**Exercitiul 1 a)**

* **instructiunea if:**

if (max(magazin\_1) > 10){

print("numarul maxim de vanzari intr-o zi in magazinul 1 este mai mare de 10")

} else {

print("numarul maxim de vanzari intr-o zi in magazinul 1 este mai mic de 10")}

**verifica daca numarul maxim de vanzari intr-o zi in magazinul 1 este mai mare decat 10**

if(numar %% 2 == 0){

cat("numarul ", numar , " este par\n")

} else {

cat("numarul ", numar, " este impar\n")}

**verifica daca un numar este par sau impar**

* **instructiunea for:**

n for (i in 1:100){

k<-0

for(j in 1:i){

if(i%%j==0)k<-k+1;

}

if(k == 2)print(j)

}

**afisare numere prime mai mici decat 100**

for(i in 1:nrow(medie\_vanzari)) {

cat(medie\_vanzari[i, 1], "a avut un numar mediu zilnic de vanzari de: ", medie\_vanzari[ i, 2], "\n");

}

**afisez un mesaj cu magazinul si media zilnica de vanzari**

* **instructiunea while:**

i <- 1;

while(i <= nrow(medie\_vanzari)) {

cat(medie\_vanzari[i, 1], "a avut un numar mediu zilnic de vanzari de: ", medie\_vanzari[ i, 2], "\n");

i <- i+1

}

**afisez un mesaj cu magazinul si media zilnica de vanzari, de data aceasta insa cu while**

number <- 100

sum <- 0

while(number >= 0 ) {

sum = sum + number

number = number - 2

}

sum

**suma numerelor pare mai mici decat 100**

**Exercitiul 1 b)**

* **functia medie:**

s<- 0

media <- function(x) {

n <- length(x)

for(i in 1:n) s <- s + x[i]

medie<-s / n

return(medie);

}

* **functia abatere standard:**

abatere\_standard <- function(x){

n <- length(x);

suma = 0;

for( i in 1:n) suma = suma + (x[i] - media(x))^2;

rezultat = sqrt(suma/(n-1));

return(rezultat);

}

[**link formula abatere standard**](https://www.investopedia.com/terms/s/standarddeviation.asp)

* **functia test student:**

student\_test <-function(x,prob){

se <- sd(x) / sqrt(length(x))

alpha <- 1 - prob #qnorm ii normal distribution

limite<-c(mean(x) - se \* qnorm(1-alpha / 2), mean(x) + se \* qnorm(1-alpha / 2))

return(limite)

}

* **functia coeficient corelatie:**

corelatie <- function(x,y){

suma = 0;

n = length(x);

for(i in 1:n){

suma = suma + (x[i]-mean(x))\*(y[i]-mean(y))

}

rezultat = suma/(sd(x)\*sd(y));

return(rezultat/4)

}

[**link formula coeficient corelatie**](http://www.umfcv.ro/files/b/i/Biostatistica%20MG%20-%20Cursul%205%20-%20Corelatii.pdf)slide 16/28

**Exercitiul 1 c)**

x <- c(12,7,34,9,14,22,17,16,42,15,11,22,24,7,44,19,2,76,62,18,13,15,40,23,80,60,45,12);

**x este un vector ce contine vanzarea medie a unui produs in fiecare zi pentru 4 magazine**

luni = x[seq(1, length(x), 7)]

marti = x[seq(2, length(x), 7)]

miercuri = x[seq(3, length(x), 7)]

joi = x[seq(4, length(x), 7)]

vineri = x[seq(5, length(x), 7)]

sambata = x[seq(6, length(x), 7)]

duminica = x[seq(7, length(x), 7)]

**am creat vectori pentru fiecare zi parcurgand din 7 in 7 pozitii vectorul initial x**

media\_pe\_zi <- data.frame (

ziua = c("luni", "marti", "miercuri", "joi", "vineri", "sambata", "duminica"),

media = c(mean(luni),mean(marti),mean(miercuri),mean(joi), mean(vineri), mean(sambata), mean(duminica))

)

**am creat un data frame, pentru media in functie de zi am folosit functia mean, la fel de bine puteam folosi deja functia media, definita intr-un exercitiu anterior, acolo este folosita structura de control for**

media\_pe\_zi\_reversed <- t(media\_pe\_zi);

