Basic Concepts of the R Language

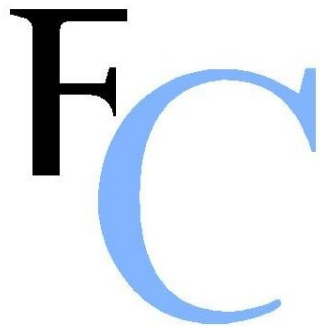
Solutions to Hands On Exercises

L. Torgo

ltorgo@fc.up.pt

Departamento de Ciência de Computadores / Faculdade de Ciências
Universidade do Porto

Feb, 2021



Hands On 1

A survey was carried out on several countries to find out the average price of a certain product, with the following resulting data:

Portugal	Spain	Italy	France	Germany	Greece	UK	Finland	Belgium	Austria
10.3	10.6	11.5	12.3	9.9	9.3	11.4	10.9	12.1	9.1

1. What is the adequate data structure to store these values?
2. Create a variable with this data, taking full advantage of R facilities in order to facilitate the access to the information. solution
3. Obtain another vector with the prices after VAT. solution
4. Which countries have prices above 10?
5. Which countries have prices above the average? solution
6. Which countries have prices between 10 and 11 euros?
7. How would you raise the prices by 10%? solution
8. How would you decrease by 2.5%, the prices of the countries with price above the average? solution

Solutions to Exercises 1 and 2

Portugal	Spain	Italy	France	Germany	Greece	UK	Finland	Belgium	Austria
10.3	10.6	11.5	12.3	9.9	9.3	11.4	10.9	12.1	9.1

- What is the adequate data structure to store these values?
  Answer : **A vector**
- Create a variable with this data, taking full advantage of R facilities in order to facilitate the access to the information.

```
prices <- c(pt=10.3,es=10.6,it=11.5,fr=12.3,de=9.9,
            gr=9.3,uk=11.4,fi=10.9,be=12.1,au=9.1)
prices
##   pt   es   it   fr   de   gr   uk   fi   be   au
## 10.3 10.6 11.5 12.3  9.9  9.3 11.4 10.9 12.1  9.1
```

Go Back

Solutions to Exercise 3

```
prices

##   pt   es   it   fr   de   gr   uk   fi   be   au
## 10.3 10.6 11.5 12.3  9.9  9.3 11.4 10.9 12.1  9.1
```

- Obtain another vector with the prices after VAT.

```
prices*1.23

##    pt    es    it    fr    de    gr    uk    fi    be    au
## 12.67 13.04 14.14 15.13 12.18 11.44 14.02 13.41 14.88 11.19
```

or if we wish to store the result,

```
pricesVAT <- prices*1.23
pricesVAT

##    pt    es    it    fr    de    gr    uk    fi    be    au
## 12.67 13.04 14.14 15.13 12.18 11.44 14.02 13.41 14.88 11.19
```

Solutions to Exercises 4 and 5

```
prices

##   pt   es   it   fr   de   gr   uk   fi   be   au
## 10.3 10.6 11.5 12.3  9.9  9.3 11.4 10.9 12.1  9.1
```

- Which countries have prices above 10?

```
prices[prices > 10]

##   pt   es   it   fr   uk   fi   be
## 10.3 10.6 11.5 12.3 11.4 10.9 12.1
```

- Which countries have prices above the average?

```
prices[prices > mean(prices)]

##   it   fr   uk   fi   be
## 11.5 12.3 11.4 10.9 12.1
```

Go Back

Solutions to Exercises 6 and 7

```
prices

##   pt   es   it   fr   de   gr   uk   fi   be   au
## 10.3 10.6 11.5 12.3  9.9  9.3 11.4 10.9 12.1  9.1
```

- Which countries have prices between 10 and 11 euros?

```
prices[prices > 10 & prices < 11]

##   pt   es   fi
## 10.3 10.6 10.9
```

- How would you raise the prices by 10%?

```
prices <- prices*1.1
prices

##    pt    es    it    fr    de    gr    uk    fi    be    au
## 11.33 11.66 12.65 13.53 10.89 10.23 12.54 11.99 13.31 10.01
```

