Lab exercises: installing R/ Rstudio, beginning to work with data: filtering, distributions, populations, Data Frames…

Thilanka Munasinghe

Data Analytics

ITWS-4600/ITWS-6600/MATP-4450/CSCI-4960
MGMT4962/6962

Group 1, Lab 1, January 23rd , 2020

Lab tasks

- Installations
- Exploring data and their distributions
- Fitting distributions
- Data Frames in R

Objectives for today

- Get familiar with:
  - R (Data Frames)
  - Distributions
  - Populations
  - Fitting
  - Filtering

Creating Data Frames in R

Using the data.frames() in R, we can create the data frames
Read the documentation for data.frame in Rstudio,
Using the **help(data.frame)**

```
# Creating a dataframe
# Example: RPI Weather dataframe.

days <- c('Mon', 'Tue','Wed','Thur','Fri','Sat','Sun') #  days
temp <- c(28,30.5,32,31.2,29.3,27.9,26.4) # Temperature in F' during the winter :)
snowed <- c('T','T','F','F','T','T','F') # Snowed on that day: T = TRUE, F= FALSE
help("data.frame")
RPI_Weather_Week <- data.frame(days,temp,snowed) # creating the dataframe using the data.frame() function

RPI_Weather_Week
head(RPI_Weather_Week) # head of the data frame, NOTE: it will show only 6 rows, usually head() function shows the
# first 6 rows of the dataframe, here we have only 7 rows in our dataframe.

str(RPI_Weather_Week) # we can take a look at the structure of the dataframe using the str() function.

summary(RPI_Weather_Week) # summary of the dataframe using the summary() function
```

Data frames …

```
RPI_Weather_Week[1,] # showing the 1st row and all the columns
RPI_Weather_Week[,1] # showiing the 1st coulmn and all the rows

RPI_Weather_Week[,'snowed']
RPI_Weather_Week[,'days']
RPI_Weather_Week[,'temp']
RPI_Weather_Week[1:5,c("days","temp")]
RPI_Weather_Week$temp
subset(RPI_Weather_Week,subset=snowed==TRUE)

sorted.snowed <- order(RPI_Weather_Week['snowed'])
sorted.snowed
RPI_Weather_Week[sorted.snowed,]
```

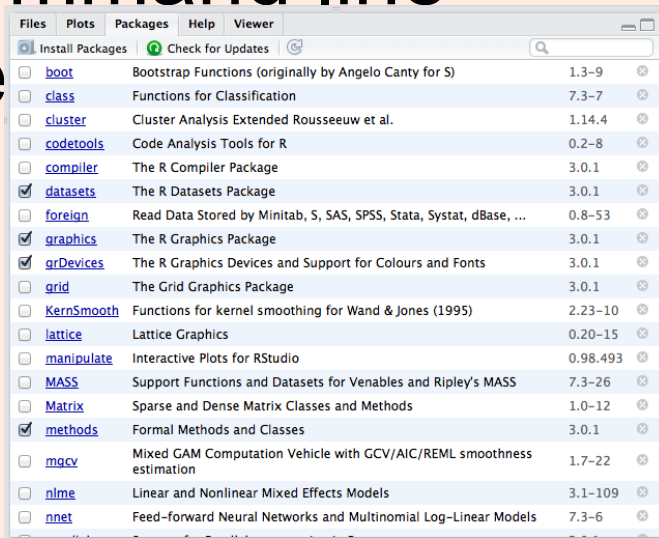
```r
# RPI_Weather_Week[descending_snowed,]
dec.snow <- order(-RPI_Weather_Week$temp)
dec.snow
# Creating Dataframes
# creating an empty dataframe
empty.DataFrame <- data.frame()
v1 <- 1:10
v1
letters
v2 <- letters[1:10]
df <- data.frame(col.name.1 = v1,col.name.2 = v2)
df
# importing data and exporting data
# writing to a CSV file:
write.csv(df,file = 'saved_df1.csv')
df2 <- read.csv('saved_df1.csv')
df2
```

