

**PROIECT**  
**ANALIZA STATISTICĂ ÎN R**

BEREGOI ALEXANDRA

BONAȚ PAULA-MIHAELA

## Problema 1

- a) Scrieți R Script-uri în care se utilizează structurile de control *if*, *for*, *while*, câte două pentru fiecare structură, și explicați care este scopul fiecăreia.

### IF

```
a <- 5
if(a==9){
  print("Yes")
}else {
  print("No")
}
```

Variabila *a* primește valoarea 5. Se verifică condiția: *a este egal cu valoarea 9*. Dacă este adevărată condiția, se afișează *Yes*, altfel se afișează *No*. În situația noastră, condiția este falsă și se afișează *No*.

```
b <- 10
if(b %% 2==0) {
  print("Valoarea este divizibila cu 2")
} else {
  print("Valoarea nu este divizibila cu 2")
}
```

Variabila *b* primește valoarea 10. Se verifică condiția: *restul împărțirii la 2 este 0*. Dacă condiția este adevărată, se afișează primul mesaj, altfel se afișează al doilea mesaj. În situația noastră restul împărțirii este 0 și se afișează primul mesaj *Valoarea este divizibila cu 2*.

### FOR

```
c <- c(5:10)
for(i in 1:6){print(c[i])}
```

Variabila *c* primește valorile de la 5 la 10. Pentru contorul *i* ce parcurge valorile variabilei, să se afișeze la fiecare repetare valoarea de pe poziția *i*.

```
d <- c(1:10)
for(i in 1:10){print(d[i]-1)}
```

Variabila  $d$  primește valorile de la 1 la 10. Pentru contorul  $i$  ce parcurge valorile variabilei, să se afișeze la fiecare repetare valoarea de pe poziția  $i$  din care se scade 1. Exemplu,  $d(1)-1=0$ , prima valoare va fi 0.

## WHILE

```
e <- 12
while(e>5) {
  e <- e-1
  print(e)
}
```

Variabila  $e$  ia valoarea 12. Cât timp variabila  $e$  mai mare decât 5, să se execute următoarele: variabila  $e$  primește valoarea  $e-1$  și se afișează această valoare. Inițial  $e=12>5$ , prin urmare,  $e$  va primi valoarea  $12-1=11$ . Analog, pentru valorile 11, 10, 9, 8, 6. Instrucțiunea se încheie când  $e$  ia valoarea 5.

```
f<-10
while (f <= 100) {print(f); f <- f + 20}
```

Variabila  $f$  ia valoarea 10. Cât timp aceasta este mai mică sau egală cu 100, să se execute următoarele: să se afișeze  $f$  și  $f$  primește valoarea sa la care se adaugă 20. Astfel se vor afișa: 10,  $10+20=30$ ,  $30+20=50$ ,  $50+20=70$ ,  $70+20=90$  și algoritmul se încheie, deoarece  $90+20=110 > 100$ .

b) Creați câte o funcție pentru a calcula:

## Media

```
s<-0
medie<-function(x) {
  n<- length(x)
  for(i in 1:n) s<-s + x[i]
  m<-s / n
  r<- paste("Media este:", m)
  print(r)
}
x<-c(3, 4.5, 7,8)
medie(x)
```

Se creează o funcție cu numele *medie*. Aceasta conține expresiile:  $n<-length(x)$  care calculează lungimea vectorului  $x$ , adică numărul de variabile (necesar în formula de calcul a

mediei); cu ajutorul structurii de control *for*, se calculează suma valorilor din vectorul *x*. Variabila *s* suma se inițializează cu 0, pentru a o putea folosi în funcție. Următoarea expresie  $m \leftarrow s/n$  calculează raportul dintre suma și lungimea vectorului, obținându-se media. Pentru afișarea rezultatului se creează variabila *r* cu ajutorul funcției *paste* care concatenează 2 componente de tip diferit: "Media este"- caracter și media obținută- numeric. Pentru a apela funcția, *x* primește valorile 3, 4.5, 7, 8 și se calculează media lor. Rezultatul afișat este: "Media este: 5.625".

Folosind funcția *mean*, se obține același rezultat:

```
x<-c(3, 4.5, 7, 8)
mean(x)
```

### Abaterea standard

```
s<-0
abatere<- function(x) {
  n<-length(x)
  m<-(x - mean(x)) ^ 2
  for(i in 1:n) s<-s+m[i]
  a<-sqrt(s / (n-1))
  r<- paste("Abaterea standard este:", a)
  print(r)
}
x<-c(3, 4.5, 7, 8)
abatere(x)
```

Se creează o funcție cu numele *abatere*. Aceasta conține expresiile:  $n \leftarrow \text{length}(x)$  care calculează lungimea vectorului *x*; expresia  $m \leftarrow (x - \text{mean}(x))^2$  calculează pătratul diferenței dintre *x* și media lui *x*. Cu ajutorul structurii de control *for*, se calculează suma acestor diferențe la pătrat. Expresia  $a \leftarrow \sqrt{s / (n-1)}$  calculează radical din varianță, care reprezintă suma pătratelor diferenței împărțită la lungimea vectorului-1. Pentru afișarea rezultatului se creează variabila *r* cu ajutorul funcției *paste* care concatenează 2 componente de tip diferit: "Abaterea standard este"- caracter și abaterea obținută- numeric. Pentru a apela funcția, *x* primește valorile 3, 4, 4.5, 7, 8 și se calculează abaterea standard. Rezultatul afișat este: "Abaterea standard este: 2.28673712233537".

