Quantitative social science with R

Introduction

Edu Gonzalo Almorox

18/10/2017

Quantitative social science with R

Introduction

Edu Gonzalo Almorox

18/10/2017

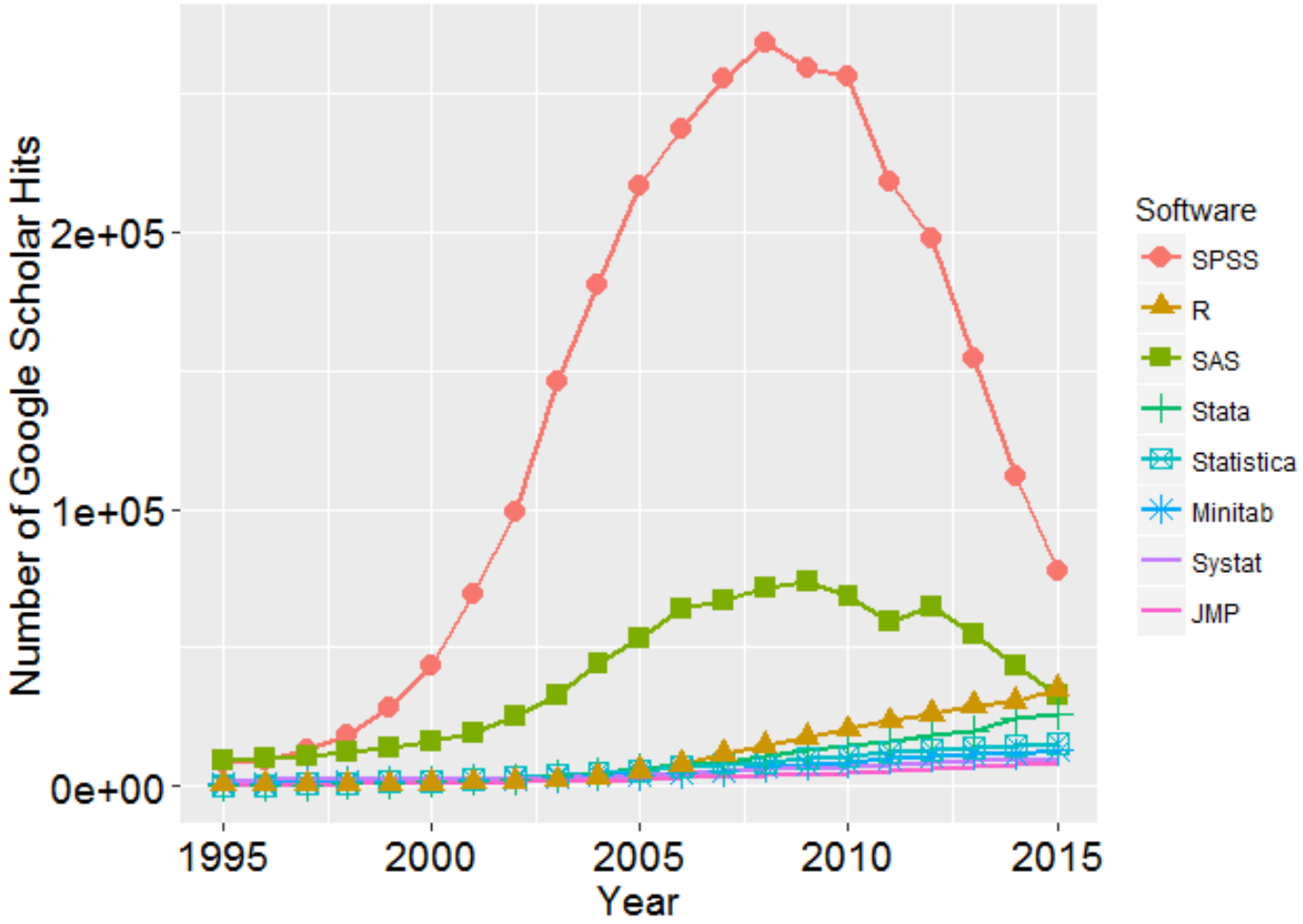
Outline

1. What´s R?
2. R-Studio
3. Start with R
4. Data structures

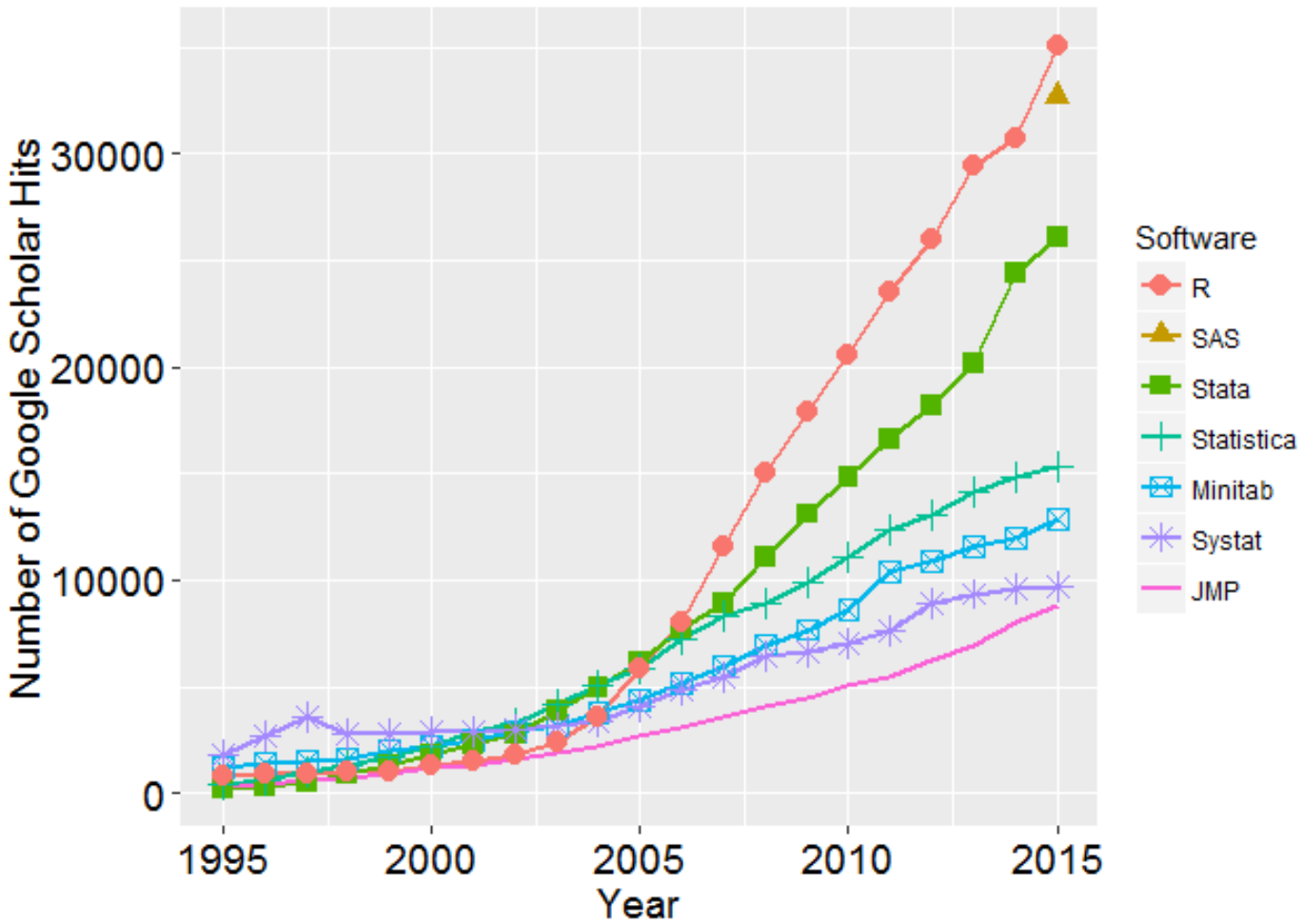
Introduction: R in a nutshell

- What is R? *programming language*, *environment*, *software*...
    - Open source and free
    - Compatibility with other languages
    - Important learning curve (different packages and libraries)
- What is RStudio? Integrated Development Environment that makes R programming more user friendly.
- What can you do with R?
    - Data analysis
    - Dynamic documents
    - Apps