Table: Matlab/R/scipy-numpy

- http://hyperpolyglot.org/numerical-analysis

Gnu R

- http://lib.stat.cmu.edu/R/CRAN/ - load this first
- http://cran.r-project.org/doc/manuals/
- http://cran.r-project.org/doc/manuals/R-lang.html
- R Studio – (see R-intro.html too) https://www.rstudio.com/products/rstudio/ (desktop version)
- Manuals - Libraries – at the command line – library(), or select the package check/ uncheck as needed



Reminder: files

- http://aquarius.tw.rpi.edu/html/DA/
- And some directories under this link
  - **please search before asking**
- This is where the files for assignments, lab exercises are
  - data and code fragments…

Exercises – getting data in

- Rstudio
  - read in csv file (two ways to do this) - GPW3_GRUMP_SummaryInformation_2010.csv
  - Read in excel file (directly or by csv convert) - 2010EPI_data.xls (EPI2010_all countries or EPI2010_onlyEPIcountries tabs)
  - Plot some variables
  - Commonalities among them?
- Also for other datasets, enter these in the R command window pane or cmd line

> data()

> help(data)

Files

- 2010EPI_data.xls – with missing values changed to suit your application (EPI2010_all countries or EPI2010_onlyEPIcountries tabs)
- See group1/lab1_data.R
- Get the data read in (in e.g. use “EPI_data” for the object (in R))

> EPI_data <- read.csv(”<path>/2010EPI_data.csv")

Note: replace default data frame name – cannot start with numbers! Munging has begun

Note: replace <path> with either a directory path or use setwd(“<path>”)

> View(EPI_data)

Tips (in R)

> attach(EPI_data) # sets the 'default' object

> fix(EPI_data) # launches a simple data editor – test it!

> EPI # prints out values EPI_data$EPI

```
 [1]   NA   NA 36.3   NA 71.4   NA   NA 40.7 61.0 60.4   NA 69.8 65.7 78.1 59.1 43.9 58.1
[18] 39.6 47.3 44.0 62.5 42.0   NA 55.9 65.4 69.9   NA 44.3 63.4   NA 60.8 68.0 41.3 33.3
[35] 66.4 89.1 73.3 49.0 54.3 44.6 51.6 54.0   NA 76.8   NA   NA 86.4 78.1   NA 56.3 71.6
[52] 73.2 60.5   NA 69.2 68.4 67.4 69.3 62.0 54.6   NA 70.6 63.8 43.1 74.7 65.9   NA 78.2
[69]   NA   NA 56.4 74.2 63.6 51.3   NA 44.4   NA 50.3 44.7 41.9 60.9   NA   NA 54.0   NA
[86]   NA 59.2   NA 49.9 68.7 39.5 69.1 44.6   NA 48.3 67.1 60.0 41.0 93.5 62.4 73.1 58.0
[103] 56.1 72.5 57.3 51.4 59.7 41.7   NA   NA 57.0 51.1 59.6 57.9   NA 50.1   NA   NA 63.7
[120]   NA 68.3 67.8 72.5   NA 65.6   NA 58.8 49.2 65.9 67.3   NA 60.6 39.4 76.3 51.3 42.8
[137]   NA 51.2 33.7   NA   NA 80.6 51.4 65.0   NA 59.3   NA 37.6   NA 40.2 57.1   NA 66.4
[154] 81.1 68.2   NA 73.4 45.9 48.0 71.4   NA 69.3 65.7   NA 44.3 63.1   NA 41.8 73.0 63.5
[171]   NA   NA 48.9   NA 67.0 61.2 44.6 55.3 69.4 47.1 42.3 69.6   NA   NA 51.1 32.1 69.1
[188]   NA   NA   NA 57.3 68.2 74.5 65.0 86.0 54.4   NA 64.6   NA 40.8 36.4 62.2 51.3   NA
[205] 38.4   NA   NA 54.2 60.6 60.4   NA   NA 47.9 49.8 58.2 59.1 63.5 42.3   NA   NA 62.9
[222]   NA   NA 59.0   NA   NA   NA 48.3 50.8 47.0 47.8
```