Pentru a nu utiliza nici o funcție implementată în R, spre exemplu funcția *mean*, se poate aplica următoare metoda: se creează o funcție în cadrul căreia se apelează o altă funcție.

```
abatere<- function(x) {  
  n<-length(x)  
  medie<-function(x) {  
    for(i in 1:n) s<-s + x[i]  
    m<-s / n  
  }  
  m<-(x - medie(x)) ^ 2  
  for(i in 1:n) s<-s+m[i]  
  a<-sqrt(s / (n-1))  
  r<- paste("Abaterea este:", a)  
  print(r)  
}  
x<-c(3, 4.5, 7, 8)  
abatere(x)
```

O variantă mai simplă ar fi calcularea abaterii standard folosind funcțiile predefinite: *mean* și *sum*.

```
abatere<- function(x) {  
  n<-length(x)  
  a<-sqrt(sum((x - mean(x)) ^ 2) / (n - 1) )  
  r<- paste("Abaterea este:", a)  
  print(r)  
}  
x<-c(3, 4.5, 7, 8)  
abatere(x)
```

Folosind funcția *sd*, se obține același rezultat:

```
x<-c(3, 4.5, 7, 8)  
sd(x, na.rm=FALSE)
```

### Testul Student ( $H_0: \bar{X}=c$ )

```
testul_student<-function(x, c=165, prob=0.95) {  
  H0<-paste("Media din populatie este egala cu", c)  
  H1<-paste("Media din populatie nu este egala cu", c)  
  n<-length(x)  
  t_calculat<-(mean(x)-c)/(sd(x, na.rm=FALSE)/sqrt(n))  
}
```

```

alpha<-1-prob
t_tab<-qt(alpha/2, df=299)
if (abs(t_calculat)< abs(t_tab)) {
  print(H0)
}else{
  print(H1)
}
rez<-paste("t calculat=", t_calculat)
rez2<-paste("t tabelar= ", t_tab)
print(rez)
print(rez2)
}
x<-(1:300)
testul_student(x)

```

Se creează o funcție cu numele *testul\_student*. Funcția are ca parametru formal x, variabila locală c=165 (valoarea cu care se compară media) și prob=0.95. Se creează vectorii H0 și H1 (pentru a formula ipoteza nulă și alternativă) cu ajutorul funcției paste pentru a concatena componente de tip diferit: text și valoarea numerică a constantei. Pentru a calcula valoarea testului Student, se calculează valoarea statisticii calculate t\_calculat, alpha și valoarea statisticii teoretice t\_tab. Folosind structura de control if, se compară valoarea în modul a statisticii calculate cu cea teoretică. Dacă statistica calculată este mai mică decât statistica teoretică, se afișează H0 (se acceptă ipoteza nulă), altfel, se afișează H1("Media din populație nu este egală cu", c). De asemenea, se afișează valorile pentru statistica calculată și cea teoretică. Apelarea funcției se face pentru un set de valori de la 1 la 300 și se testează egalitatea mediei reale din populație cu valoarea 165.

Rezultatele obținute:

"Media din populație nu este egală cu 165"

[1] "t calculat= -2.89517871653279"

[1] "t tabelar= -1.96792966906567"

Valoarea testului Student, folosind funcția *t.test*:

```

x<-(1:300)
t.test(x, mu=165)

```

Rezultatul afișat este același:  $t = -2.8952$ ,  $df = 299$ ,  $p\text{-value} = 0.004069$

*alternative hypothesis: true mean is not equal to 165*

### Coeficientul de corelație liniară

```
s<-0
s2<-0
s3<-0
corelatie<-function(x,y) {
  n<-length(x)
  xi<-x-mean(x)
  yi<-y-mean(y)
  m<-xi*yi
  for (i in 1:n){
    s<-s+m[i]
    s2<-s2+(xi[i])^2
    s3<-s3+(yi[i])^2
  }
  c<-s/sqrt(s2*s3)
  r<- paste("Coeficientul de corelatie liniara dinte x si y este:",
c)
  print(r)
}
x<-c(2,9,4.5,10.8,6,1.5,8.7)
y<-c(3,2.4,2.9,7,9,4.5,10.3)
corelatie(x,y)
```

Se creează o funcție cu numele *corelatie*. Variabilele *s*, *s2* și *s3*, din structura de control *for*, se inițializează pentru a se putea fi folosi în funcție. Funcția are ca argumente vectorii *x* și *y*. *n<-length(x)* calculează lungimea vectorului *x* (lungimea vectorului *y* coincide cu cea a lui *x*). Expresiile *xi<-x-mean(x)* și *yi<-y-mean(y)* calculează diferența dintre valorile lui *x*, *y* și mediile acestora, *m* reprezintă produsul dintre *xi* și *yi*. Cu ajutorul structurii de control *for*, se calculează 3 sume utilizate în formula corelației liniare: suma produselor dintre *xi* și *yi*, suma pătratelor diferențelor dintre *x* și *media(x)* și suma pătratelor diferențelor dintre *y* și *media(y)*. *c<-s/sqrt(s2\*s3)* calculează raportul dintre variabila *s* și radical din produsul variabilelor *s2* și *s3*, rezultatul reprezentând valoarea corelației. Pentru afișarea rezultatului se creează variabilă *r* cu ajutorul funcției *paste* care concatenează 2 componente de tip diferit.

