STATISTICAL ANALYSIS AND DOCUMENT MINING

Introduction to R and RStudio

Author: Pedro L. C. RODRIGUES – February 2022

Slides based on materials developed by Charlotte LACLAU and Vasilii FEOFANOV

What is the best programming language for data science ?

R
Python
Python
R
Julia
C

What is the best programming language for data science ?

- All languages have their **advantages** and **disadvantages**

- A good language is one that help you achieve **your own goals**

- There are many **bindings** between languages too:

  - Python <-> R
  - Python <-> Julia
  - Python <-> C

What is the best programming language for data science ?

Some use cases…

“I want to do research in bioinformatics like microarray data analysis, genetics, epidemeology”

↳ R has many relevant and state of the art packages for it, made by researchers

What is the best programming language for data science ?

Some use cases…

“I want to do research in bioinformatics like microarray data analysis, genetics, epidemeology”

↳ R has many relevant and state of the art packages for it, made by researchers

“I want to do make a data crawler to gather iMDB reviews and then run a clustering algorithm”

↳ Python is the most popular programming language in applied research

What is the best programming language for data science ?

Some use cases…

“I want to do research in bioinformatics like microarray data analysis, genetics, epidemeology”

↳ R has many relevant and state of the art packages for it, made by researchers

“I want to do make a data crawler to gather iMDB reviews and then run a clustering algorithm”

↳ Python is the most popular programming language in applied research

“I want to work in a BIG company with BIG data making BIG money”

↳ Scala and Spark are probably the most suitable for you

In this tutorial, we will be focusing on R…

What is R

- Manipulation of dataframes
- Calculus, statistics, optimization, etc.
- Data vizualisation

Some characteristics…

- Interpreted language
- Based on vectorization
- No variable declaration

What is R

- Manipulation of dataframes
- Calculus, statistics, optimization, etc.
- Data vizualisation

