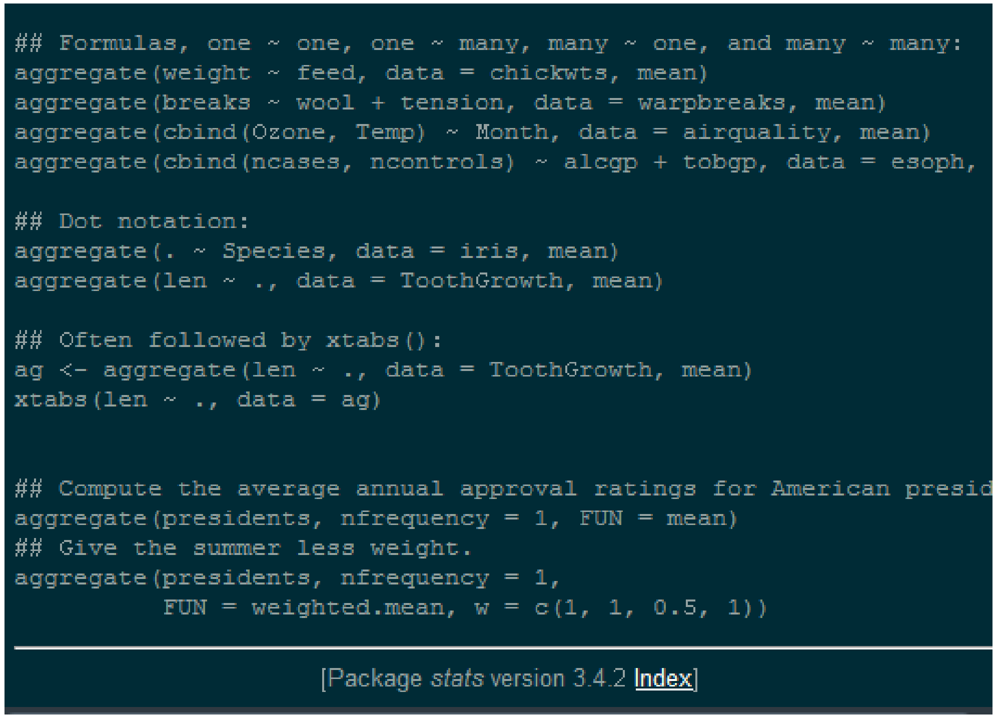
Splits the data into subsets, computes summary statistics for each, and returns the result in a convenient form.

Usage

```
aggregate(x, ...)

## Default S3 method:
aggregate(x, ...)

## S3 method for class 'data.frame'
aggregate(x, by, FUN, ..., simplify = TRUE, drop = TRUE)

## S3 method for class 'formula'
aggregate(formula, data, FUN, ...,
          subset, na.action = na.omit)

## S3 method for class 'ts'
aggregate(x, nfrequency = 1, FUN = sum, ndeltat = 1,
          ts.eps = getOption("ts.eps"), ...)
```

Arguments

`x`	an R object.
`by`	a list of grouping elements, each as long as the variables in the data frame