Pentru a apela funcția, *x* primește valorile 2, 9, 4.5, 10.8, 6, 1.5, 8.7, iar *y* primește valorile 3, 2.4, 2.9, 7, 9, 4.5, 10.3. Coeficientul de corelație liniară dintre *x* și *y* este 0.423753.

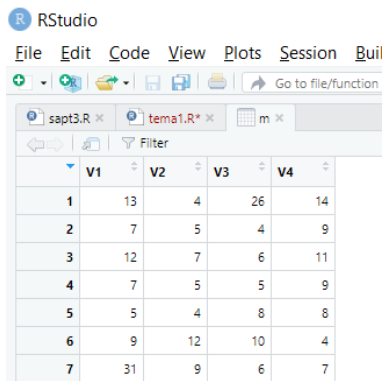
Valoarea coeficientului de corelație liniară este aceeași și în cazul folosirii funcției *cor*..

```
x<-c(2,9,4.5,10.8,6,1.5,8.7)
y<-c(3,2.4,2.9,7,9,4.5,10.3)
cor(x,y, use="everything",
     method=c("pearson", "kendall", "spearman"))
```

c) Scrieți un R Script care afișează *valoarea medie a vânzărilor în fiecare zi*

```
x<-c(13, 7, 12, 7, 5, 9, 31, 4, 5, 7, 5, 4, 12, 9, 26, 4, 6, 5,
8, 10, 6, 14, 9, 11, 9, 8, 4, 7)
m <- matrix(data=c(13, 7, 12, 7, 5, 9, 31, 4, 5, 7, 5, 4, 12, 9,
26, 4, 6, 5, 8, 10, 6, 14, 9, 11, 9, 8, 4, 7), nrow=7, ncol=4)
```

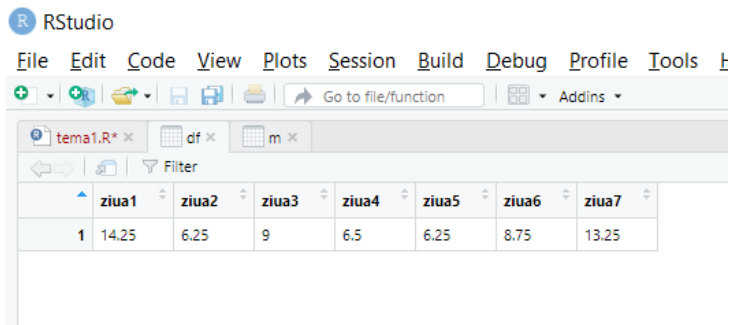
m



	V1	V2	V3	V4
1	13	7	12	7
2	5	9	31	4
3	4	5	7	5
4	4	12	9	26
5	4	6	5	8
6	5	8	10	6
7	14	9	11	9

```
df <- data.frame(ziua1 = mean(m[1,]), ziua2 = mean(m[2,]), ziua3
= mean(m[3,]), ziua4=mean(m[4,]), ziua5 = mean(m[5,]), ziua6 =
mean(m[6,]), ziua7 = mean(m[7,]))
```

df



	ziua1	ziua2	ziua3	ziua4	ziua5	ziua6	ziua7
1	14.25	6.25	9	6.5	6.25	8.75	13.25



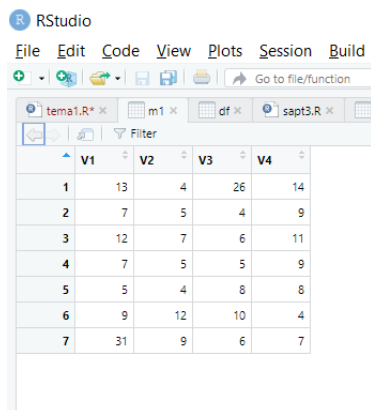
Se creează un *vector*,  $x$ , care ia valori, însemnând vânzările celor 4 magazine pentru cele 7 zile ale săptămânii. Mai apoi, valorile se pun într-o *matrice*,  $m$ , cu 7 linii și 4 coloane (liniile reprezintă zilele săptămânii iar coloanele reprezintă cele 4 magazine). După aceea, pentru a calcula media vânzărilor se creează un *data frame*,  $df$ , unde pe prima linie se pun cele 4 magazine, iar pe cea de-a doua linie se calculează media vânzărilor fiecărui magazin în cele 7 zile.