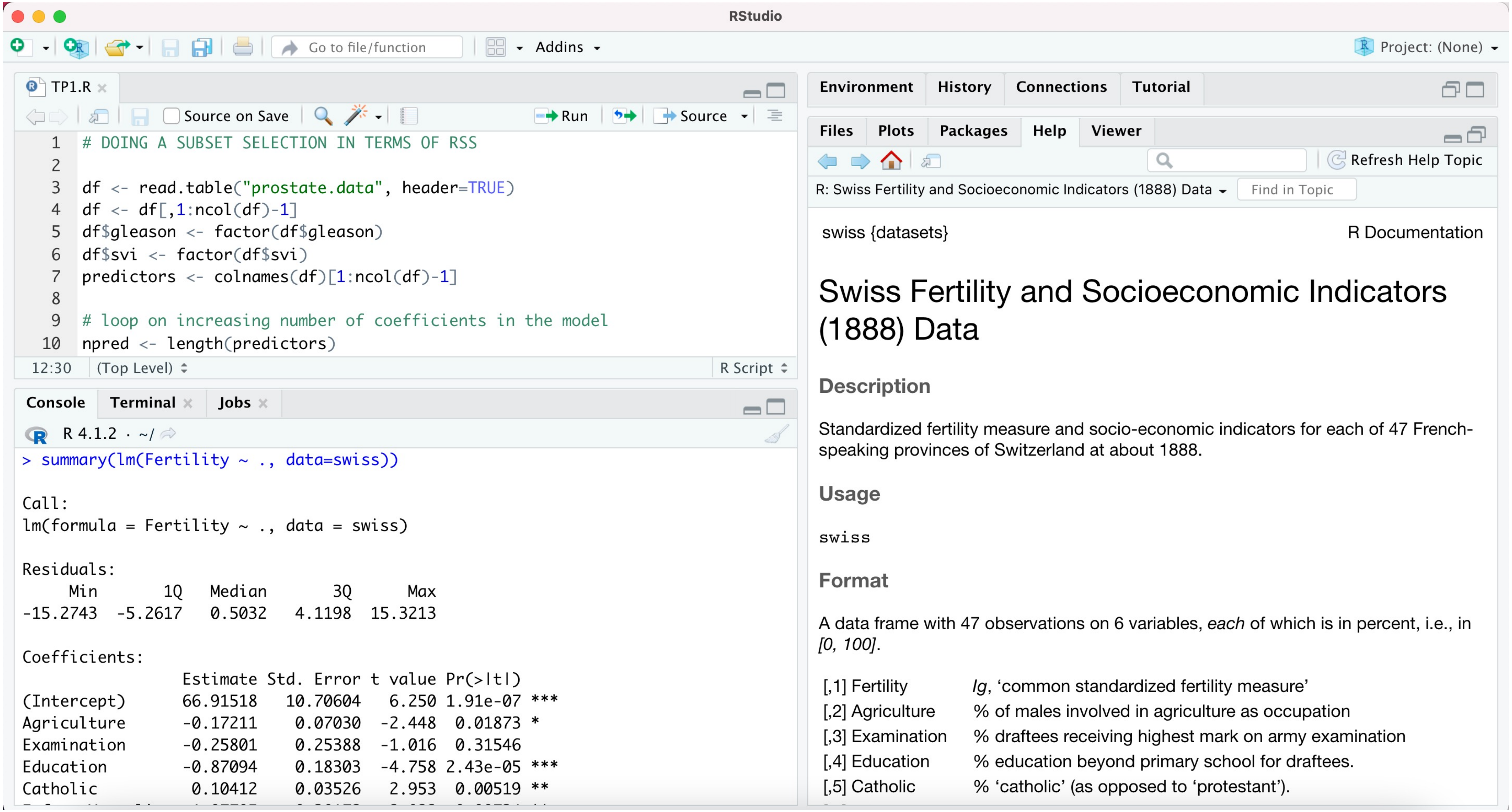
Some characteristics…

- Interpreted language
- Based on vectorization
- No variable declaration

How to use R

- Via command-line
- Via Rstudio

RStudio®



```r
# DOING A SUBSET SELECTION IN TERMS OF RSS

df <- read.table("prostate.data", header=TRUE)
df <- df[,1:ncol(df)-1]
df$gleason <- factor(df$gleason)
df$svi <- factor(df$svi)
predictors <- colnames(df)[1:ncol(df)-1]

# loop on increasing number of coefficients in the model
npred <- length(predictors)
```

```
> summary(lm(Fertility ~ ., data=swiss))

Call:
lm(formula = Fertility ~ ., data = swiss)

Residuals:
     Min       1Q   Median       3Q      Max 
-15.2743  -5.2617   0.5032   4.1198  15.3213 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)      66.91518   10.70604   6.250 1.91e-07 ***
Agriculture      -0.17211    0.07030  -2.448  0.01873 *  
Examination      -0.25801    0.25388  -1.016  0.31546    
Education        -0.87094    0.18303  -4.758 2.43e-05 ***
Catholic          0.10412    0.03526   2.953  0.00519 ** 
```

swiss {datasets} R Documentation

Swiss Fertility and Socioeconomic Indicators (1888) Data

Description

Standardized fertility measure and socio-economic indicators for each of 47 French-speaking provinces of Switzerland at about 1888.

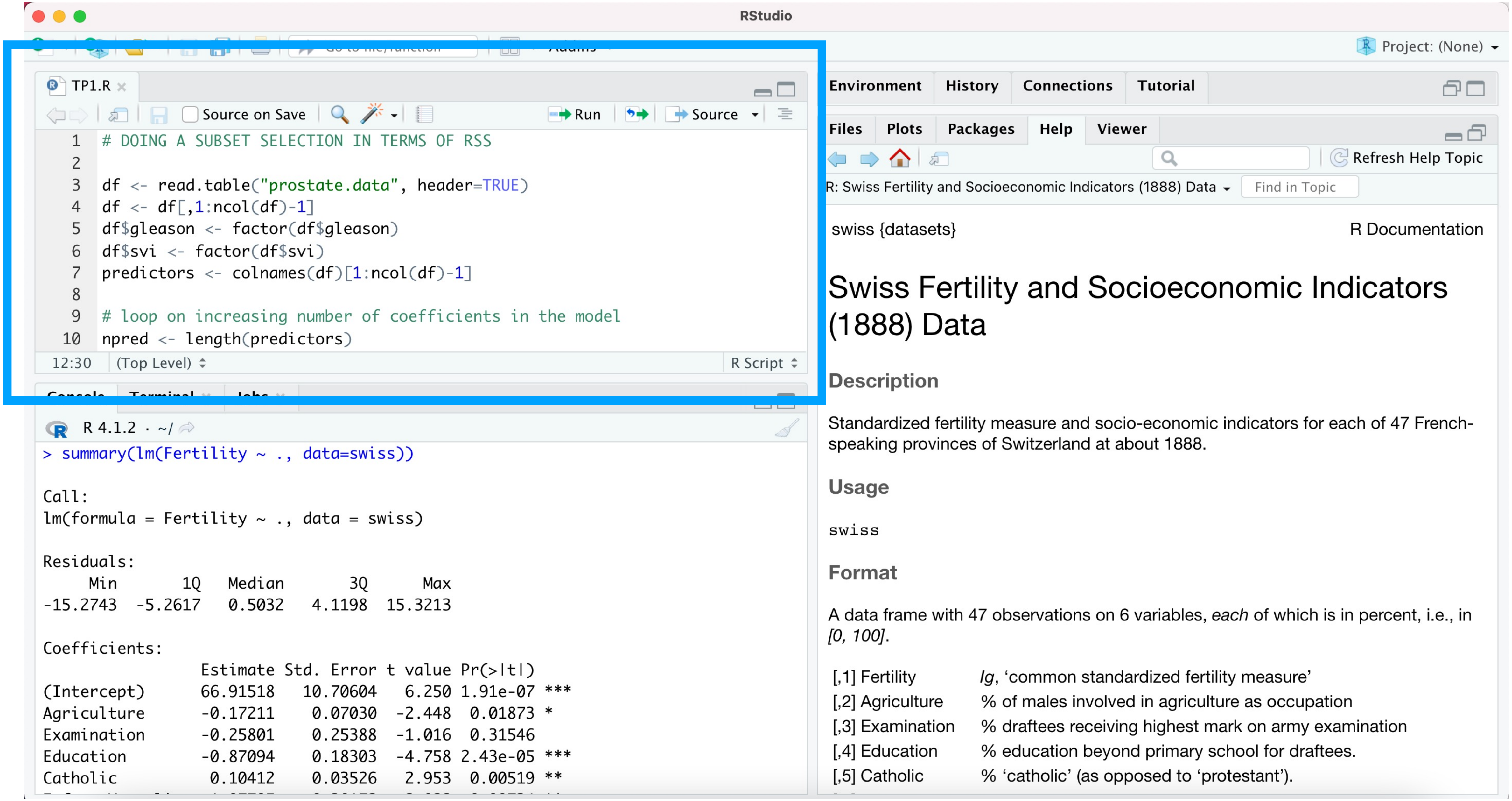
Usage

```
swiss
```

Format

A data frame with 47 observations on 6 variables, *each* of which is in percent, i.e., in *[0, 100]*.