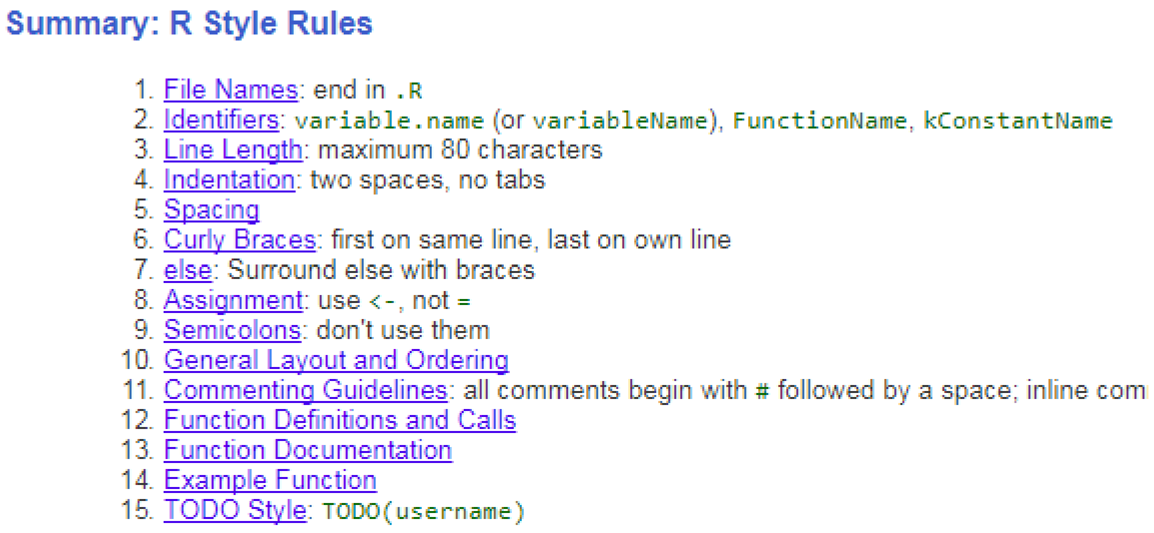


```
## Formulas, one ~ one, one ~ many, many ~ one, and many ~ many:
aggregate(weight ~ feed, data = chickwts, mean)
aggregate(breaks ~ wool + tension, data = warpbreaks, mean)
aggregate(cbind(Ozone, Temp) ~ Month, data = airquality, mean)
aggregate(cbind(ncases, ncontrols) ~ alcgp + tobgp, data = esoph,

## Dot notation:
aggregate(. ~ Species, data = iris, mean)
aggregate(len ~ ., data = ToothGrowth, mean)

## Often followed by xtabs():
ag <- aggregate(len ~ ., data = ToothGrowth, mean)
xtabs(len ~ ., data = ag)


## Compute the average annual approval ratings for American presid
aggregate(presidents, nfrequency = 1, FUN = mean)
## Give the summer less weight.
aggregate(presidents, nfrequency = 1,
          FUN = weighted.mean, w = c(1, 1, 0.5, 1))
```

[Package *stats* version 3.4.2 Index]

GPP

- https://google.github.io/styleguide/Rguide.xml



Summary: R Style Rules

1. File Names: end in `.R`
2. Identifiers: `variable.name` (or `variableName`), `FunctionName`, `kConstantName`
3. Line Length: maximum 80 characters
4. Indentation: two spaces, no tabs
5. Spacing
6. Curly Braces: first on same line, last on own line
7. else: Surround else with braces
8. Assignment: use `<-`, not `=`
9. Semicolons: don't use them
10. General Layout and Ordering
11. Commenting Guidelines: all comments begin with `#` followed by a space; inline com
12. Function Definitions and Calls
13. Function Documentation
14. Example Function
15. TODO Style: `TODO(username)`

Alternative:

Variable names separated with _

Function names start with small letter, then camel-case

Constants without the k in front.

GPP

Write multiple R scripts each performing a certain definable or related task.

Not to long and not to short….

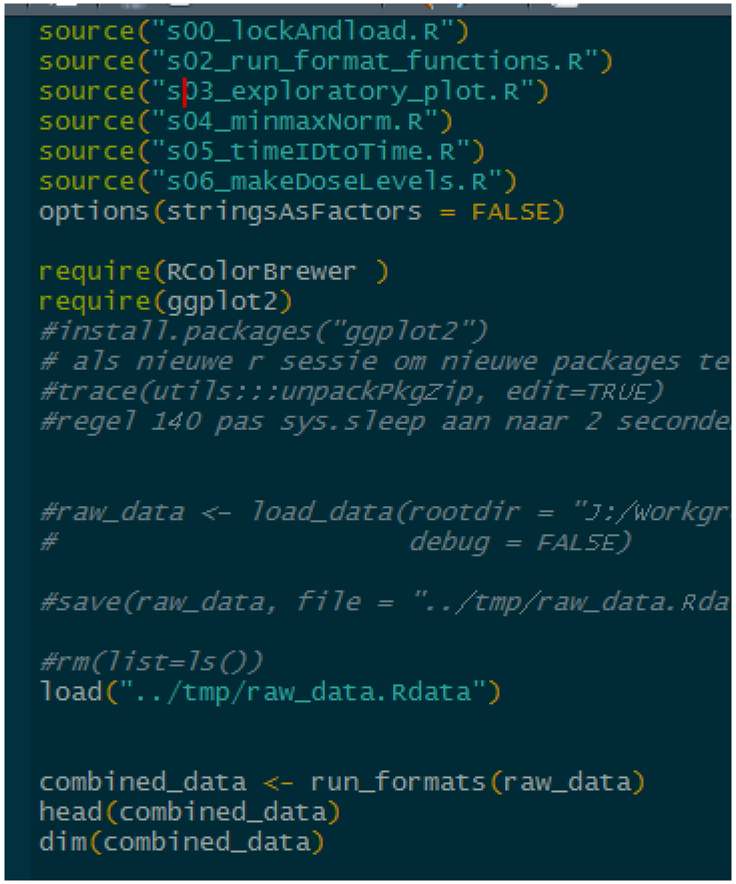
- Easier to understand
- Easier to debug
- Easier to extend / modify
- Easier to reuse