### Rezolvare alternativă cu o funcție din familia *apply*

```
xx <- c(13, 7, 12, 7, 5, 9, 31, 4, 5, 7, 5, 4, 12, 9, 26, 4, 6,
5, 8, 10, 6, 14, 9, 11, 9, 8, 4, 7)
```

```
m1 <- matrix(xx, nrow=7, ncol=4)
```

```
m1
```



	V1	V2	V3	V4
1	13	4	26	14
2	7	5	4	9
3	12	7	6	11
4	7	5	5	9
5	5	4	8	8
6	9	12	10	4
7	31	9	6	7

```
rez <- apply(m1, MARGIN=1, FUN=mean) #media pe linii
```

```
rez
```

Rezultatele:

```
14.25 6.25 9.00 6.50 6.25 8.75 13.25
```

Se creează un *vector*,  $xx$ , care ia valori însemnând vânzările celor 4 magazine pentru cele 7 zile ale săptămânii. Mai apoi, valorile se pun într-o *matrice*,  $m1$ , cu 7 linii și 4 coloane (liniile reprezintă zilele săptămânii iar coloanele reprezintă cele 4 magazine). Pentru a calcula media vânzărilor se folosește funcția *apply*, iar ca argumente se pun matricea  $m1$ ,  $margin=1$  însemnând ca se vor lua în considerare liniile,  $fun=mean$  funcția apelată este media valorilor. Astfel se va calcula media vânzărilor magazinelor, adică media valorilor de pe linii.

d) Analiză comparativă a funcțiilor din familia *apply*

Funcțiile	input	output
apply	array/matrice	vector/array/listă
lapply	vector/listă	listă
sapply	vector/listă	vector/matrice/array
mapply	vector/listă (lungime>0)/array	caracter/valoare logică
tapply	vector/data frame	valoare/array
vapply	vector/listă	Vector/matrice/array (tip de date pre-definit)

Sursele bibliografice parcurse:

- Manualul *R in a Nutshell*
- Documentația din *Rstudio*

- e) Descarcați de pe Eurostat evoluția anuală a unei variabile, pentru mai multe țări, în Excel. Transformați setul de date în format tidy data. Calculați valoarea medie a variabilei într-un an. Calculați valoarea medie a variabilei în fiecare an; rezultatele se afișează într-un tabel.

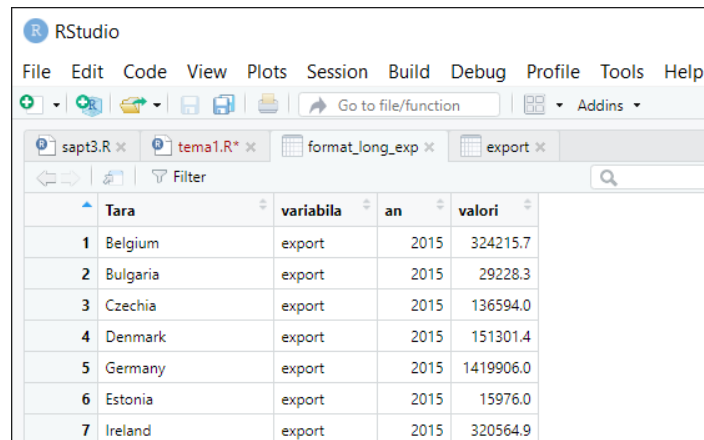
Am descărcat de pe Eurostat datele cu privire la exporturile de bunuri și servicii în milioane de euro, date cu frecvență anuală din perioada 2015-2021. Mai apoi, am importat baza de date în R.

```
tec00110_page_spreadsheet<-
read_excel("C:/Users/paula/Downloads/tec00110_page_spreadsheet.xlsx", +sheet = "Sheet 1")
export <- tec00110_page_spreadsheet
format_long_exp <- export %>%
  gather(key=an,
         value=valori,
         starts_with('export'),
         convert=TRUE)
view(format_long_exp)
```

	Tara	an	valori
1	Belgium	export-2015	324215.7
2	Bulgaria	export-2015	29228.3
3	Czechia	export-2015	136594.0
4	Denmark	export-2015	151301.4
5	Germany	export-2015	1419906.0
6	Estonia	export-2015	15976.0

Am transformat baza de date în format tidy data

```
format_long_exp <- format_long_exp %>%
  separate(an,
           into= c('variabila', 'an'),
           sep="-",
           convert=TRUE)
view(format_long_exp)
```



	Tara	variabila	an	valori
1	Belgium	export	2015	324215.7
2	Bulgaria	export	2015	29228.3
3	Czechia	export	2015	136594.0
4	Denmark	export	2015	151301.4
5	Germany	export	2015	1419906.0
6	Estonia	export	2015	15976.0
7	Ireland	export	2015	320564.9

Mai apoi, am calculat valoarea medie pentru anul 2015:

```
medie_export <- format_long_exp %>%
  filter(an==2015) %>%
  summarize(medie=mean(valori))
medie_export
```

Pentru calcularea mediei aferentă tuturor anilor prezenți în tabel, am realizat o funcție:

```
medie_export_ani <- function(x){medie_export_ani=format_long_exp
%>%
  filter(an==x) %>%
  summarize(medie=mean(valori)) ;
medie_export_ani}
```