> tf <- is.na(EPI) # records True values if the value is NA

> E <- EPI[!tf] # filters out NA values, new array

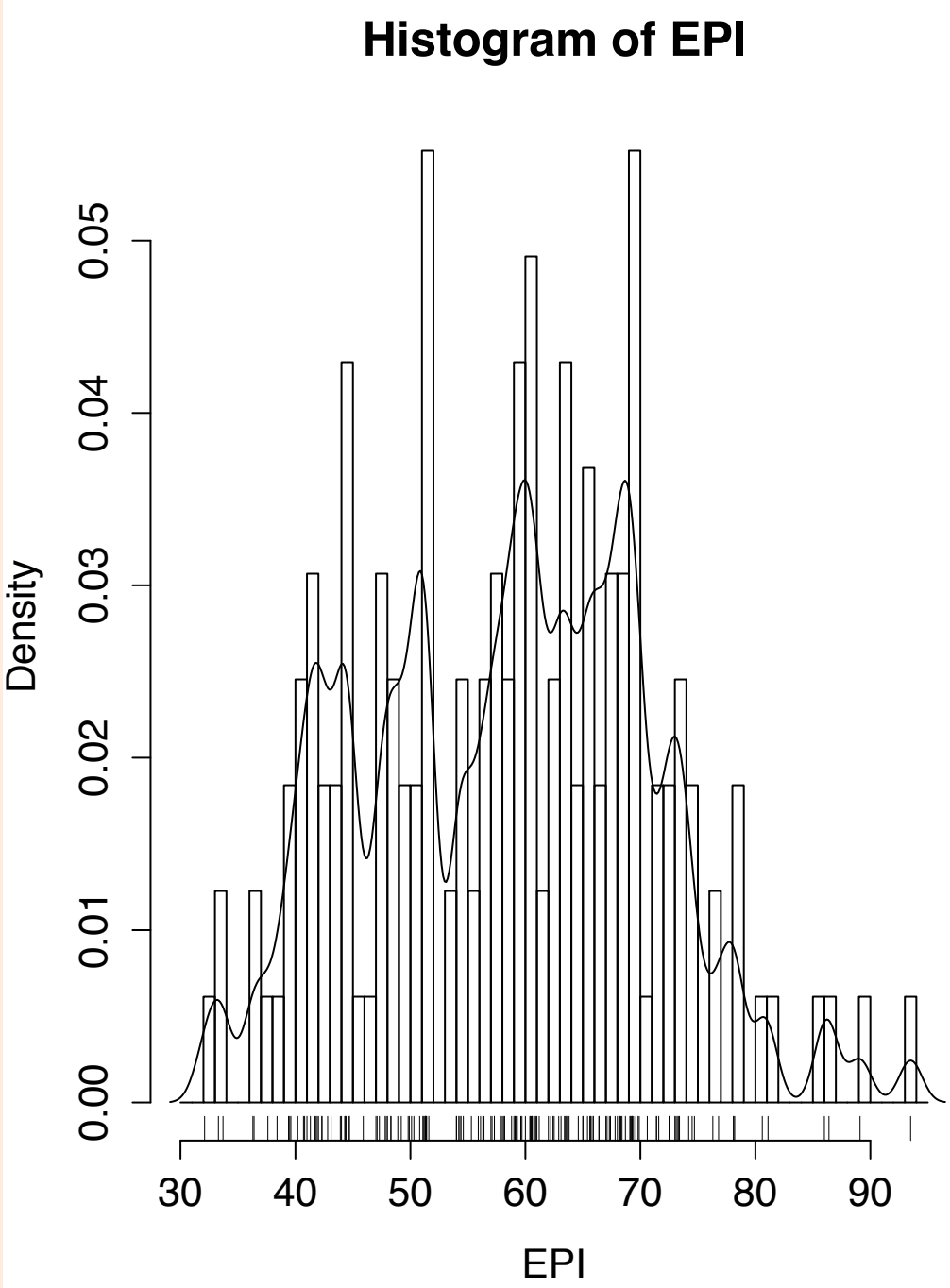
Exercise 1: exploring the distribution

```
> summary(EPI) # stats
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
  32.10   48.60   59.20   58.37   67.60   93.50      68
> fivenum(EPI,na.rm=TRUE)
[1] 32.1 48.6 59.2 67.6 93.5
> stem(EPI)                    # stem and leaf plot
> hist(EPI)
> hist(EPI, seq(30., 95., 1.0), prob=TRUE)
> lines(density(EPI,na.rm=TRUE,bw=1.)) # or try
bw="SJ"
> rug(EPI)
#Use help(<command>), e.g. > help(stem)
```