Go Back

Solutions to Exercise 8

```
prices

##    pt    es    it    fr    de    gr    uk    fi    be    au
## 11.33 11.66 12.65 13.53 10.89 10.23 12.54 11.99 13.31 10.01
```

- How would you decrease by 2.5%, the prices of the countries with price above the average?

```
prices[prices > mean(prices)] <- prices[prices > mean(prices)]*0.975
prices

##    pt    es    it    fr    de    gr    uk    fi    be    au
## 11.33 11.66 12.33 13.19 10.89 10.23 12.23 11.69 12.98 10.01
```

Go Back

Hands On 2

Go to the site `http://www.xe.com` and create a vector with the information you obtain there concerning the exchange rate between some currencies. You may use the ones appearing at the opening page.

1. Create a function with 2 arguments: the first is a value in Euros and the second the name of other currency. The function should return the corresponding value in the specified currency. Solution
2. What happens if we make a mistake when specifying the currency name? Try. Solution
3. Try to apply the function to a vector of values provided in the first argument. Solution

Solution to exercise 1

```
exchg <- c(usd=1.35402, gbp=0.82477, aud=1.54171,
           cad=1.48437,nzd=1.63934, jpy=141.155)
exchg

##      usd      gbp      aud      cad      nzd      jpy
##   1.3540   0.8248   1.5417   1.4844   1.6393 141.1550
```

- Create a function with 2 arguments: the first is a value in Euros and the second the name of other currency. The function should return the corresponding value in the specified currency.

```
conv <- function(eur,curr) eur*exchg[curr]  # depends on "exchg"
conv(234,"jpy")

##   jpy
## 33030
```

Solution to exercise 1 (cont.)

```
exchg <- c(usd=1.35402, gbp=0.82477, aud=1.54171,
           cad=1.48437,nzd=1.63934, jpy=141.155)
exchg

##      usd      gbp      aud      cad      nzd      jpy
##   1.3540   0.8248   1.5417   1.4844   1.6393 141.1550
```

- Create a function with 2 arguments: the first is a value in Euros and the second the name of other currency. The function should return the corresponding value in the specified currency.

```
conv2 <- function(eur,curr,camb) eur*camb[curr]
conv2(234,"jpy",exchg)

##   jpy
## 33030
```

Go Back

Solution to exercise 2

```
exchg

##      usd      gbp      aud      cad      nzd      jpy
##   1.3540   0.8248   1.5417   1.4844   1.6393 141.1550
```

- What happens if we make a mistake when specifying the currency name? Try.

```
conv(2356,"ukd")

## <NA>
##   NA
```

Go Back

Solution to exercise 3

```
exchg

##      usd      gbp      aud      cad      nzd      jpy
##   1.3540   0.8248   1.5417   1.4844   1.6393 141.1550
```

- Try to apply the function to a vector of values provided in the first argument.

```
conv(c(235,46576,675,453,234),"usd")

## [1]   318.2 63064.8   914.0   613.4   316.8
```

Go Back

Hands On Data Frames - Boston Housing

Load in the data set named "Boston" that comes with the package `MASS`. This data set describes the median house price in 506 different regions of Boston. You may load the data doing: `data(Boston,package='MASS')`. This should create a data frame named `Boston`. You may know more about this data set doing `help(Boston,package='MASS')`. With respect to this data answer the following questions:

1. What are the data on the regions with an median price higher than 45? solution
2. What are the values of `nox` and `tax` for the regions with an average number of rooms (`rm`) above 8? solution
3. Which regions have a median price between 10 and 15? solution
4. What is the average criminality rate (`crim`) for the regions with a number of rooms above 6? solution