După ce am realizat funcția, am apelat-o într-un array pentru anii 2015-2021

```
medii_exp <-
array(data=c(medie_export_ani(2015),medie_export_ani(2016),medie
_export_ani(2017),medie_export_ani(2018),medie_export_ani(2019),
medie_export_ani(2020),medie_export_ani(2021)))
medii_exp
```

Pentru afișarea mediilor într-un tabel, am creat un data frame:

```
medie_exp_toti_anii <-  
data.frame(medie2015=medii_exp[[1]],medie2016=medii_exp[[2]],med  
ie2017=medii_exp[[3]],medie2018=medii_exp[[4]],medie2019=medii_e  
xp[[5]],medie2020=medii_exp[[6]],medie2021=medii_exp[[7]])  
medie_exp_toti_anii
```

- f) Importați/citiți în R un HTML table care conține un set de date, de pe Wikipedia sau alta pagina Web. Transformați setul de date în forma tidy data, dacă este necesar, și efectuați două prelucrări statistice (medie..., grafice, tabele de frecvență, regresie, previziuni, ...).

Am importat în R un tabel HTML, incluzând informații despre costul de trai, cu funcția `read_html`:

```
pagina <-  
read_html("https://www.worlddata.info/cost-of-living.php")  
class(pagina)  
tabele <- pagina %>% html_nodes("table")  
length(tabele)  
indicator <- html_table(tabele[[1]])  
view(indicator)
```

În urma vizualizării tabelului, am constatat că setul de date nu necesită transformarea în forma tidy data. Prin urmare, nu am modificat nimic. Ca și prelucrări, am efectuat o medie a puterii de cumpărare și am realizat o regresie liniară pentru costul de trai și puterea de cumpărare.

#medie

```
medie_indicator <- indicator %>%  
  summarize(medie=mean(`Purchasing power index`))  
medie_indicator
```

#regresie

```
linear_model <-  
lm(`cost index`~`Purchasing power index`,data = indicator)  
linear_model
```

*Coefficients:*

<i>(Intercept)</i>	<i>`Purchasing power index`</i>
34.2051	0.6905

Mai exact, regresia poate fi scrisă:  $34.2051 + 0.6905 * \text{Purchasing power index}$

Am aplicat funcția `summary` asupra regresiei, astfel putem descoperi statisticile descriptive:

```
summary(linear_model)
```

*Residuals:*

<i>Min</i>	<i>1Q</i>	<i>Median</i>	<i>3Q</i>	<i>Max</i>
-66.895	-13.126	-2.388	8.534	75.525

*Coefficients:*

	<i>Estimate</i>	<i>Std. Error</i>	<i>t</i>	<i>value Pr(&gt; t )</i>
<i>(Intercept)</i>	34.20510	3.51292	9.737	2.61e-16 ***
<i>`Purchasing power index`</i>	0.69052	0.06813	10.136	2e-16 ***

*Signif. codes:* 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

*Residual standard error:* 23.33 on 104 degrees of freedom

*Multiple R-squared:* 0.4969, *Adjusted R-squared:* 0.4921

*F-statistic:* 102.7 on 1 and 104 DF, *p-value:* < 2.2e-16

## Problema 2

- a) Pentru această problemă am importat de pe site-ul Kaggle baza de date **”Students Performance in Exams”**. Baza de date conține 8 variabile și 1000 de observații. În primă fază voi descrie pe scurt variabilele.

*Gender* – este o variabilă categorială și reprezintă sexul persoanelor;

*Race/ethnicity* – este o variabilă categorială și indică etnia respondenților clasificate în 5 grupe;

*Parental level of education* – este o variabilă categorială și arată nivelul studiilor părinților;

*Lunch* – este o variabilă categorială care reprezintă tipul abonamentelor pentru masa de prânz;

*Test preparation course* – este o variabilă categorială și indică stadiul fiecărui student cu privire la cursurile de pregătire pentru test;

*Math score* – este o variabilă cantitativă și arată scorul obținut la matematică;

*Reading score* – este o variabilă cantitativă și semnifică scorul obținut la partea de citire;

*Writing score* – este o variabilă cantitativă și indică scorul obținut la proba de scriere;

```
StudentsPerformance<-
read_excel("C:/Users/paula/Desktop/3.2/Analiza statistica in
R/Proiect/StudentsPerformance.xls")
> View(StudentsPerformance)
> stud <- StudentsPerformance
```

În momentul în care am importat baza de date am observat că tipul variabilelor coincide cu tipul descris anterior, deci nu mai era necesară nicio verificare a tipului datelor. În continuare, voi transforma variabilele categoriale în variabile de tip factor, astfel voi transforma doar variabile "gender", "race/ethnicity", "parental level of education", "lunch", "test preparation score".

```
gen <- c("female", "male")
stud$gender <- parse_factor(stud$gender, levels=gen)

ethnicity <- c("group A", "group B", "group C", "group D", "group
E")
stud$`race/ethnicity` <- parse_factor(stud$`race/ethnicity`,
levels=ethnicity)

education <- c("associate's degree", "bachelor's degree", "high
school", "master's degree", "some college", "some high school")
stud$`parental level of education` <- parse_factor(stud$`parental
level of education`, levels=education)

lunch <- c("free/reduced", "standard")
stud$lunch <- parse_factor(stud$lunch, levels=lunch)

test <- c("none", "completed")
stud$`test preparation course` <- parse_factor(stud$`test
preparation course`, levels=test)
```