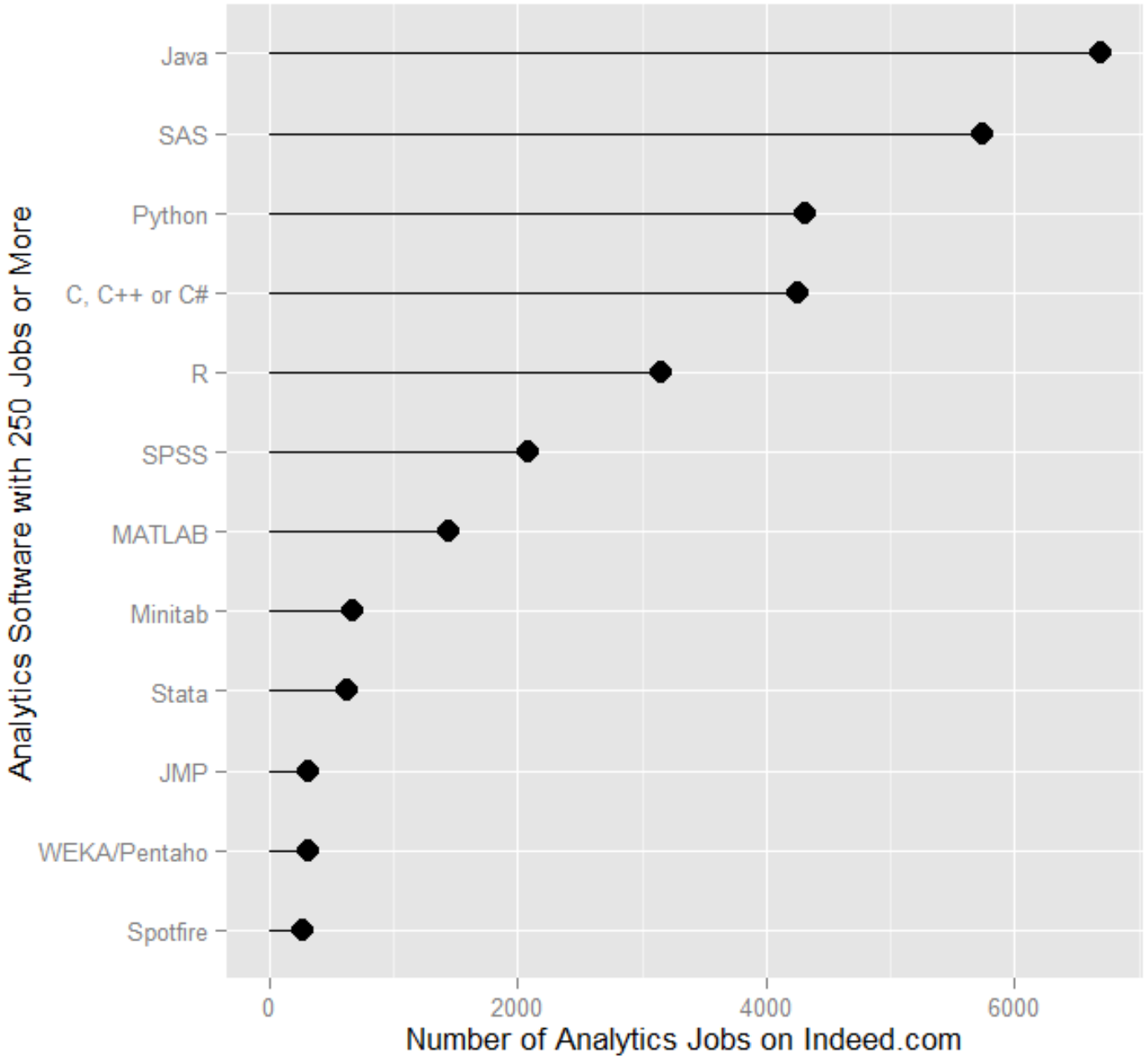
Is R a good investment?



Source: *"The popularity of Data Analysis software"*
Muenchen (2016)



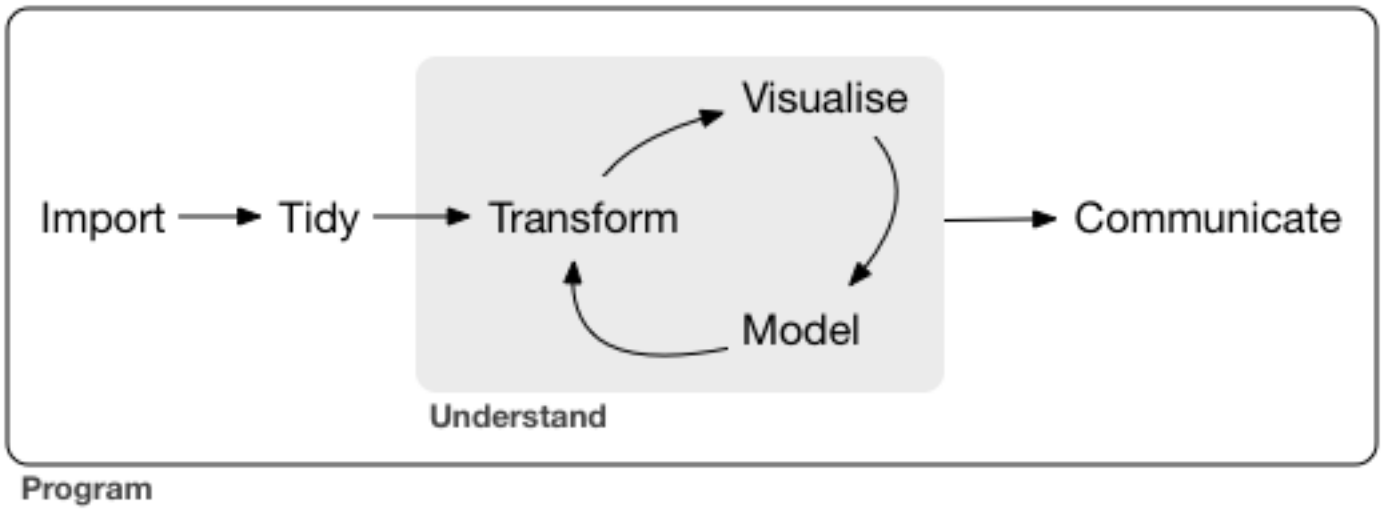
Source: *"The popularity of Data Analysis software"* Muenchen (2016)