[,1] Fertility	*Ig*, ‘common standardized fertility measure’
[,2] Agriculture	% of males involved in agriculture as occupation
[,3] Examination	% draftees receiving highest mark on army examination
[,4] Education	% education beyond primary school for draftees.
[,5] Catholic	% ‘catholic’ (as opposed to ‘protestant’).



```
# DOING A SUBSET SELECTION IN TERMS OF RSS

df <- read.table("prostate.data", header=TRUE)
df <- df[,1:ncol(df)-1]
df$gleason <- factor(df$gleason)
df$svi <- factor(df$svi)
predictors <- colnames(df)[1:ncol(df)-1]

# loop on increasing number of coefficients in the model
npred <- length(predictors)
```

```
> summary(lm(Fertility ~ ., data=swiss))

Call:
lm(formula = Fertility ~ ., data = swiss)

Residuals:
     Min       1Q   Median       3Q      Max 
-15.2743  -5.2617   0.5032   4.1198  15.3213 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)      66.91518   10.70604   6.250 1.91e-07 ***
Agriculture      -0.17211    0.07030  -2.448  0.01873 *  
Examination      -0.25801    0.25388  -1.016  0.31546    
Education        -0.87094    0.18303  -4.758 2.43e-05 ***
Catholic          0.10412    0.03526   2.953  0.00519 ** 
```

swiss {datasets} R Documentation

Swiss Fertility and Socioeconomic Indicators (1888) Data

Description

Standardized fertility measure and socio-economic indicators for each of 47 French-speaking provinces of Switzerland at about 1888.