## b) Funcția filter

Am utilizat funcția "Filter" pentru a filtra datele, astfel încât să se afișeze genul studenților, nivelul studiilor părinților și scorul la matematică, în condițiile în care scorul este mai mare decât 85. Prin urmare, se vor observa studenții cu cel mai mare scor, ordinea fiind aleatoare.

```
scor<- stud %>%
  select(`gender`,`parental level of education`,`math score`)%>%
  filter(`math score`>=85)
scor
```

```

# A tibble: 117 x 3
  gender `parental level of education` `math score`
  <fct>   <fct>                        <dbl>
1 female master's degree                90
2 female some college                   88
3 male   high school                    88
4 male   some college                   97
5 male   high school                    88
6 female associate's degree             85
7 male   some college                   98
8 female master's degree                87
9 female bachelor's degree             99
10 male  associate's degree             91
# ... with 107 more rows

```

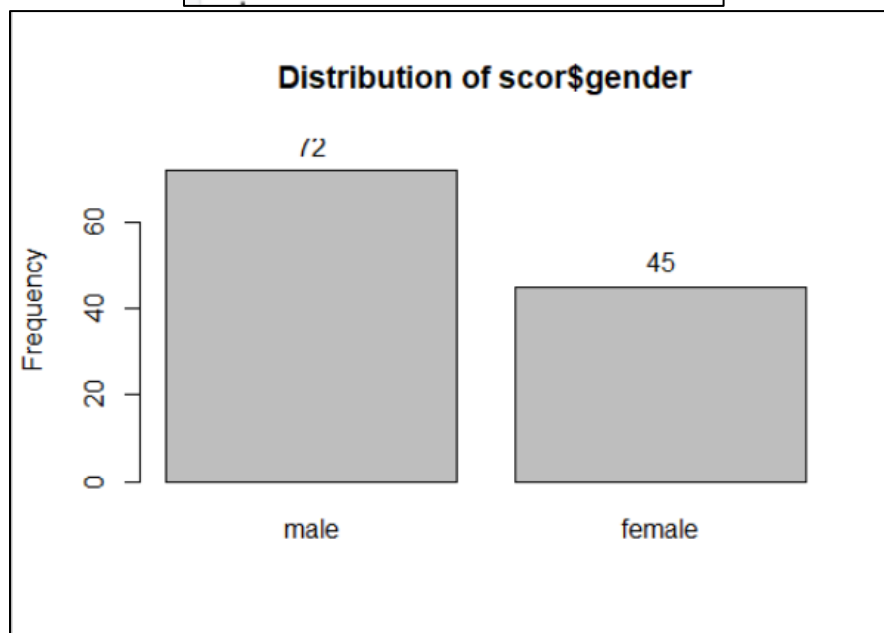
Am creat un tabel de frecvențe pentru a vedea diferența dintre bărbați și femei cu privire la cele mai bune scoruri la matematică, iar după cum se poate observa bărbații domină femeile, cu un procent de 59.4%.

```

tab1(scor$gender, sort.group = "decreasing", cum.percent = FALSE
)

```

	Frequency	Percent
male	72	61.5
female	45	38.5
Total	117	100.0



Am modificat tabelul "scor" pentru a afișa în ordine descrescătoare studenții în funcție de scorul acestora. Pentru aceasta am folosit funcția *arrange*.

```

scor_descrescator <- arrange(scor, desc(`math score`))
scor_descrescator

```

	gender	parental level of education	math score
	<fct>	<fct>	<dbl>
1	male	associate's degree	100
2	female	some college	100
3	female	bachelor's degree	100
4	male	some college	100
5	male	some college	100
6	male	bachelor's degree	100
7	female	associate's degree	100
8	female	bachelor's degree	99
9	female	high school	99
10	male	some college	99
#	... with 107 more rows		

c) Aplicați 6 funcții din familia *str\_...* din pachetul *stringr*, în care utilizați următoarele meta-caractere: `.` `|` `{` `}` `[` `]` `^` `$` `-` `*` `+` `?` pentru a construi regular expressions.