Source: *"The popularity of Data Analysis software"* Muenchen (2016)

How do we structure a data analysis project?

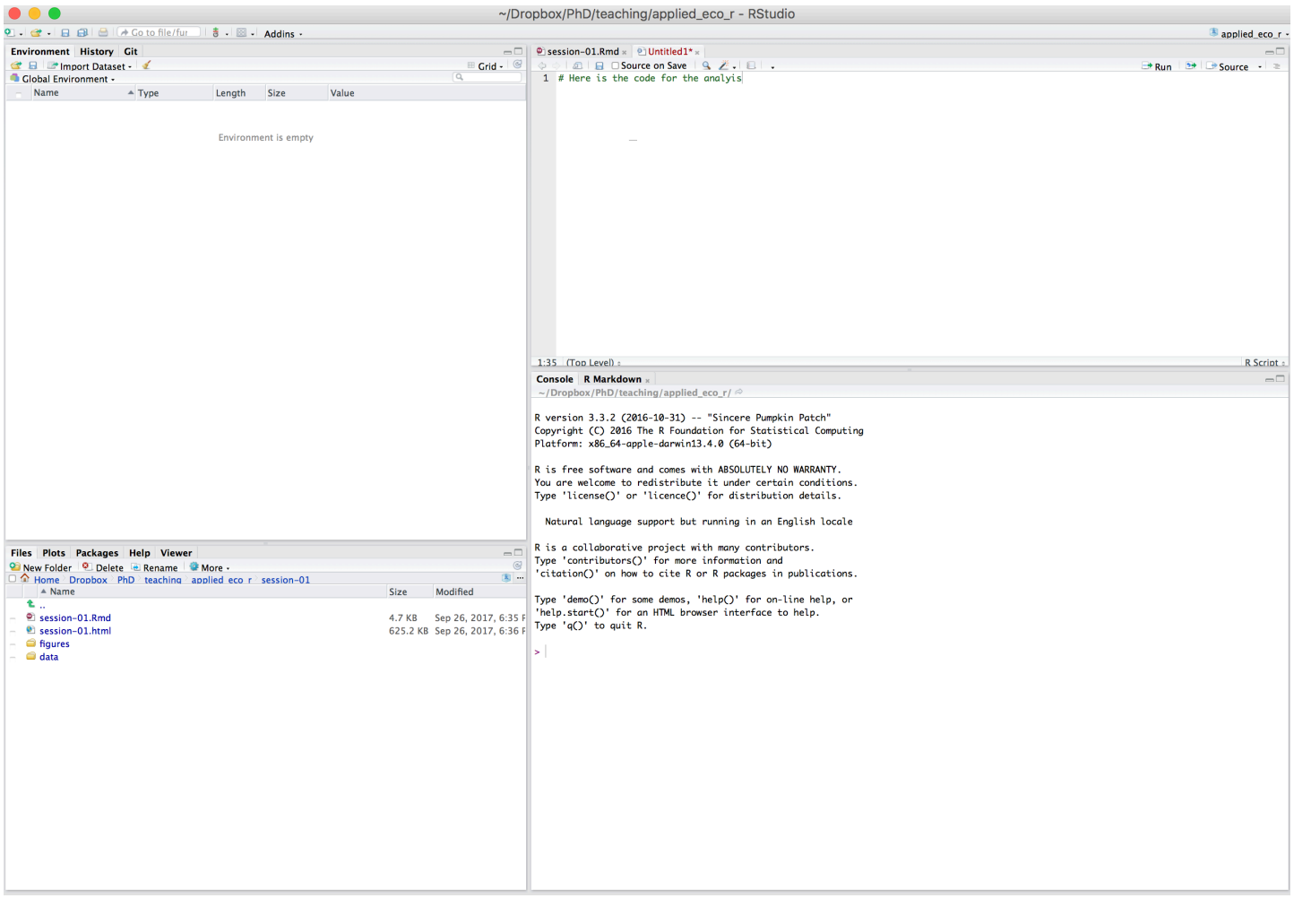
- Steps in a data analysis project



Source: *R for Data Science* (Wickham and Grolemund (2016))

