

# 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/  
Cheminformatics



Dr G.J. P. van Westen  
(Gerard)

R/Bioinformatics



Dr S. Wink (Steven)

Pharmacokinetics



Dr E.H.J. Krekels (Elke)

Organizer



Ivonne Koomen

ICT developer



Michael van Vliet

| Session 1: Tuesday 6 November 2018 |                          |              |                       |
|------------------------------------|--------------------------|--------------|-----------------------|
| What                               | Time                     | Lecture hall | Lecturer              |
| Session 1: Python basic            | 09.00 – <del>13.00</del> | DM021PC      | Steven Wink (opening) |
|                                    | <del>13.00</del> – 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<br>Brandon Bongers<br>Gerard van Westen |

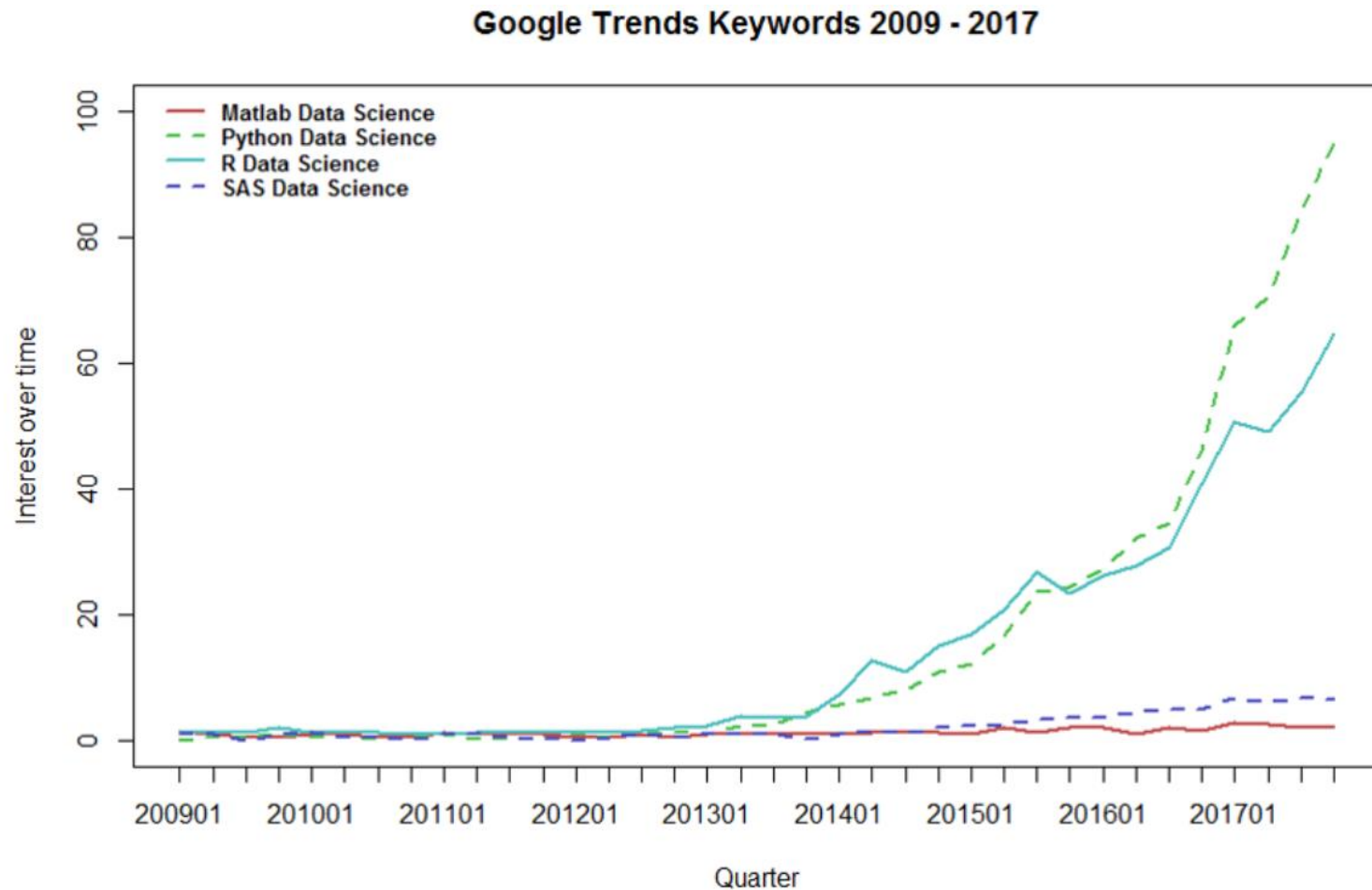
| Session 3: Monday 12 November 2018 |               |              |                             |
|------------------------------------|---------------|--------------|-----------------------------|
| What                               | Time          | Lecture hall | Lecturer                    |
| Session 3: R basic                 | 09.00 – 17.00 | DM021PC      | Steven Wink<br>Elke Krekels |