Solution to Exercise 1

- What are the data on the regions with an median price higher than 45?

```
data(Boston,package="MASS")
subset(Boston,medv > 45)
```

```
##        crim zn indus chas    nox    rm   age   dis rad tax ptratio blac
## 162 1.46336  0 19.58    0 0.6050 7.489  90.8 1.971   5 403    14.7 374.
## 163 1.83377  0 19.58    1 0.6050 7.802  98.2 2.041   5 403    14.7 389.
## 164 1.51902  0 19.58    1 0.6050 8.375  93.9 2.162   5 403    14.7 388.
## 167 2.01019  0 19.58    0 0.6050 7.929  96.2 2.046   5 403    14.7 369.
## 187 0.05602  0  2.46    0 0.4880 7.831  53.6 3.199   3 193    17.8 392.
## 196 0.01381 80  0.46    0 0.4220 7.875  32.0 5.648   4 255    14.4 394.
## 204 0.03510 95  2.68    0 0.4161 7.853  33.2 5.118   4 224    14.7 392.
## 205 0.02009 95  2.68    0 0.4161 8.034  31.9 5.118   4 224    14.7 390.
## 226 0.52693  0  6.20    0 0.5040 8.725  83.0 2.894   8 307    17.4 382.
## 229 0.29819  0  6.20    0 0.5040 7.686  17.0 3.375   8 307    17.4 377.
## 234 0.33147  0  6.20    0 0.5070 8.247  70.4 3.652   8 307    17.4 378.
## 258 0.61154 20  3.97    0 0.6470 8.704  86.9 1.801   5 264    13.0 389.
## 263 0.52014 20  3.97    0 0.6470 8.398  91.5 2.288   5 264    13.0 386.
## 268 0.57834 20  3.97    0 0.5750 8.297  67.0 2.422   5 264    13.0 384.
## 281 0.03578 20  3.33    0 0.4429 7.820  64.5 4.695   5 216    14.9 387.
## 283 0.06129 20  3.33    1 0.4429 7.645  49.7 5.212   5 216    14.9 377.
```

Solution to Exercise 2

- What are the values of `nox` and `tax` for the regions with an average number of rooms (`rm`) above 8?

```
subset(Boston,rm > 8,c(nox,tax))
```

```
##         nox tax
## 98   0.4450 276
## 164  0.6050 403
## 205  0.4161 224
## 225  0.5040 307
## 226  0.5040 307
## 227  0.5040 307
## 233  0.5070 307
## 234  0.5070 307
## 254  0.4310 330
## 258  0.6470 264
## 263  0.6470 264
## 268  0.5750 264
## 365  0.7180 666
```

Solution to Exercise 3

- Which regions have a median price between 10 and 15?

```
subset(Boston, medv > 10 & medv < 15)
```

```
##         crim zn indus chas   nox    rm   age   dis rad tax ptratio  bla
## 21   1.25179  0  8.14    0 0.538 5.570  98.1 3.798   4 307    21.0 376.
## 24   0.98843  0  8.14    0 0.538 5.813 100.0 4.095   4 307    21.0 394.
## 26   0.84054  0  8.14    0 0.538 5.599  85.7 4.455   4 307    21.0 303.
## 28   0.95577  0  8.14    0 0.538 6.047  88.8 4.453   4 307    21.0 306.
## 31   1.13081  0  8.14    0 0.538 5.713  94.1 4.233   4 307    21.0 360.
## 32   1.35472  0  8.14    0 0.538 6.072 100.0 4.175   4 307    21.0 376.
## 33   1.38799  0  8.14    0 0.538 5.950  82.0 3.990   4 307    21.0 232.
## 34   1.15172  0  8.14    0 0.538 5.701  95.0 3.787   4 307    21.0 358.
## 35   1.61282  0  8.14    0 0.538 6.096  96.9 3.760   4 307    21.0 248.
## 49   0.25387  0  6.91    0 0.448 5.399  95.3 5.870   3 233    17.9 396.
## 130  0.88125  0 21.89    0 0.624 5.637  94.7 1.980   4 437    21.2 396.
## 139  0.24980  0 21.89    0 0.624 5.857  98.2 1.669   4 437    21.2 392.
## 141  0.29090  0 21.89    0 0.624 6.174  93.6 1.612   4 437    21.2 388.
## 142  1.62864  0 21.89    0 0.624 5.019 100.0 1.439   4 437    21.2 396.
## 143  3.32105  0 19.58    1 0.871 5.403 100.0 1.322   5 403    14.7 396.
## 145  2.77974  0 19.58    0 0.871 4.903  97.8 1.346   5 403    14.7 396.
## 146  2.37934  0 19.58    0 0.871 6.130 100.0 1.419   5 403    14.7 172.
## 148  2.36862  0 19.58    0 0.871 4.926  95.7 1.461   5 403    14.7 391.
```