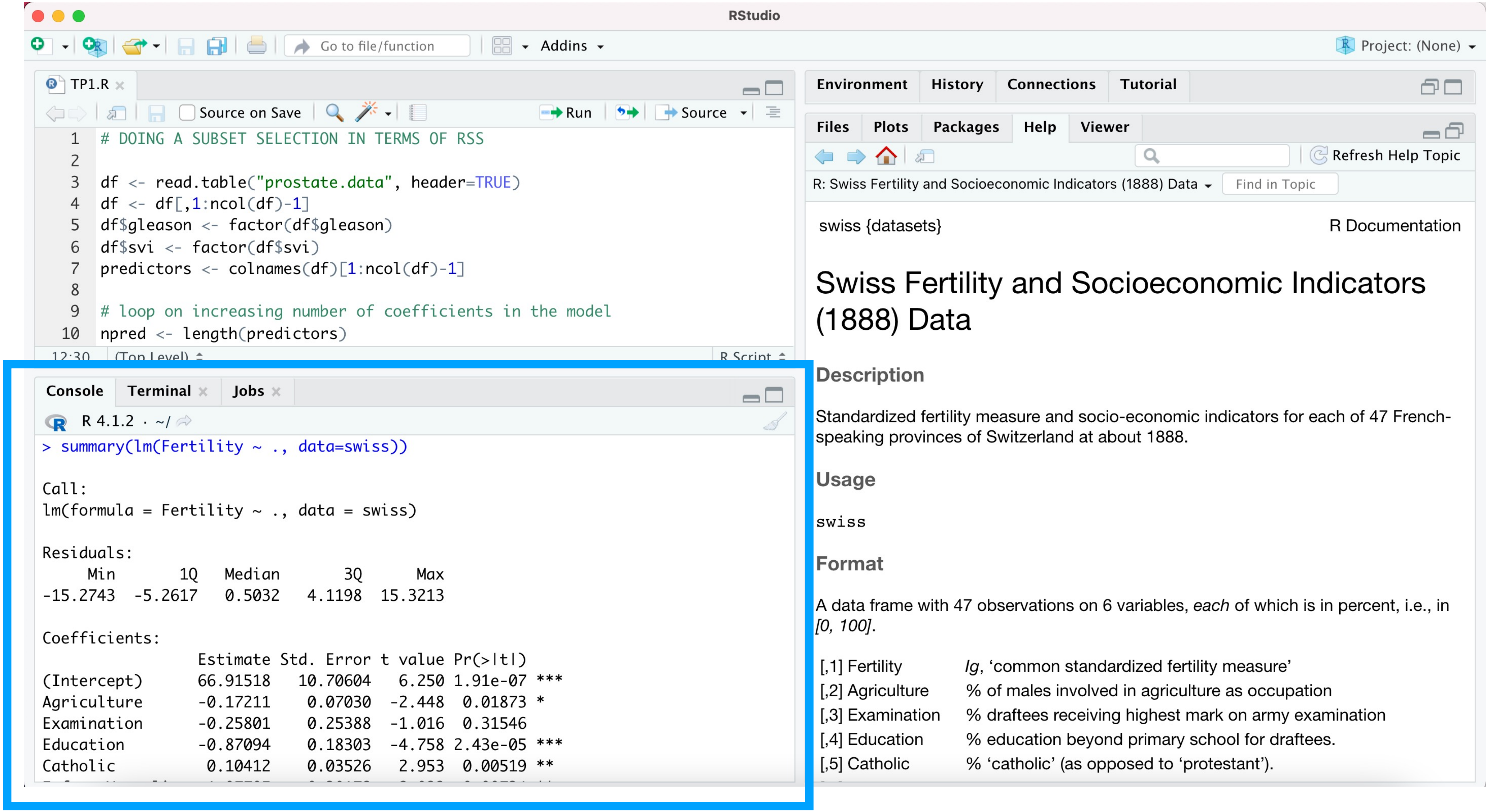
Usage

```
swiss
```

Format

A data frame with 47 observations on 6 variables, *each* of which is in percent, i.e., in *[0, 100]*.

[,1] Fertility	*Ig*, ‘common standardized fertility measure’
[,2] Agriculture	% of males involved in agriculture as occupation
[,3] Examination	% draftees receiving highest mark on army examination
[,4] Education	% education beyond primary school for draftees.
[,5] Catholic	% ‘catholic’ (as opposed to ‘protestant’).



```
# DOING A SUBSET SELECTION IN TERMS OF RSS

df <- read.table("prostate.data", header=TRUE)
df <- df[,1:ncol(df)-1]
df$gleason <- factor(df$gleason)
df$svi <- factor(df$svi)
predictors <- colnames(df)[1:ncol(df)-1]

# loop on increasing number of coefficients in the model
npred <- length(predictors)
```

```
> summary(lm(Fertility ~ ., data=swiss))

Call:
lm(formula = Fertility ~ ., data = swiss)

Residuals:
     Min       1Q   Median       3Q      Max 
-15.2743  -5.2617   0.5032   4.1198  15.3213 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)      66.91518   10.70604   6.250 1.91e-07 ***
Agriculture      -0.17211    0.07030  -2.448  0.01873 *  
Examination      -0.25801    0.25388  -1.016  0.31546    
Education        -0.87094    0.18303  -4.758 2.43e-05 ***
Catholic          0.10412    0.03526   2.953  0.00519 ** 
```

swiss {datasets}   R Documentation

Swiss Fertility and Socioeconomic Indicators (1888) Data

Description

Standardized fertility measure and socio-economic indicators for each of 47 French-speaking provinces of Switzerland at about 1888.