Hello world with R

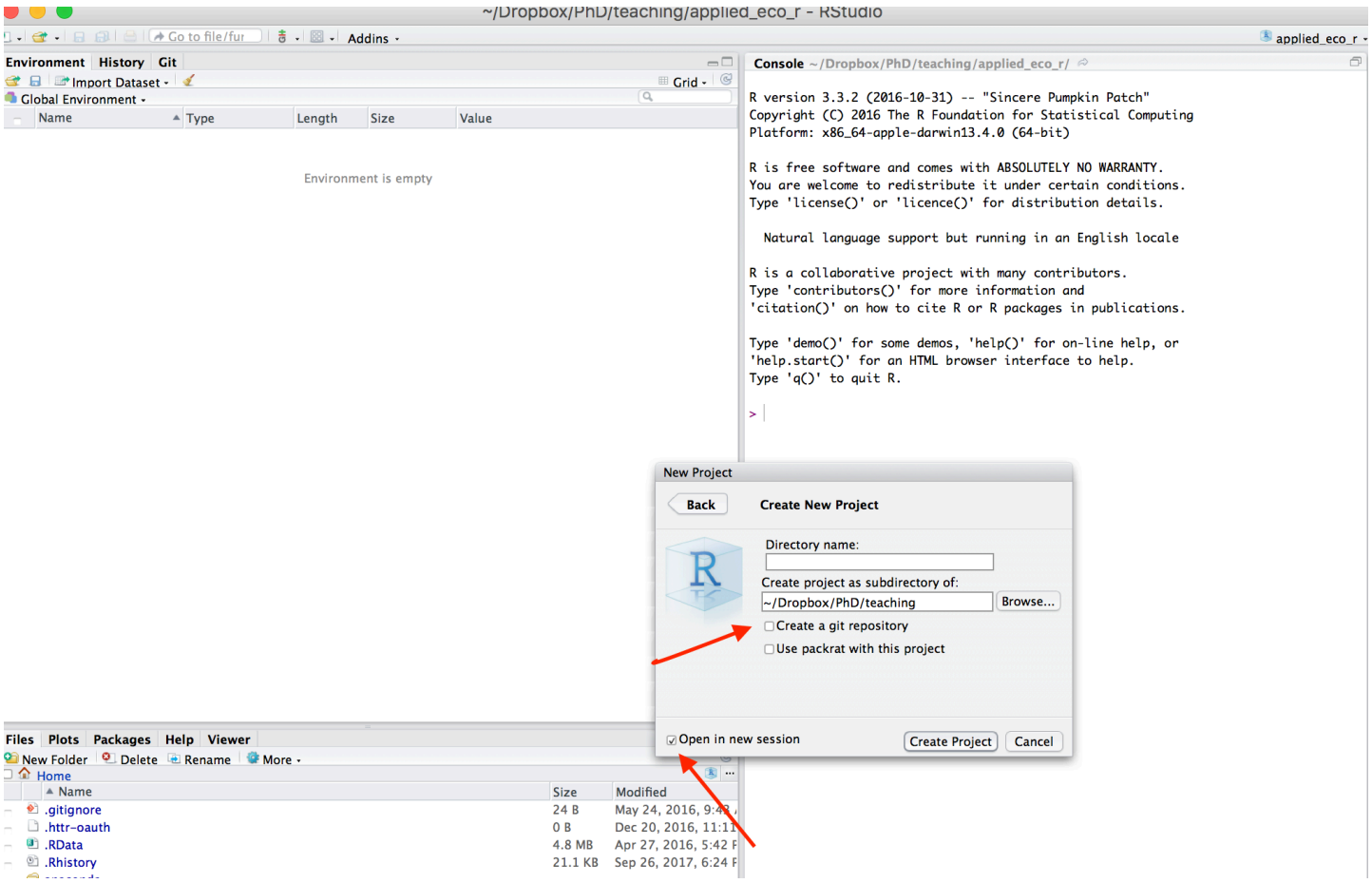
Let´s get our head around...