Solution to Exercise 4

- What is the average criminality rate (`crim`) for the regions with a number of rooms above 6?

```
colMeans(subset(Boston, rm > 6, crim))

##  crim
## 2.535
```

Go Back

Hands On Data Manipulation - Houston Crime Data

- The site `https://www.houstontx.gov/police/cs/Monthly_Crime_Data_by_Street_and_Police_Beat.htm` contains a series of spreadsheets with information on crime events at the city of Houston, US
  1. Download the spreadsheet of March 2015 to your computer.
  2. Import the spreadsheet into **dplyr** data frame table and try to understand its structure and information (you may search for more information at the above site) solution
  3. What is the total number of offenses that were registered? solution
  4. Obtain these totals by type of offense solution
  5. Explore the function `weekdays` and add a new column with the name of the week day when each event happened solution
  6. What is the number of events per day of the week (present the days by decreasing number of events) solution

Hands On Data Manipulation - Houston Crime Data

7. What are the top 10 police beats with larger number of events? solution
8. Explore the function `cut` and add a new column that breaks the hour when each event happened into a series of periods (e.g. morning, afternoon, etc.) solution
9. Check the number of events per period of the day solution
10. What is the period of the day with more events per day of the week? solution
11. Check the data carefully and try to find and solve some of its problems solution

Solutions

- Import the spreadsheet into R.

```
library(readxl)
library(dplyr)
dat <- read_excel("mar15.xls")
```

```
dat[1:2,]

## Source: local data frame [2 x 10]
##
##         Date Hour Offense Type  Beat                    Premise BlockRange
## 1 2015-03-04   14        Theft 10H10         Bar or Night Club  3100-3199
## 2 2015-03-13   22      Robbery 10H10 Road, Street, or Sidewalk  1900-1999
## Variables not shown: StreetName (chr), Type (chr), Suffix (chr), #
##   offenses (dbl)

colnames(dat)[c(3,10)] <- c("Offense","NrOffs")
```

Go Back

Solutions

- What is the total number of offenses that were registered?

```
dat %>% summarise(totOff=sum(NrOffs))

## Source: local data frame [1 x 1]
##
##   totOff
## 1   9415
```

Go Back

Solutions

- Obtain these totals by type of offense

```
group_by(dat,Offense) %>% summarise(totOff=sum(NrOffs))
```

```
## Source: local data frame [8 x 2]
##
##              Offense totOff
## 1           1.000000      1
## 2 Aggravated Assault    918
## 3         Auto Theft    977
## 4           Burglary   1536
## 5             Murder     23
## 6               Rape     49
## 7            Robbery    749
## 8              Theft   5162
```

Go Back

Solutions

- Explore the function `weekdays` and add a new column with the name of the week day when each event happened

```
dat <- dat %>% mutate(WeekDay=weekdays(as.Date(Date)))
```

Go Back

Solutions

- What is the number of events per day of the week (present the days by decreasing number of events)

```
group_by(dat,WeekDay) %>%
    summarise(totEvents=sum(NrOffs)) %>%
        arrange(desc(totEvents))
```

```
## Source: local data frame [7 x 2]
##
##   WeekDay totEvents
## 1 Segunda      1556
## 2 Domingo      1444
## 3   Terça      1425
## 4  Sábado      1286
## 5   Sexta      1245
## 6  Quinta      1230
## 7  Quarta      1229
```

Go Back

Solutions

- What are the top 10 police beats with larger number of events?

```
group_by(dat,Beat) %>%
    summarise(totEvents=sum(NrOffs)) %>%
        arrange(desc(totEvents)) %>%
            slice(1:10)

## Source: local data frame [10 x 2]
##
##     Beat totEvents
## 1  15E40       258
## 2   1A20       229
## 3  13D20       207
## 4  19G10       204
## 5   6B60       203
## 6  12D10       191
## 7  18F20       173
## 8   1A10       159
## 9  18F50       152
## 10  1A50       150
```