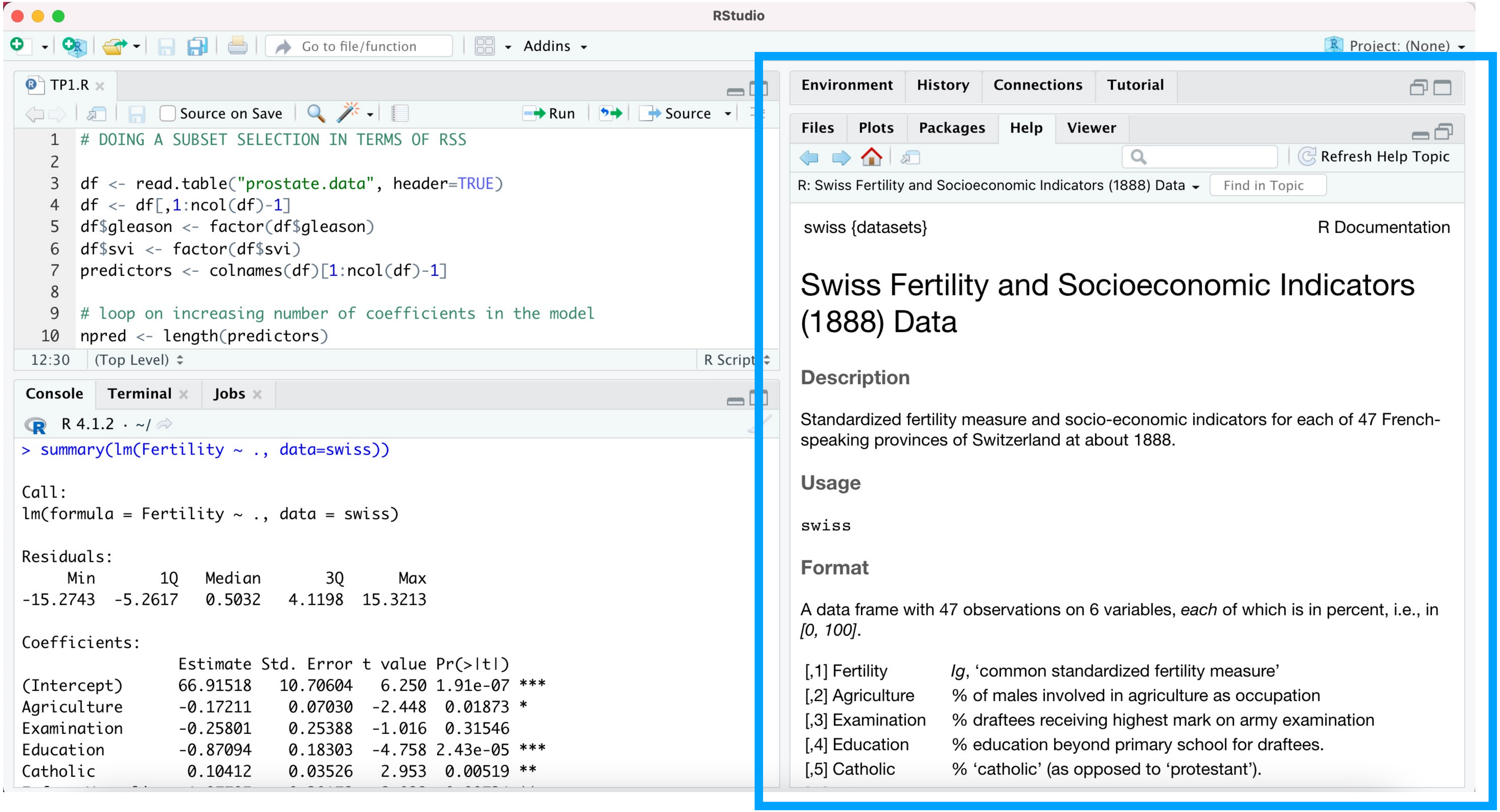
Usage

```
swiss
```

Format

A data frame with 47 observations on 6 variables, *each* of which is in percent, i.e., in *[0, 100]*.

[,1]	Fertility	*Ig*, ‘common standardized fertility measure’
[,2]	Agriculture	% of males involved in agriculture as occupation
[,3]	Examination	% draftees receiving highest mark on army examination
[,4]	Education	% education beyond primary school for draftees.
[,5]	Catholic	% ‘catholic’ (as opposed to ‘protestant’).



```
# DOING A SUBSET SELECTION IN TERMS OF RSS

df <- read.table("prostate.data", header=TRUE)
df <- df[,1:ncol(df)-1]
df$gleason <- factor(df$gleason)
df$svi <- factor(df$svi)
predictors <- colnames(df)[1:ncol(df)-1]

# loop on increasing number of coefficients in the model
npred <- length(predictors)
```

```
> summary(lm(Fertility ~ ., data=swiss))

Call:
lm(formula = Fertility ~ ., data = swiss)

Residuals:
     Min       1Q   Median       3Q      Max 
-15.2743  -5.2617   0.5032   4.1198  15.3213 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)      66.91518   10.70604   6.250 1.91e-07 ***
Agriculture      -0.17211    0.07030  -2.448  0.01873 *  
Examination      -0.25801    0.25388  -1.016  0.31546    
Education        -0.87094    0.18303  -4.758 2.43e-05 ***
Catholic          0.10412    0.03526   2.953  0.00519 ** 
```

swiss {datasets} R Documentation

Swiss Fertility and Socioeconomic Indicators (1888) Data

Description

Standardized fertility measure and socio-economic indicators for each of 47 French-speaking provinces of Switzerland at about 1888.

Usage

```
swiss
```

Format

A data frame with 47 observations on 6 variables, *each* of which is in percent, i.e., in *[0, 100]*.

[,1] Fertility	*Ig*, ‘common standardized fertility measure’
[,2] Agriculture	% of males involved in agriculture as occupation
[,3] Examination	% draftees receiving highest mark on army examination
[,4] Education	% education beyond primary school for draftees.
[,5] Catholic	% ‘catholic’ (as opposed to ‘protestant’).

A first script with RStudio

- Launch *Rstudio* on your machine
- Create a new *R* script
- Write the following lines and save the file

```r
ls()
pi
v <- c(1, 5, 8)
v * 2
x <- v + c(2, 1, 7)
x
ls()
```

- Run the code

The working directory

The working directory is a default folder where Rstudio looks for:

- files and/or data
- save the workspace in .Rdata

It is very important to set the working directory each time we run a new script or open a new session

```r
# Get the path of the working directory
getwd()
# Set a new working directory
setwd("/Users/plcrodrigues/Courses/ENSIMAG/SADM/Week 02/OH")
```

Help

To get info about a function you can run

```
?lm
#or
help(lm)
```

To see an example using this function you can run

```
example(lm)
```

You will also find a lot of helpful posts at stats.stackexchange and stackoverflow

R markdown

This is a tool for reproducible documents with integrated R commands and much more

- Files with extension **.rmd**
- Need to install the package **rmarkdown**
- Automate the report of the TP
- Reproducibility of the TP results
- Formatted output in html, pdf, etc.

Using R for data manipulation

Objects in R can be variables, tables, data frames, functions, text, formulae, etc.

- Names of objects always start with a letter and may be followed by digits and/or dots
- Names of objects are case-sensitive