- Avoid using global variables as much as possible, use local variables inside functions instead.
- Name stuff in a way that makes sense to everyone.

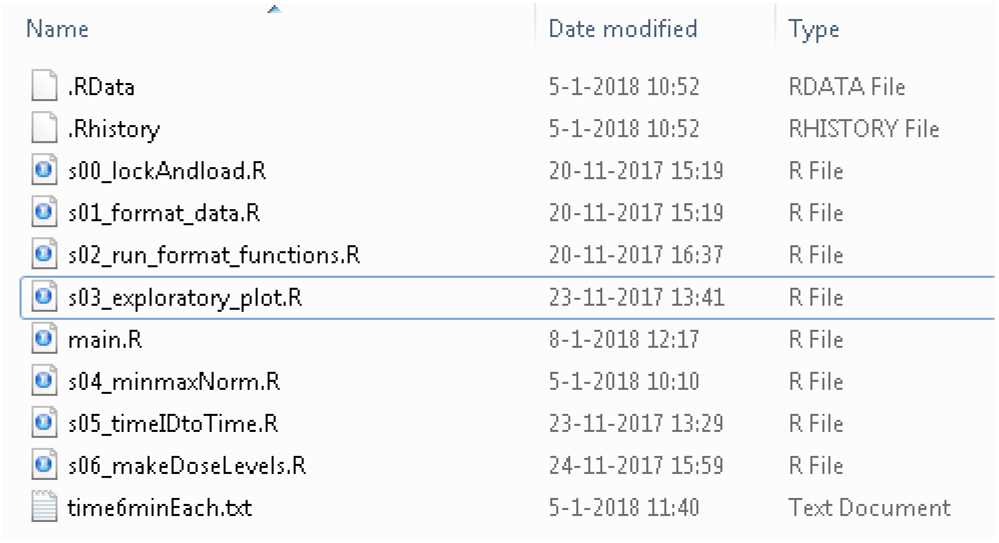
- Save your workspace
- Save .Rdata R objects

GPP

- **Example**



```
source("s00_lockAndload.R")
source("s02_run_format_functions.R")
source("s03_exploratory_plot.R")
source("s04_minmaxNorm.R")
source("s05_timeIDtoTime.R")
source("s06_makeDoseLevels.R")
options(stringsAsFactors = FALSE)

require(RColorBrewer )
require(ggplot2)
#install.packages("ggplot2")
# als nieuwe r sessie om nieuwe packages te
#trace(utils:::unpackPkgZip, edit=TRUE)
#regel 140 pas sys.sleep aan naar 2 seconde


#raw_data <- load_data(rootdir = "J:/workgr
#                       debug = FALSE)

#save(raw_data, file = "../tmp/raw_data.Rda

#rm(list=ls())
load("../tmp/raw_data.Rdata")


combined_data <- run_formats(raw_data)
head(combined_data)
dim(combined_data)
```



Name	Date modified	Type
.RData	5-1-2018 10:52	RDATA File
.Rhistory	5-1-2018 10:52	RHISTORY File
s00_lockAndload.R	20-11-2017 15:19	R File
s01_format_data.R	20-11-2017 15:19	R File
s02_run_format_functions.R	20-11-2017 16:37	R File
s03_exploratory_plot.R	23-11-2017 13:41	R File
main.R	8-1-2018 12:17	R File
s04_minmaxNorm.R	5-1-2018 10:10	R File
s05_timeIDtoTime.R	23-11-2017 13:29	R File
s06_makeDoseLevels.R	24-11-2017 15:59	R File
time6minEach.txt	5-1-2018 11:40	Text Document