**trebuie sa transpun data frame-ul**

solutie cu apply:

matricea\_x <- matrix(x, 7);

**pe prima linie vanzarile de luni, a doua linie vanzarile de marti si asa mai departe, coloanele reprezinta magazine asadar coloana 1 reprezinta primul mazagin**

data\_frame\_x <- as.data.frame(matricea\_x);

row.names(data\_frame\_x) <- c("luni", "marti", "miercuri", "joi", "vineri", "sambata", "duminica");

colnames(data\_frame\_x) <- c("magazin\_1", "magazin\_2", "magazin\_3", "magazin\_4")

data\_frame\_x = rev(data\_frame\_x);

data\_frame\_x

apply(data\_frame\_x, MARGIN=1, FUN=mean)

**MARGIN = 1 indica fptul ca ne referim la linii care reprezinta zilele**

**iar la FUN specificam ce functie vrem sa folosim, in cazul nostru mean**

**Exercitiul 1 d)**

|  |  |  |  |
| --- | --- | --- | --- |
| functia | input | output | exemple |
| apply() | matrice sau array | vector | calcul medie pe coloane, suma pe linii |
| lapply() | vector, lista, data frame | lista | aplica aceasi functie pentru fiecare element din lista, pot ridica fiecare element din lista la patrat, pot calcula radical din el sau orice alta functie |
| sapply() | vector, matrice, lista | array, matrice | aproximativ acelasi lucru ca lapply() |
| tapply() | vector | array | ne ajuta sa impartim in submultimi si aplicam anumite functii pe ele ( de exemplu daca am un date frame cu produse(in care am pret si categorie) lopot afisa pretul mediu pe categorii de produse) |
| mapply() | vector, matrix | vector, matrix, lista | este o versiune multivariata a lui sapply() |

Exercitiul 1 e)

**set de date: Level of internet access – households**

**link:** <https://ec.europa.eu/eurostat/databrowser/view/TIN00134/default/table>

importare date

format\_wide<-read\_xlsx('C:\\Users\\liber\\OneDrive\\Desktop\\proiecte r studio\\proiect de semestru\\proiect semestru\\tin00134\_spreadsheet.xlsx')

View(format\_wide)

de mentionat ca fisierul excel a fost formatat de catre mine (un mic cleaning)

transformare format long

format\_long <- format\_wide %>%

gather(key = denumire,

value = valori, -TIME, convert = TRUE)

colnames(format\_long)

colnames(format\_long) <- c("tara", "an", "procent") #redenumiri

format\_long[format\_long == ":"] <- NA #am inlocuit : din celule cu NA

media pe un anumit an

format\_long$procent<-as.integer(format\_long$procent,na.omit = TRUE)

media\_pe\_an <- format\_long %>%

filter(an == 2020) %>%

summarize(

media\_pe\_an = mean(procent, na.rm= TRUE)

)

cat("Media pe anul 2020 este: ", sum(media\_pe\_an))

media pe fiecare an

media\_pe\_ani <- format\_long %>%

group\_by(an)%>%

summarise(mean(procent, na.rm = TRUE))

colnames(media\_pe\_ani) <- c("an", "procent")

media\_pe\_ani

Exercitiul 1 f)

sursa: [https://en.wikipedia.org/wiki/List\_of\_countries\_by\_average\_wage#cite\_note-OECDaaw-3](https://en.wikipedia.org/wiki/List_of_countries_by_average_wage%23cite_note-OECDaaw-3)

importare tabel html

pagina<-read\_html("https://en.wikipedia.org/wiki/List\_of\_countries\_by\_average\_wage#cite\_note-OECDaaw-3")

class(pagina)

library(magrittr)

tabele<-pagina %>% html\_nodes("table")

length(tabele)

hpi<-html\_table(tabele[[1]])

hpi[1] <- lapply(hpi[1], gsub, pattern = "\*", replacement = "", fixed = TRUE)

salarii\_long <- hpi %>%

gather(key = an,

value = salariu, -Country, convert = TRUE)

View(salarii\_long)

salariul mediu in fiecare an

salarii\_long$salariu<-as.integer(salarii\_long$salariu,na.omit = TRUE)

salarii\_pe\_ani <- salarii\_long %>%

group\_by(an)%>%

summarise(mean(salariu, na.rm = TRUE))

colnames(salarii\_pe\_ani) <- c("an", "salariu")