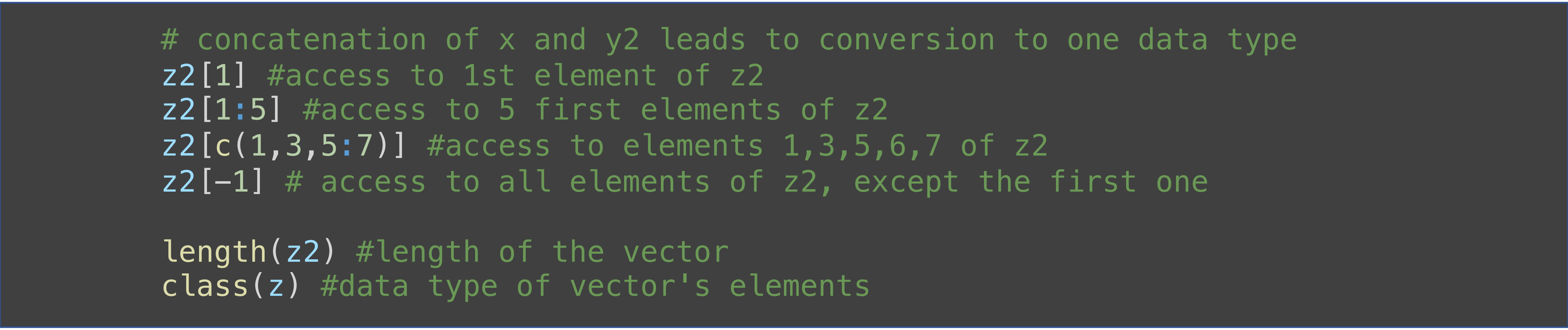
Data can be of many different types

- **Vector**: a vector of fixed size data of the same type
- **List**: an ordered collection of objects which can be of different types
- **Matrix**: a table of dimension two
- **Array**: a table of dimension larger than two
- **Data frame**: a matrix with columns that can be of different types

Using R for data manipulation

```r
x<-3 # Scalar
x # Display x
y<-c(5,2,3) # Vector
y # Display y

x <- rep(0,15) # vector of 0 of size 15
x <- rep(F,7) # a boolean vector of FALSE of size 7

x <- 1:15 # integer values from 1 to 15
x <- 0.25:12 # values from 0.25 with increment=1 until sup=12
x <- seq(from=0,to=1,by=0.1) # values from 0 to 1 with a step=0.1

x <- -2:2
y <- rep(1:3,4)
z<-c(x,y) #z is x appended by y

y2 <- c("a", "b", "c", "d")
z2 <- c(x, y2)
```

Using R for data manipulation



```
# concatenation of x and y2 leads to conversion to one data type
z2[1] #access to 1st element of z2
z2[1:5] #access to 5 first elements of z2
z2[c(1,3,5:7)] #access to elements 1,3,5,6,7 of z2
z2[-1] # access to all elements of z2, except the first one

length(z2) #length of the vector
class(z) #data type of vector's elements
```

Using R for data manipulation

```r
# concatenation of x and y2 leads to conversion to one data type
z2[1] #access to 1st element of z2
z2[1:5] #access to 5 first elements of z2
z2[c(1,3,5:7)] #access to elements 1,3,5,6,7 of z2
z2[-1] # access to all elements of z2, except the first one

length(z2) #length of the vector
class(z) #data type of vector's elements
```

Working with matrices

```r
x1 = seq(5,25,5)
x2 = c(-3,4.5,2,18,0)
x = cbind(x1,x2); #column bind
y = rbind(x1,x2); #row bind
```

```r
M = matrix(1:12,3,4)
M = matrix(1:12,3,4,T)
length(M) # total number of elements
dim(M) # matrix size
nrow(M) # number of rows
ncol(M) # number of columns
```

Using R for data manipulation

Accessing elements of a matrix

```
M[1,1] # element lying on the intersection of 1st row and 1st column
M[2,] # all elements of the 2nd row
M[,3] # all elements of the 3rd column
M[1:3,c(1,3,5:7)]
M[-1,-2] # everything except 1st row and 2nd column
```

Accessing elements satisfying a certain set of conditions

```
M[M[,2]==2,]
M[M[,2]>3 & M[,4]==8,]
```

Using R for data manipulation

Creating a list

```r
lis <- list(firstname = "John",
       lastname = "Smith",
       age = 35,
       childAges = c(3, 5, 9))
lis$firstname # access to an element of lis
names(lis) # names of elements of the list
```

Creating a dataframe

```r
df <- data.frame(age = c(15,20), name = c("paul","jean"),
       row.names = c("I1", "I2"))
df[,2] # Access to the second variable of df
df$name
```