Go Back

Solutions

- Explore the function `cut` and add a new column that breaks the hour when each event happened into a series of periods (e.g. morning, afternoon, etc.)

```
dat <- dat %>%
    mutate(Period=cut(as.numeric(dat$Hour),
                breaks=c(-1,6,11,17,23),
                labels=c("latenight","morning","afternoon","night")))
```

Go Back

Solutions

- Check the number of events per period of the day

```
group_by(dat,Period) %>%
    summarise(totEvents=sum(NrOffs)) %>%
        arrange(desc(totEvents))

## Source: local data frame [4 x 2]
##
##       Period totEvents
## 1  afternoon      3366
## 2      night      2448
## 3    morning      2374
## 4  latenight      1227
```

Go Back

Solutions

- What is the period of the day with more events per day of the week?

```
group_by(dat,WeekDay,Period) %>%
    summarise(totEvents=sum(NrOffs)) %>%
        arrange(desc(totEvents)) %>% slice(1)

## Source: local data frame [7 x 3]
## Groups: WeekDay
##
##   WeekDay    Period totEvents
## 1 Domingo afternoon       495
## 2  Quarta afternoon       439
## 3  Quinta afternoon       464
## 4 Segunda afternoon       581
## 5   Sexta afternoon       448
## 6  Sábado afternoon       437
## 7   Terça afternoon       502
```

Go Back

Solutions

- Check the data carefully and try to find and solve some of its problems

- The data contains strange dates (they should all be Mar 2015!)

```
library(lubridate)
dat%>% filter(year(Date) != 2015, month(Date) != 3)
```

```
## Source: local data frame [56 x 12]
##
##          Date Hour  Offense  Beat                                Premise
## 1  2014-12-16   09   Murder 10H20                    Residence or House
## 2  2013-01-01   19    Theft 10H20                    Residence or House
## 3  2013-11-07   12    Theft 10H30      Other, Unknown, or Not Listed
## 4  2014-01-01   09    Theft 10H40                                    NA
## 5  2014-11-28   14    Theft 10H40                                    NA
## 6  2014-02-19   09    Theft 10H40 Social Services or Public Charities
## 7  2014-02-17   11    Theft 10H70                           Hospital
## 8  2014-11-10   09 Burglary 10H70             Rental Storage Facility
## 9  2014-11-22   01  Robbery 10H70           Road, Street, or Sidewalk
## 10 2014-06-01   10    Theft 11H30   Pawn, Resale Shop, or Flea Market
## ..        ...  ...      ...   ...                                 ...
## Variables not shown: BlockRange (chr), StreetName (chr), Type (chr),
##   Suffix (chr), NrOffs (dbl), WeekDay (chr), Period (fctr)
```

```
dat <- dat %>% filter(year(Date) == 2015, month(Date) == 3)
```

Solutions

- Check the data carefully and try to find and solve some of its problems

- The data contains strange dates (they should all be Mar 2015!)

```
library(lubridate)
dat%>% filter(year(Date) != 2015, month(Date) != 3)

## Source: local data frame [56 x 12]
##
##          Date Hour  Offense  Beat                                Premise
## 1  2014-12-16   09   Murder 10H20                 Residence or House
## 2  2013-01-01   19    Theft 10H20                 Residence or House
## 3  2013-11-07   12    Theft 10H30       Other, Unknown, or Not Listed
## 4  2014-01-01   09    Theft 10H40                                   NA
## 5  2014-11-28   14    Theft 10H40                                   NA
## 6  2014-02-19   09    Theft 10H40 Social Services or Public Charities
## 7  2014-02-17   11    Theft 10H70                            Hospital
## 8  2014-11-10   09 Burglary 10H70             Rental Storage Facility
## 9  2014-11-22   01  Robbery 10H70           Road, Street, or Sidewalk
## 10 2014-06-01   10    Theft 11H30   Pawn, Resale Shop, or Flea Market
## ..        ...  ...      ...   ...                                 ...
## Variables not shown: BlockRange (chr), StreetName (chr), Type (chr),
##   Suffix (chr), NrOffs (dbl), WeekDay (chr), Period (fctr)

dat <- dat %>% filter(year(Date) == 2015, month(Date) == 3)
```