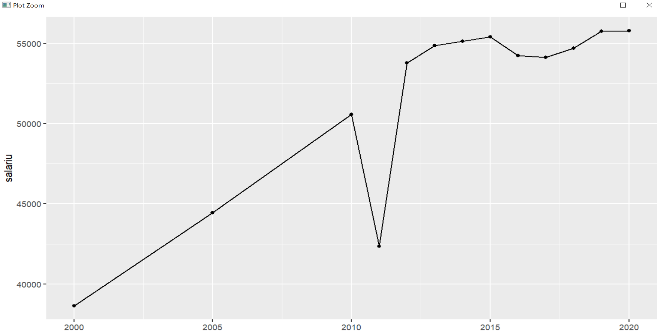
salarii\_pe\_ani

grafic evolutie salariu anual in Norvegia

salarii\_Norvegia <- salarii\_long

salarii\_Norvegia$Country<- as.character(salarii\_long$Country)

salarii\_Norvegia <- salarii\_Norvegia %>% filter(str\_detect(Country, "^Norway"))

salarii\_Norvegia %>%

ggplot(aes(x = an, y = salariu)) +

geom\_line() +

geom\_point()

labs(

y = "salariu",

x = "an")

**Exercitiul 2**

set de date: top 100 songs Spotify(2010-2019)

<https://www.kaggle.com/datasets/muhmores/spotify-top-100-songs-of-20152019?resource=download>

obiective:

sa obseram trenduri si patternuri intre melodii (cu ajutorul tabelelor de frecventa si al graficelor)

variabile selectate:

top genre - categoriala

artist\_type – categoriala

beats\_per\_minute - cantitativa continua numerica

duration - cantitativa continua numerica

dance - cantitativa continua numerica

energy - cantitativa continua numerica

acoustic - cantitativa continua numerica

top year - cantitativa continua numerica

live - cantitativa continua numerica

cleaning

songs\_wide<-readr::read\_csv('C:\\Users\\liber\\OneDrive\\Desktop\\proiecte r studio\\proiect de semestru\\proiect semestru\\Spotify 2010 - 2019 Top 100.csv')

songs\_wide <- na.omit(songs\_wide)

songs\_wide$added <- as.Date(songs\_wide$added)

songs\_wide$`top genre` <- as.factor(songs\_wide$`top genre`) #le facem tip categorii

songs\_wide$`artist type` <- as.factor(songs\_wide$`artist type`)

#cateva renameuri

songs\_wide <- songs\_wide %>%

rename(beats\_per\_minute = "bpm") %>%

rename(energy = "nrgy") %>%

rename(dance = "dnce") %>%

rename(duration = "dur") %>%

rename(acoustic = "acous") %>%

rename(speech = "spch") %>%

rename(top\_genre = "top genre") %>%

rename(top\_year = "top year") %>%

rename(artist\_type = "artist type") %>%

rename(year\_released = "year released")

songs\_wide <- select(songs\_wide, -val) #am sters coloana val pentru ca nu stiu ce semnifica

View(songs\_wide)

transformare format long

#transformare in format long

songs\_long <- songs\_wide %>%

gather(key = variabila, value = valoare, -title, convert = TRUE)

valoarea medie beats\_per\_minute a tuturor pieselor

beats\_per\_minute\_mediu <- songs\_long %>%

filter(variabila == "beats\_per\_minute") %>%

summarise(

beats\_per\_minute\_mediu <- mean(as.numeric(valoare), na.rm=TRUE)

)

cat("valoarea medie beats\_per\_minute a tuturor pieselor este: ", sum(beats\_per\_minute\_mediu))