1. Se creează un vector *x*. Se vor detecta valorile care încep cu litera S, utilizând funcția *str\_detect* și metacaracterul `^`. Rezultatul afișat va fi: Spania, Suedia.

```
x<-c("Italia" , "Portugalia", "Spania", "Suedia", "15.2")
str_view(x, "Italia|Portugalia")
str_detect(x, "^S")
```

2. Funcția *str\_replace\_all* va înlocui cu spațiu, în textul inițial, simbolul `\`, dacă se găsește o dată sau de mai multe ori. Se utilizează metacaracterul `+`. Rezultatul afișat va fi „Analiza statistica in R”

```
text<-"Analiza\\statistica\\in\R"
str_replace_all(text, "\\s+", " ")
```

3. Funcția *str\_extract* va extrage din vector textul care are forma evolua sau evalua. Se utilizează metacaracterul `|` și `()` pentru a alege una dintre literele *o* sau *a*. Rezultatul afișat va fi: evolua, avalua.

```
str_extract(c("evolua", "evalua"), "ev(o|a)lua")
```

4. Funcția *str\_count* va număra de câte ori apare litera A sau a în fiecare nume. Se utilizează metacaracterul `[ ]`. Rezultatul afișat va fi: 2, 1, 2, pentru că în Ana apare atât A, cât și a.

```
str_count(c("Ana", "Andrei", "Maria"), "[Aa]" )
```



5. Funcția `str_subset` va afișa șirul care va conține denumirile de țări care se termină cu litera a. Se folosește metacaracterul \$, pentru caracterele de la sfârșit. Rezultatul afișat va fi: *Italia Portugalia Spania Suedia*, ultimul element 15.2 nu va fi afișat.

```
x<-c("Italia" , "Portugalia", "Spania", "Suedia", "15.2")
str_subset(x, "a$")
```

6. Funcția va detecta din șirul x, doar elementele care conțin "2" de 0 sau 1 ori, utilizând metacaracterul ?. Rezultatul afișat va fi: 15.2.

```
x<-c("Italia" , "Portugalia", "Spania", "Suedia", "15.2")
str_detect(x, "2?")
```

- d) Formulați obiective (întrebări) asupra datelor și găsiți răspunsul, utilizând următoarele analize preliminare (de tipul analiza exploratorie a datelor): grafice, descriptive, corelații, medii condiționate și anova, testul chi-square, regresie.

### #Statisticile descriptive

Urmăresc ca obiectiv să analizez care este media probelor de citire, matematică și scris, care este scorul maxim și cel minim. Pentru a vedea statisticile descriptive ale variabilelor de tip cantitativ am apelat funcția `summary`. Pentru variabilele de tip caracter s-a afișat numărul de observații pentru fiecare grup. Astfel scorul mediu pentru examenul de matematică este 66.09, valoarea maximă fiind 100 de puncte, iar minimul 0. Pentru proba de citire, scorul mediu este 69.17 puncte, minimul fiind 17, iar maximum 100. Pentru proba de scriere, scorul mediu este 68.05 puncte, cu valoarea maximă fiind de 100 de puncte și minimul de 10 puncte. Variabila `gender` este categorială, femeii sunt 518, iar bărbații 482.

```
stud %>% summary()
```

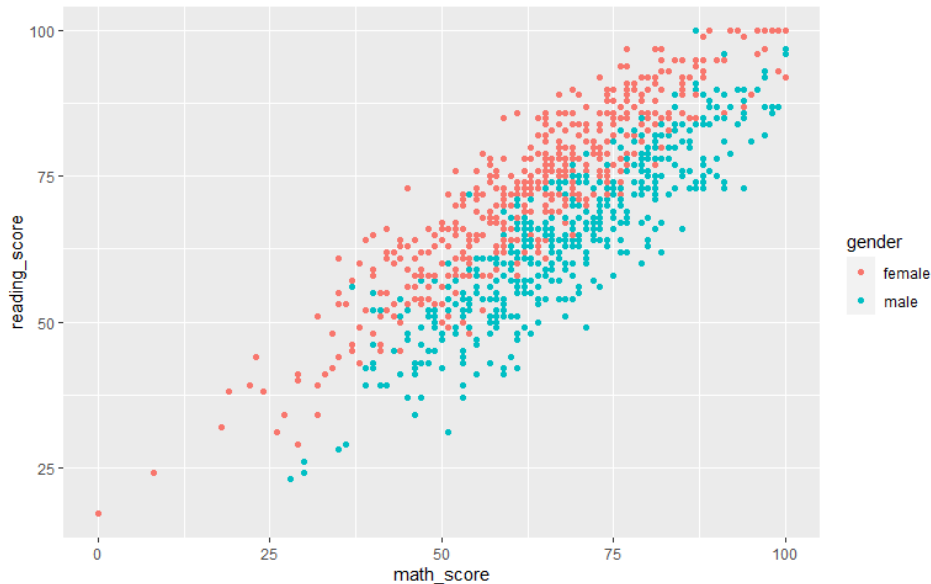
### Rezultatele afișate:

gender	race_ethnicity	parents_education	lunch	test_preparation_course	math_score
female:518	group A: 89	associate's degree:222	free/reduced:355	none :642	Min. : 0.00
male :482	group B:190	bachelor's degree :118	standard :645	completed:358	1st Qu.: 57.00
	group C:319	high school :196			Median : 66.00
	group D:262	master's degree : 59			Mean : 66.09
	group E:140	some college :226			3rd Qu.: 77.00
		some high school :179			Max. :100.00
reading_score	writing_score				
Min. : 17.00	Min. : 10.00				
1st Qu.: 59.00	1st Qu.: 57.75				
Median : 70.00	Median : 69.00				
Mean : 69.17	Mean : 68.05				
3rd Qu.: 79.00	3rd Qu.: 79.00				
Max. :100.00	Max. :100.00				