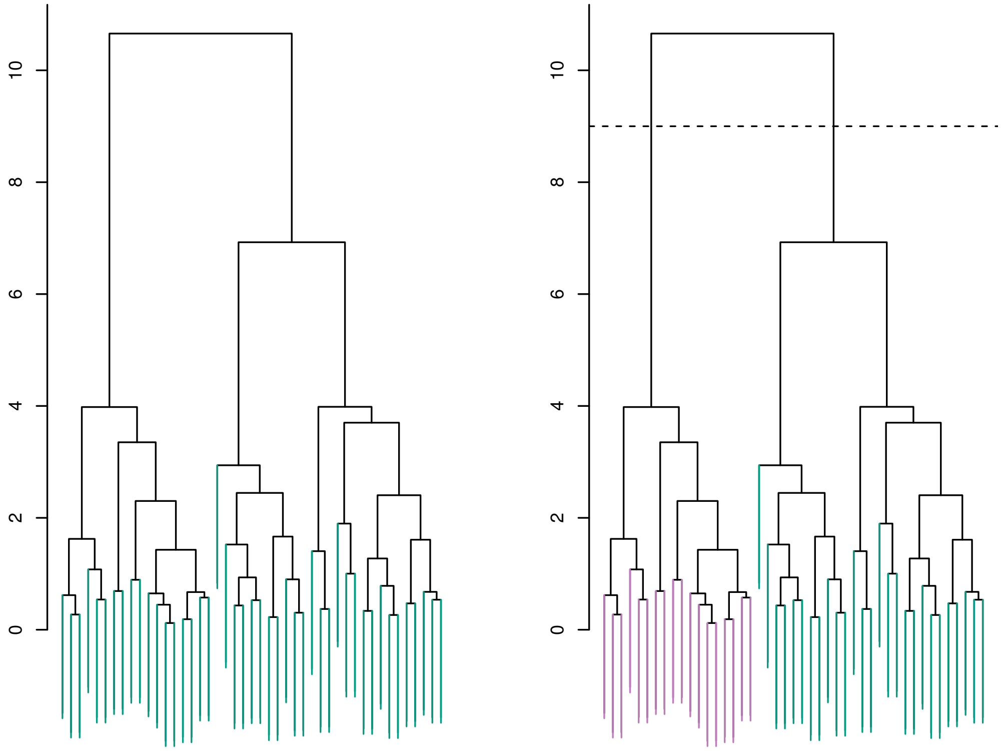
Using R for data manipulation

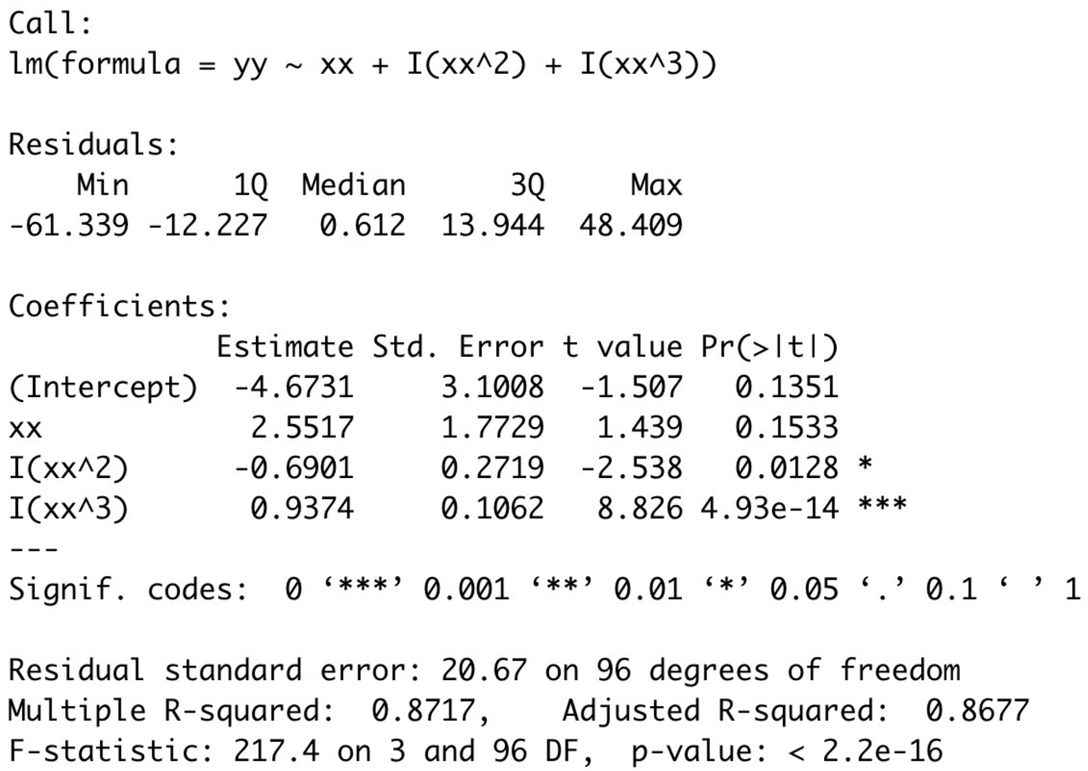
- In R, missing values are indicated by “NA” (not available)
- To check if an object contains any missing values, run: `is.na(x)`
- Some functions can’t handle objects with missing values. To prevent problems, these functions usually have an argument saying whether NA should be ignored or not

```r
# Compute mean
mean(airquality$Ozone, na.rm=TRUE)
```

- Functions like `is._()` test whether an object is of a certain type and output T/F
- Functions like `as._()` convert the type of an object to a different one