numarul de piese per artist\_type

numar\_songs\_per\_artist\_type <- songs\_wide %>%

group\_by(artist\_type) %>%

summarise(

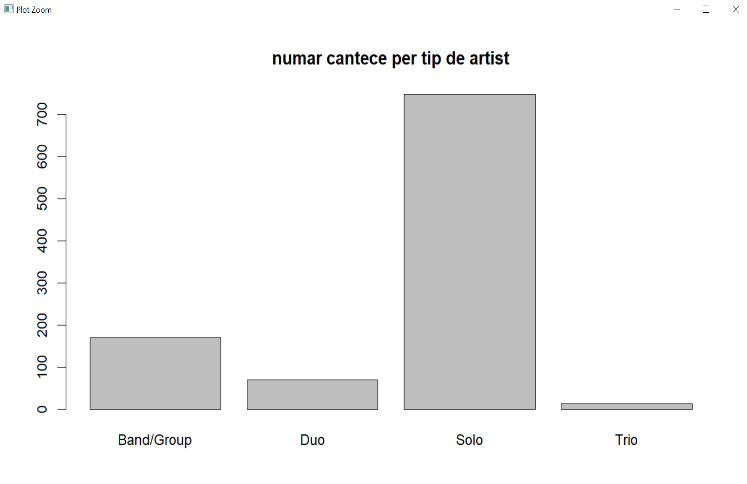
numar <- n()

)

colnames(numar\_songs\_per\_artist\_type) <- c("tip artist", "numar cantece")

numar\_songs\_per\_artist\_type <- as.data.frame(numar\_songs\_per\_artist\_type)

numar\_songs\_per\_artist\_type

barplot(numar\_songs\_per\_artist\_type$`numar cantece`, names.arg=numar\_songs\_per\_artist\_type$`tip artist`, main = "numar cantece per tip de artist")

durata medie a unei piese in functie de artist\_type

durata\_songs\_per\_artist\_type <- songs\_wide %>%

group\_by(artist\_type) %>%

summarise(

durata <- mean(duration)

)

colnames(durata\_songs\_per\_artist\_type) <- c("tip artist", "durata medie")

durata\_songs\_per\_artist\_type <- as.data.frame(durata\_songs\_per\_artist\_type)

durata\_songs\_per\_artist\_type

barplot(durata\_songs\_per\_artist\_type$`durata medie`, names.arg=durata\_songs\_per\_artist\_type$`tip artist`, main = "numar cantece per tip de artist")

observam ca nu exista diferente majore in durata cantecelor in fuctie de artist\_type, cea ce ne duce cu gandul ca nu exista o corelatie intre ele

evolutie numar piese cantate de Band/Group anual + grafic

band\_songs\_per\_year <- songs\_wide %>%

filter(artist\_type == "Band/Group") %>%

group\_by(top\_year) %>%

summarise(numar <- n())

colnames(band\_songs\_per\_year) <- c("an", "nr piese")

band\_songs\_per\_year <- as.data.frame(band\_songs\_per\_year)

band\_songs\_per\_year %>%

ggplot(aes(x = an, y = `nr piese`)) +

geom\_line() +

geom\_point()

labs(

y = "nr piese",

x = "an")

observam o tendinta clara de scadere a numarului de cantece cantate de bands/group

grafice corelatii intre variabile

ggplot(songs\_wide,aes(x = duration,y = beats\_per\_minute)) +

geom\_point() +

geom\_smooth(method='lm')

intre beats\_per minute si duration, nu pare sa existe corelatie

ggplot(songs\_wide,aes(x = live,y = acoustic)) +

geom\_point() +

geom\_smooth(method='lm')

pare sa existe o corelatie mai mare intre variabilele live si acoustic

ggplot(songs\_wide,aes(x = dance,y = speech)) +

geom\_point() +

geom\_smooth(method='lm')

**linkuri folosite:**

* <https://www.guru99.com/r-apply-sapply-tapply.html>
* suportul de curs/laborator
* <https://www.datacamp.com/community/tutorials/r-tutorial-apply-family?utm_source=adwords_ppc&utm_medium=cpc&utm_campaignid=12492439802&utm_adgroupid=122563404161&utm_device=c&utm_keyword=sapply%20r&utm_matchtype=b&utm_network=g&utm_adpostion=&utm_creative=504158805007&utm_targetid=kwd-302622694743&utm_loc_interest_ms=&utm_loc_physical_ms=1011806&gclid=Cj0KCQjw5-WRBhCKARIsAAId9FnIZ2eYwkUFIAlT-bl6sLVzlNRUkFmdYtiiUMSmUkoUhuC-WNaN_oEaAie3EALw_wcB>
* help-ul din R Studio
* <https://r-coder.com/tapply-r/>
* <https://www.statology.org/ggplot2-linear-regression/>
* <https://tidyr.tidyverse.org/>