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

- **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 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

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)
```

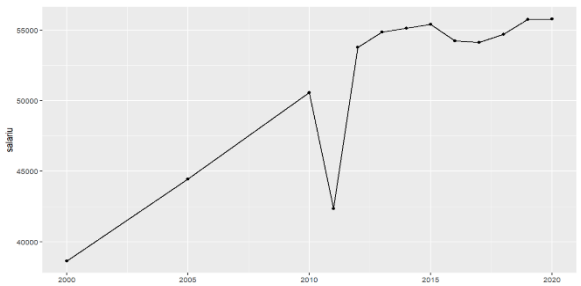
salariul mediu in fiecare an

```
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 observam 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)
```

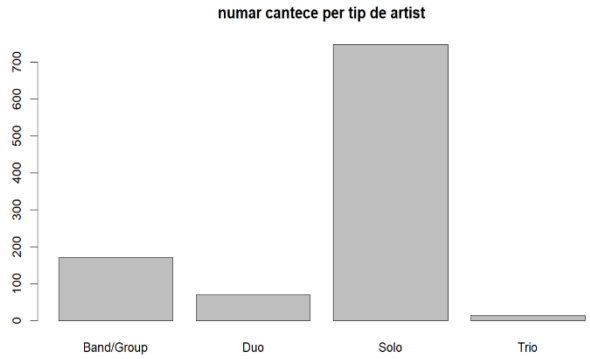
valoarea medie beats_per_minute a tuturor pieselor

```
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/