Create a project

- Projects enable a better organization of the files and keeps a better control of the workflow (e.g. scripts, data, final documents, etc...)
- They improve the efficiency of the workflow.
- They make your life easier
- `Project - New Project - New Directory - Empty project - (select directory...)`

Management of a project

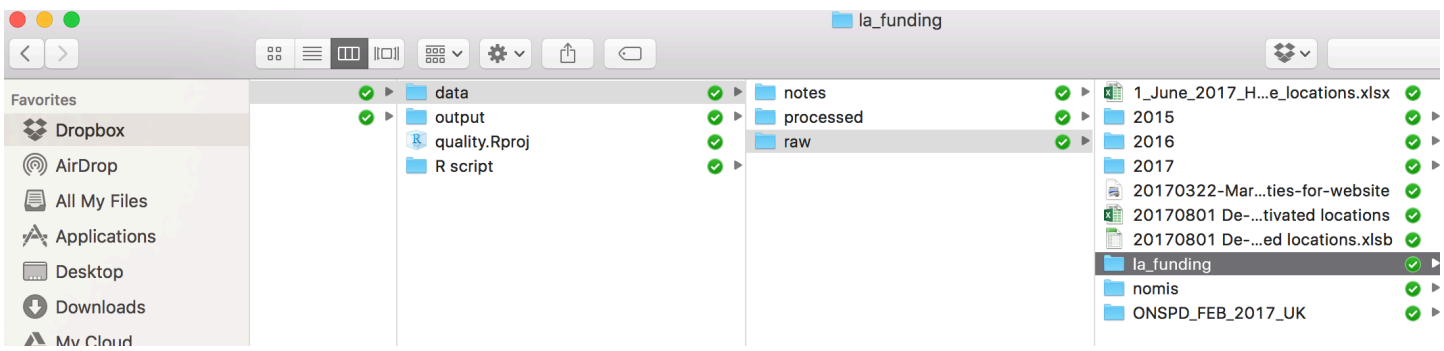


Create a project: Control version (optional)

- It enhances reproducibility and collaboration
- Keeps record and reduces errors
- Limits software dependency
- Back up big data projects

Organisation of the project

- Create different folders with different types of files
    - Data
    - Scripts
    - Outputs



Data types and data structures

Data types and structures

- In R every element is regarded as an object.
- Most general data structures are organised according to two main elements
  - Dimensionality
  - Type of the contents (homogeneous, heterogeneous)
- In most cases, it is necessary to carry out **conversions of objects** in order to meet our needs

Data types

Type	Characteristics	Example
`character`	single letter (or number in some cases)	`"a"`, `"s","34"`
`numeric`	single number	`34`, `5`, `9.8`
`logical`	logical output	`TRUE`, `FALSE`
`integer`	2 (Must add a L at end to denote integer)	`2L`, `5L`
`complex`	complex numbers with real and imaginary parts	`1+4i`

Functions to check some features

- `class()` - what kind of object is it (high-level)?
- `length()` - how long is it? What about two dimensional objects?

Data structures: Vectors

This is the most basic data structure which can be of two types, atomic or list, depending on the type of data contained on it

- Atomic `vectors`

```
v = c(1,2,3,4)
v
```

```
## [1] 1 2 3 4
```

```
z = c("Paul", "Sarah",
z
```

```
## [1] "Paul"  "Sarah" "Joe
```

- Examine the vectors

```
class(v)
```

```
## [1] "numeric"
```

```
length(z)
```

```
## [1] 3
```

```
typeof(z)
```

```
## [1] "character"
```

Data structures: Vectors

- **Add** new attributes to your vector

```
z1 = c(z, "friend1", "friend2")
z1
```

```
## [1] "Paul"    "Sarah"   "Joe"     "friend1" "friend2"
```

- **Replicate** the attributes of your object

```
z2 = rep(z1, 2)
z2
```

```
##  [1] "Paul"    "Sarah"   "Joe"     "friend1" "friend2"
##  [8] "Joe"     "friend1" "friend2"
```

```
z3 = rep(z1, each = 2)
z3
```

```
##  [1] "Paul"    "Paul"    "Sarah"   "Sarah"   "Joe"
##  [8] "friend1" "friend2" "friend2"
```

Data structures: Matrices

These are atomic vectors that have a greater dimmension than 1.

```
m<- matrix(1:6, nrow=2,
m
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

```
x <- 1:3
y <- 10:12

m1  = rbind(x,y)
m1
```

```
##   [,1] [,2] [,3]
## x    1    2    3
## y   10   11   12
```

Data structures: lists

`lists` have various types of data...