- See group1/lab1_summary.R

Save your plots, name them.

Save the commands you used to generate them.



Exercise 1: fitting a distribution beyond histograms

- Cumulative density function?

  > plot(ecdf(EPI), do.points=FALSE, verticals=TRUE)

- Quantile-Quantile?

  > par(pty="s")

  > qqnorm(EPI); qqline(EPI)

- Make a Q-Q plot against the generating distribution by: x<-seq(30,95,1)

  > qqplot(qt(ppoints(250), df = 5), x, xlab = "Q-Q plot for t dsn")

  > qqline(x)

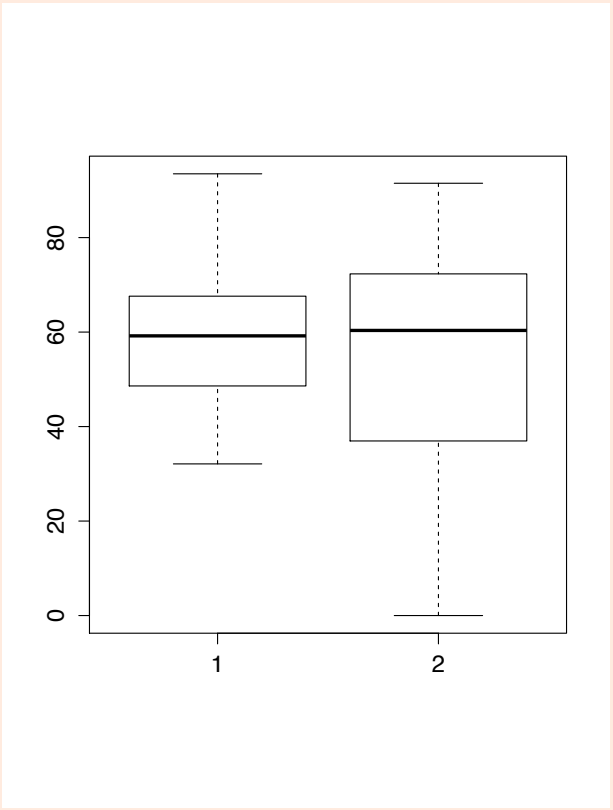
Exercise 1: fitting a distribution

- Your exercise: do the same exploration and fitting for another 2 variables in the EPI_data, i.e. primary variables (DALY, WATER_H, …)