Solutions

- The data contains a strange offense type (1.000000)

```
dat <- filter(dat,Offense!="1.000000")
```

- The data contains some unknown premises

```
dat[which(is.na(dat$Premise)),"Premise"] <- "Unknown"
```

Solutions

- The data contains a strange offense type (1.000000)

```
dat <- filter(dat,Offense!="1.000000")
```

- The data contains some unknown premises

```
dat[which(is.na(dat$Premise)),"Premise"] <- "Unknown"
```

Solutions

- The data contains a strange offense type (1.000000)

```
dat <- filter(dat,Offense!="1.000000")
```

- The data contains some unknown premises

```
dat[which(is.na(dat$Premise)),"Premise"] <- "Unknown"
```

Solutions

- The data contains a strange offense type (1.000000)

```
dat <- filter(dat,Offense!="1.000000")
```

- The data contains some unknown premises

```
dat[which(is.na(dat$Premise)),"Premise"] <- "Unknown"
```

Hands On Time Series

Package **quantmod** (an extra package that you need to install) contains several facilities to handle financial time series. Among them, the function `getMetals` allows you to download the prices of metals from `oanda.com`. Explore the help page of the function to try to understand how it works, and the answer the following:

1. Obtain the prices of gold of the current year solution
2. Show the prices in January solution
3. Show the prices from January 10 till February 15 solution
4. Obtain the prices of silver in the last 30 days
   Tip: explore the function `days()` from package **lubridate** solution
5. Plot the prices of silver in the last 7 days
   Tip: explore the function `last()` on package **xts** solution

Solution to Exercise 1

- Obtain the prices of gold of the current year

```
library(quantmod)
getMetals("gold",from="2017-01-01",base.currency="EUR")

## [1] "XAUEUR"
```

Go Back

Solution to Exercise 2

- Show the prices in January

```
XAUEUR["2017-01"]

##            XAU.EUR
## 2017-01-01 1094.64
## 2017-01-02 1094.64
## 2017-01-03 1105.41
## 2017-01-04 1114.41
## 2017-01-05 1114.66
## 2017-01-06 1112.49
## 2017-01-07 1113.49
## 2017-01-08 1113.49
## 2017-01-09 1117.20
## 2017-01-10 1119.10
## 2017-01-11 1126.98
## 2017-01-12 1128.78
## 2017-01-13 1124.87
## 2017-01-14 1125.05
## 2017-01-15 1125.01
## 2017-01-16 1133.62
## 2017-01-17 1135.78
## 2017-01-18 1135.47
```

Solution to Exercise 3

- Show the prices from January 10 till February 15

```
XAUEUR["2017-01-10/2017-02-15"]

##            XAU.EUR
## 2017-01-10 1119.10
## 2017-01-11 1126.98
## 2017-01-12 1128.78
## 2017-01-13 1124.87
## 2017-01-14 1125.05
## 2017-01-15 1125.01
## 2017-01-16 1133.62
## 2017-01-17 1135.78
## 2017-01-18 1135.47
## 2017-01-19 1130.25
## 2017-01-20 1129.53
## 2017-01-21 1131.03
## 2017-01-22 1131.09
## 2017-01-23 1131.76
## 2017-01-24 1129.88
## 2017-01-25 1121.38
## 2017-01-26 1113.74
## 2017-01-27 1110.07
```

Solution to Exercise 4

- Obtain the prices of silver in the last 30 days

```
fstDate <- Sys.Date() - 30
getMetals("silver",from=fstDate,base.currency="EUR")

## [1] "XAGEUR"
```

or a more general setting

```
library(lubridate)
getMetals("silver",from=Sys.Date() - days(30), base.currency="EUR")
```

Go Back

Solution to Exercise 5

- Plot the prices of silver in the last 7 days

```
library(xts)
plot(last(XAGEUR,"7 days"), main = "Silver in the last 7 days")
```