```
x <- list(1, "a", TRUE)

x
```

```
## [[1]]
## [1] 1
##
## [[2]]
## [1] "a"
##
## [[3]]
## [1] TRUE
```

... and various dimensions

```
y = list(a = "Mary",
    b = 1:5,
    c = c("Male", "25",

y
```

```
## $a
## [1] "Mary"
##
## $b
## [1] 1 2 3 4 5
##
## $c
## [1] "Male" "25"   "TRUE"
```

Data structures: Data frames

- `Data.frames` are the most common data structure for gathering information.
    - **Variables**: Collect different arguments associated with the information to be analysed
    - **Observations**: Units of analysis (individuals, firms, etc…)
- The structure of a `data.frame` consists of columns that contain labelled variables and rows that contain observations.

Example of a messy dataset

	Active locations for providers registered under the Health and Social Care Act	X__1	X__2	X__3	X__4	X__5	X__6
1	Source: CQC database at 1 June 2017	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*
2	Data Requests Team/Strategy & Intell...	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*
3	Location ID	Location HSCA start date	Care home?	Location Name	Location Telephone Number	Registered manager (note; where the...	Locat
4	1-1000210669	41620	Y	Kingswood House Nursing Home	01424716303	Turner, Patricia	*NA*
5	1-1000312641	41565	N	Human Support Group Limited - Sale	01619429490	Nixon, Yvonne	www
6	1-1000401911	41582	Y	Little Haven	02086974246	Muriuki, Martin	*NA*
7	1-1000587219	41582	Y	Highlands Borders Care Home	01392491261	Martin, Fiona	*NA*
8	1-1000711804	41620	Y	Belmont Grange Nursing and Residen...	01913849853	Shaw, June	*NA*
9	1-1001764404	41558	N	Everycare Midsussex	01444244770	Manville, Katie	www
10	1-1001764472	41575	N	Cherish UK Ltd	01253766888	Stockell, Sam	www
11	1-1001764512	41561	N	Optical Express - Bluewater Clinic	08000232020	Leadley, Robert	www
12	1-1001765343	41561	N	Optical Express - Cambridge Clinic	08000232020	Norman, Elaine	www
13	1-1001875873	41561	N	Optical Express - Leeds (Albion Stree...	08702202020	Saward, Louise	www
14	1-1001876258	41561	N	Optical Express - Liverpool Clinic	08000232020	*	www
15	1-1001899520	41561	N	Optical Express - London (Harley Str...	08000232020	Sutton, Paul	www
16	1-1001900393	41561	N	Aspire Dental Care Ltd - Aylesbury	01296336137	Warren, Jane	*NA*
17	1-1001911451	41561	N	Optical Express - London (Shaftesbur...	08000232020	Coulter, Tiffany	www
18	1-1001911572	41561	N	Optical Express - London (White City)...	08000232020	Dabrowska, Benita	www
19	1-1001911912	41561	N	Aspire Dental Care Ltd - Amersham	*NA*	Warren, Jane	*NA*
20	1-1001921065	41564	Y	Thomas Road	01223514418	Mead, Jackie	*NA*
21	1-1001973807	41561	N	Optical Express - Manchester (Deans...	08000232020	Keegan, Joanne	www
22	1-1002025035	41561	N	Optical Express - Northampton Clinic	08702202020	Spellman, Mary	www
23	1-1002025397	41561	N	Optical Express - Norwich Clinic	08000232020	Spellman, Mary	www
24	1-1002056748	41561	N	Optical Express - Nottingham Clinic	08000232020	Elliott, Judith	www
25	1-1002057300	41561	N	Optical Express - Sheffield (Meadowh...	08000232020	*	www
26	1-1002140522	41556	N	Avant Garde New Eltham	02088501870	*	www
27	1-1002185812	41565	Y	Mayfield Adult Services	01435872201	Watts, Luke	www

Example of a messy dataset

	Active locations for providers registered under the Health and Social Care Act	X__1	X__2	X__3	X__4	X__5	X__6
1	Source: CQC database at 1 June 2017	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*
2	Data Requests Team/Strategy & Intell...	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*
3	Location ID	Location HSCA start date	Care home?	Location Name	Location Telephone Number	Registered manager (note; where the...	Locat
4	1-1000210669	41620	Y	Kingswood House Nursing Home	01424716303	Turner, Patricia	*NA*
5	1-1000312641	41565	N	Human Support Group Limited - Sale	01619429490	Nixon, Yvonne	www
6	1-1000401911	41582	Y	Little Haven	02086974246	Muriuki, Martin	*NA*
7	1-1000587219	41582	Y	Highlands Borders Care Home	01392491261	Martin, Fiona	*NA*
8	1-1000711804	41620	Y	Belmont Grange Nursing and Residen...	01913849853	Shaw, June	*NA*
9	1-1001764404	41558	N	Everycare Midsussex	01444244770	Manville, Katie	www
10	1-1001764472	41575	N	Cherish UK Ltd	01253766888	Stockell, Sam	www
11	1-1001764512	41561	N	Optical Express - Bluewater Clinic	08000232020	Leadley, Robert	www
12	1-1001765343	41561	N	Optical Express - Cambridge Clinic	08000232020	Norman, Elaine	www
13	1-1001875873	41561	N	Optical Express - Leeds (Albion Stree...	08702202020	Saward, Louise	www
14	1-1001876258	41561	N	Optical Express - Liverpool Clinic	08000232020	*	www
15	1-1001899520	41561	N	Optical Express - London (Harley Str...	08000232020	Sutton, Paul	www
16	1-1001900393	41561	N	Aspire Dental Care Ltd - Aylesbury	01296336137	Warren, Jane	*NA*
17	1-1001911451	41561	N	Optical Express - London (Shaftesbur...	08000232020	Coulter, Tiffany	www
18	1-1001911572	41561	N	Optical Express - London (White City)...	08000232020	Dabrowska, Benita	www

Tidy(er) data frame