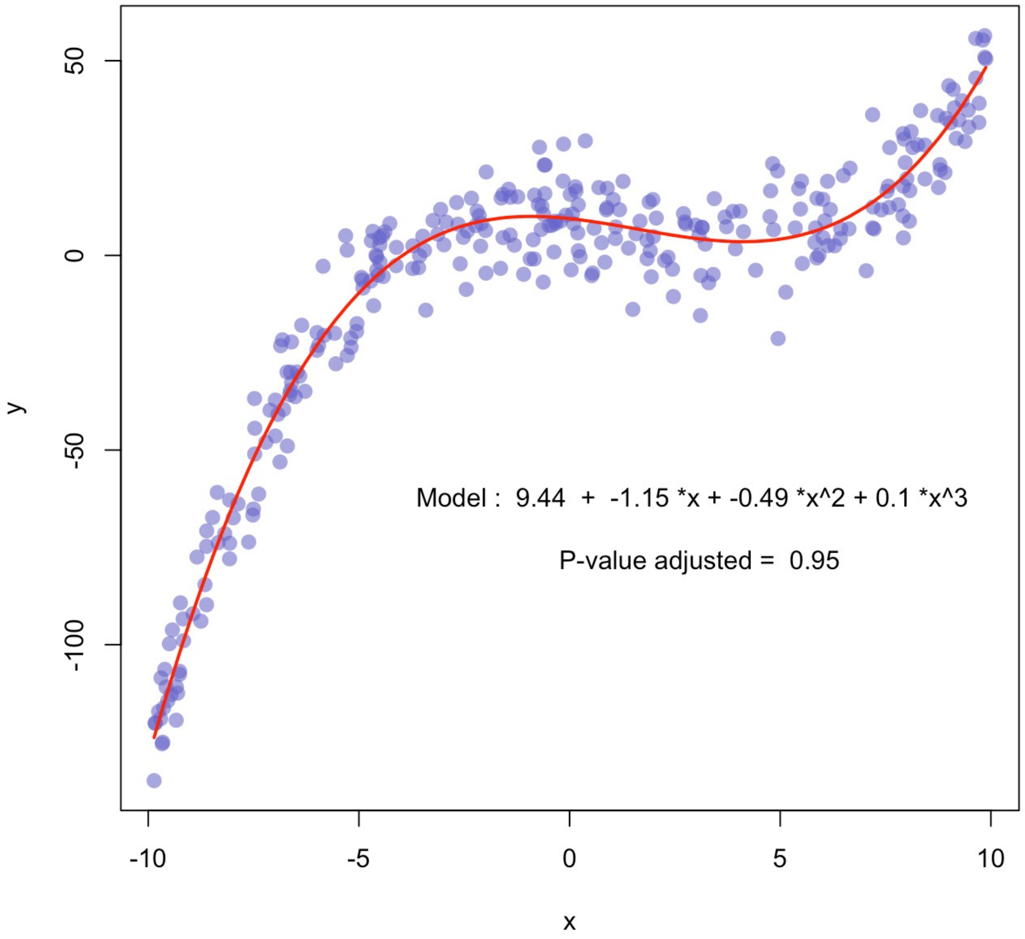
Using R for data analysis

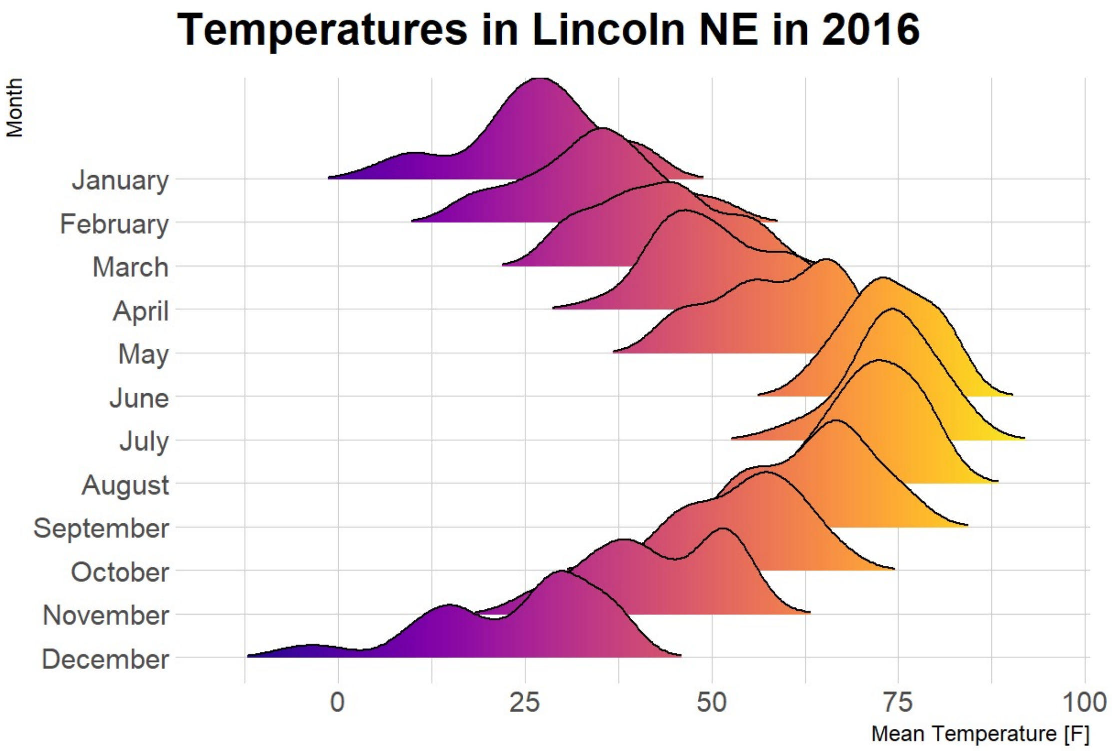




```
Call:
lm(formula = yy ~ xx + I(xx^2) + I(xx^3))

Residuals:
    Min      1Q  Median      3Q     Max 
-61.339 -12.227   0.612  13.944  48.409 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -4.6731     3.1008  -1.507   0.1351    
xx            2.5517     1.7729   1.439   0.1533    
I(xx^2)      -0.6901     0.2719  -2.538   0.0128 *  
I(xx^3)       0.9374     0.1062   8.826 4.93e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 20.67 on 96 degrees of freedom
Multiple R-squared:  0.8717,	Adjusted R-squared:  0.8677 
F-statistic: 217.4 on 3 and 96 DF,  p-value: < 2.2e-16
```

Using R for data visualization





Let’s do an example for real …