| Session 4: Thursday 15 November 2018 |               |              |                             |
|--------------------------------------|---------------|--------------|-----------------------------|
| What                                 | Time          | Lecture hall | Lecturer                    |
| Session 4: R basic                   | 09.00 – 17.00 | DM017PC      | Steven Wink<br>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'

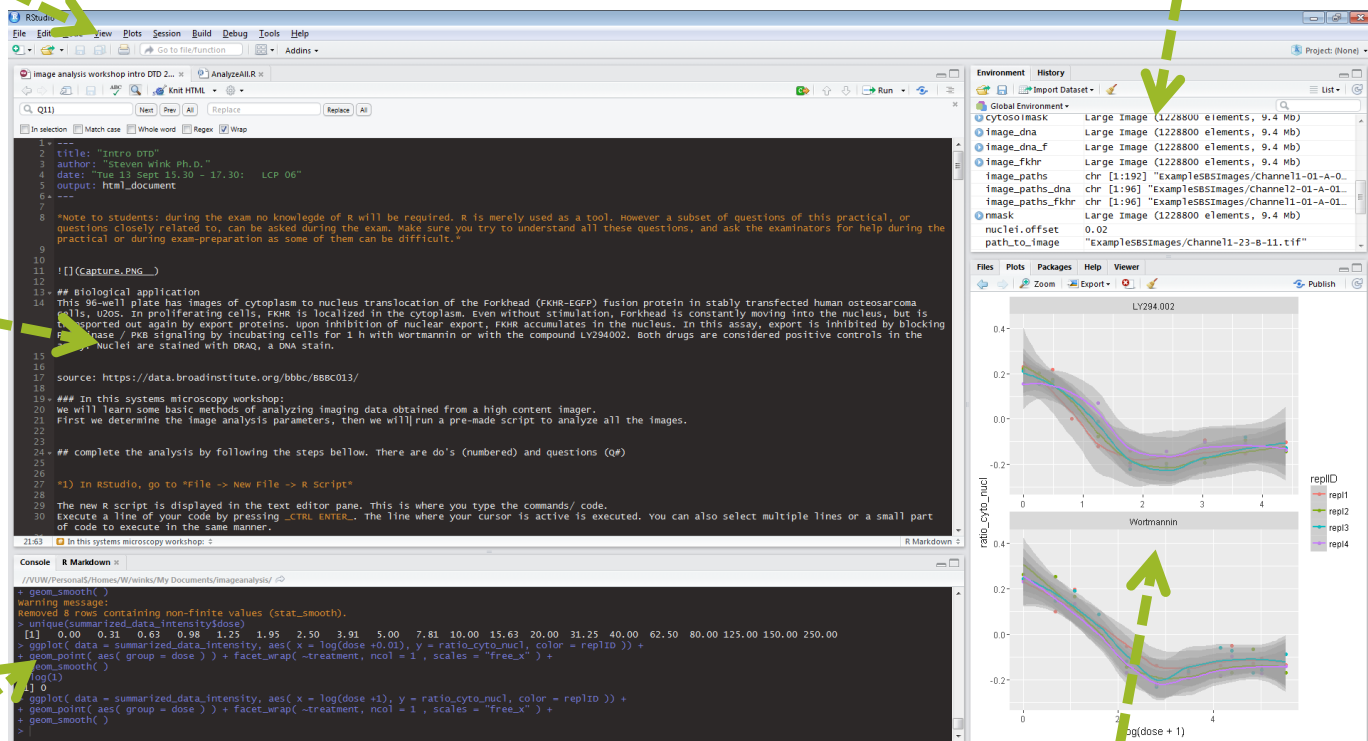
Menu bar

Environment

Text editor

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](http://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 too long and not too 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 |