```
##   Name AGE GenDER        Region   ID.code Treated.
## 1 Mary  12      0    North East   A-00345      yes
## 2 John  25      1    North East A-1243009       no
## 3 Tony  20      1 East Midlands A-0012456      yes
```

- Can we make it cleaner?

```
library(janitor)


df_clean = clean_names(df)

df_clean
```

```
##   name age gender        region   id_code treated
## 1 Mary  12      0    North East   A-00345     yes
## 2 John  25      1    North East A-1243009      no
## 3 Tony  20      1 East Midlands A-0012456     yes
```

Explore the data frame

- Look at the first and last rows

```
head(df_clean,1)
```

```
##   name age gender     region id_code treated
## 1 Mary  12      0 North East A-00345     yes
```

```
tail(df_clean,2)
```

```
##   name age gender         region   id_code treated
## 2 John  25      1    North East A-1243009      no
## 3 Tony  20      1 East Midlands A-0012456     yes
```

Explore the data frame

- Have a complete vision of the `data.frame`

```
library(dplyr)
glimpse(df_clean)
```

```
## Observations: 3
## Variables: 6
## $ name    <fctr> Mary, John, Tony
## $ age     <dbl> 12, 25, 20
## $ gender  <fctr> 0, 1, 1
## $ region  <fctr> North East, North East, East Midlands
## $ id_code <fctr> A-00345, A-1243009, A-0012456
## $ treated <fctr> yes, no, yes
```

Tibbles

Tibbles are a new form of expressing data frames.

- Are more efficient
- Printing: They print first ten rows and all the columns that fit on one screen - good when dealing with big data frames
- Easier subsetting

```
as_tibble(df)
```

```
## # A tibble: 3 x 6
##     Name   AGE GenDER        Region   ID.code Treated.
##   <fctr> <dbl> <fctr>        <fctr>    <fctr>   <fctr>
## 1   Mary    12      0    North East   A-00345      yes
## 2   John    25      1    North East A-1243009       no
## 3   Tony    20      1 East Midlands A-0012456      yes
```

Exercise

How can your create a data.frame with the individuals in `z` and three more friends who are followed during 5 periods of time?

```
##       id time
## 1   Paul    1
## 2   Paul    2
## 3   Paul    3
## 4   Paul    4
## 5   Paul    5
## 6 Sarah    1
```

- One way with `data.frame()`

```
id = rep(c(z, "Beth",
           "Mike",
           "Martha"), e

time = rep(1:5, 6)


friends = data.frame(ic

head(friends)
```

```
##       id time
## 1   Paul    1
## 2   Paul    2
## 3   Paul    3
## 4   Paul    4
## 5   Paul    5
## 6 Sarah    1
```

- Another way with `cbind()`

```
id = rep(c(z, "Beth",
           "Mike",
           "Martha"), e

time = rep(1:5, 6)


friends = cbind(id, tim

head(friends)
```

```
##       id      time
## [1,] "Paul"  "1"
## [2,] "Paul"  "2"
## [3,] "Paul"  "3"
## [4,] "Paul"  "4"
## [5,] "Paul"  "5"
## [6,] "Sarah" "1"
```

```
# Data frame

friends = as.data.frame
head(friends, 2)
```

```
##     id time
## 1 Paul    1
## 2 Paul    2
```

Thanks!

@EdudinGonzalo

e.gonzalo-almorox@newcastle.ac.uk