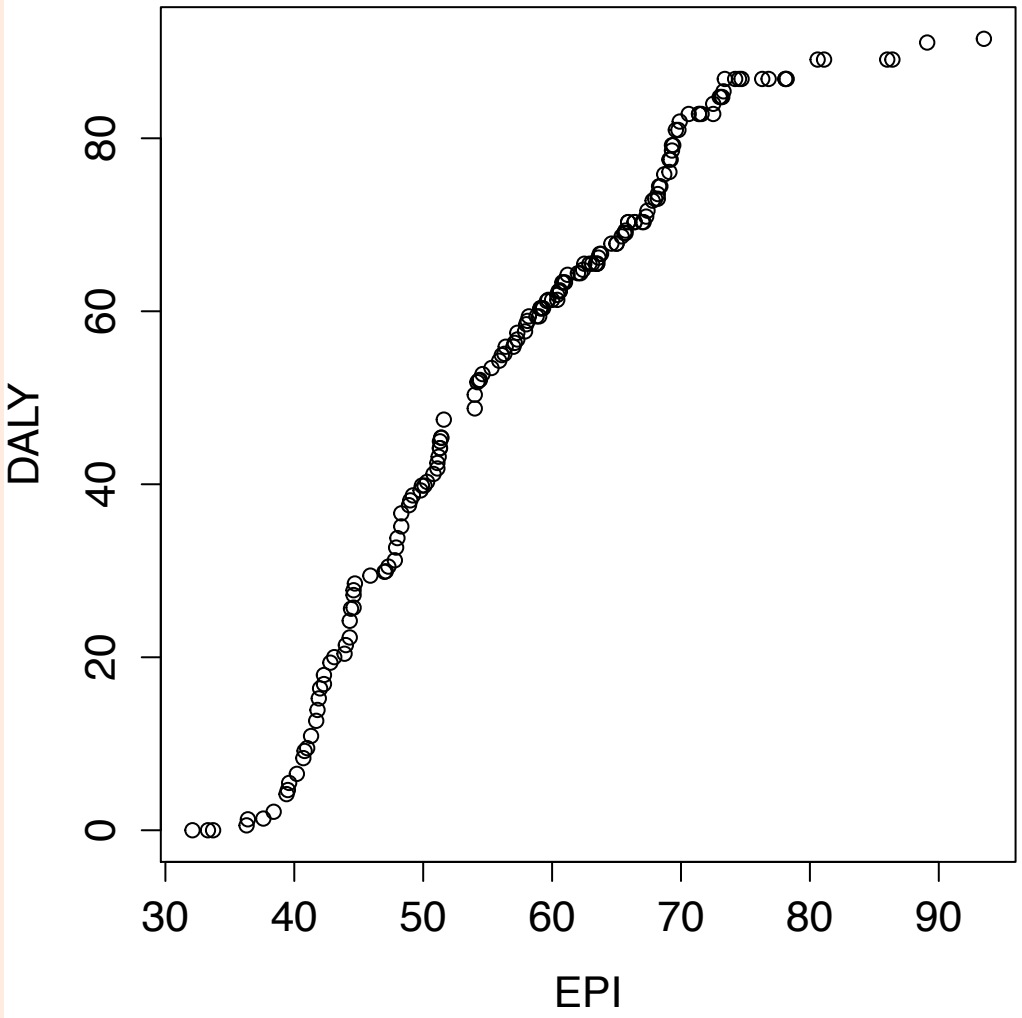
- Try fitting other distributions – i.e. as ecdf or qq-

Comparing distributions

```
> boxplot(EPI,DALY)
```



qqplot(EPI,DALY)



But there is more

- Your exercise – intercompare: EPI, ENVHEALTH, ECOSYSTEM, DALY, AIR_H, WATER_H, AIR_EWATER_E, BIODIVERSITY ** (subject to possible filtering…)

- Note 2010 and 2016 datasets….

Distributions

- See archive “alldist.zip” or individual files in /distribution
  - Excel files for different distributions…
  - Expanded in the folder: e.g. lognorm.xls

- In R:

> **help(distributions)**

Exercise 2: filtering (populations)

- Conditional filtering:
  > EPILand<-EPI[!Landlock]
  > Eland <- EPILand[!is.na(EPILand)]
  > hist(ELand)
  > hist(ELand, seq(30., 95., 1.0), prob=TRUE)
- Repeat exercise 1…
- Also look at: No_surface_water, Desert and High_Population_Density
- Your exercise: how to filter on EPI_regions or GEO_subregion?
- E.g. EPI_South_Asia <- EPI[<what is this>]

GPW3_GRUMP – repeat…

- The reading in, then exploration/summary, plots, histograms, distributions, filtering, tests….

water_treatment

- water-treatment.csv

Objectives for this lab are:

- Get familiar with:
  - Distributions
  - Populations
  - Fitting
  - Creating Data Frames in R.