## #Grafice

**Este vreo legătură între genul persoanelor și scorul obținut la testul de matematică și citire?**

```
ggplot(stud, aes(x = math_score, y = reading_score)) +  
geom_point(aes(colour = gender))
```



În baza rezultatului afișat se poate concluziona că scorul obținut pentru proba de matematică și cel pentru testul de citire sunt corelate pozitiv, cu cât e mai mare scorul obținut la testul de matematică, cu atât și scorul la proba de citire va fi mai mare. Însă, există legătură și între genul persoanelor și scorul obținut pentru fiecare test, după cum se poate observa din grafic. Pentru același scor obținut în cadrul testului de matematică, fetele tind să aibă un scor mai mare în cadrul probei de citire, pe când, băieții tind să aibă un scor mai mare pentru testul de matematică, la același scor pentru probei de citire. Astfel, băieții au rezultate mai bune la matematică, iar fetele la proba de citire.

## #Medii condiționate și ANOVA

**Se diferențiază media scorului obținut pentru proba de scriere în funcție de etnia elevului?**

```
group_by(stud, race_ethnicity) %>%  
  summarise(  
    mean = mean(writing_score, na.rm = TRUE),  
    sd = sd(writing_score, na.rm = TRUE)  
  )
```

	race_ethnicity	mean	sd
	<fct>	<dbl>	<dbl>
1	group A	62.7	15.5
2	group B	65.6	15.6
3	group C	67.8	15.0
4	group D	70.1	14.4
5	group E	71.4	15.1

În acest tabel sunt afișate mediile și abaterile standard ale testului de scriere pentru fiecare grup etnic. Pentru a răspunde la întrebare vom rula un test ANOVA. Ipoteza nulă este:  $H_0$ : *valorile medii ale scorului obținut pentru proba de scriere nu diferă în cele 5 categorii.*

```
anova<-aov(formula = writing_score ~ race_ethnicity, data= stud)
summary(anova)
```

Rezultatul afișat:

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
race_ethnicity	4	6456	1614.0	7.162	1.1e-05	***
Residuals	995	224221	225.3			
---						
signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1						

Conform rezultatelor afișate, ipoteza nulă se respinge, cu un prag de risc de 1%, întrucât p-value=0.000 este mai mică decât orice prag de risc. Respectiv, putem concluziona că valorile medii diferă în cele 5 categorii, adică se diferențiază media scorului pentru proba de scriere în funcție de etnie.

### #Testul Person chi-square

**Există asociere între variabilele test\_preparation\_course(curs de pregătire) și parents\_education (nivelul de educație al părinților)?**

```
ass<-assocstats(table(stud$test_preparation_course,
stud$parents_education))
summary(ass)
mosaic(~ test + education,
data = stud, shade = TRUE)
```

Rezultatele afișate:

```
Number of cases in table: 1000
Number of factors: 2
Test for independence of all factors:
    chisq = 9.544, df = 5, p-value = 0.08923
      X^2 df P(> X^2)
Likelihood Ratio 9.5966 5 0.087507
Pearson          9.5441 5 0.089234

Phi-Coefficient : NA
Contingency Coeff.: 0.097
Cramer's V      : 0.098
```

Conform rezultatelor, Ipoteza nulă,  $H_0$ : nu există legătură dintre variabile, acestea sunt independente, se respinge pentru un prag de risc de 10%. Respectiv, există legătură între variabile.

- e) Pornind de la rezultatele anticipate la d), propuneți și estimați două modele de regresie pentru aceeași variabilă dependentă, ce includ cel puțin o variabilă categorială. Alegeți-l pe cel mai potrivit după criteriile mape și rmse. Testați semnificativitatea, interpretați coeficienții, salvați predicțiile în setul de date și analizați comportamentul erorilor de predicție (reziduurilor) din perspectiva ipotezelor econometrice.

### Model 1

Primul model de regresie are ca variabilă dependentă scorul testului de matematică și ca variabile independente: scorul probei de citire și genul, care e variabila factor, de referință fiind categoria- băieți.

```
modell1<- lm(math_score ~ reading_score+ factor2(gender), data =  
stud)  
modell1
```

Coefficients:					
	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	6.6384	1.0078	6.587	7.26e-11	***
reading_score	0.9483	0.0147	64.523	< 2e-16	***
factor(gender)female	-11.8614	0.4292	-27.634	< 2e-16	***
---					
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1					
Residual standard error: 6.577 on 997 degrees of freedom					
Multiple R-squared: 0.8122, Adjusted R-squared: 0.8119					
F-statistic: 2157 on 2 and 997 DF, p-value: < 2.2e-16					

Toți coeficienții sunt semnificativi, p-value este mai mic decât oricare prag de risc, deci ipoteza nulă se respinge. Dacă crește scorul probei de citire cu 1, crește și scorul testului de matematică cu 0.9483. De asemenea, fetele au, în medie, cu 11.86 puncte mai puțin decât băieții la testul de matematică.

### Model 2

Al doilea model de regresie are ca variabilă dependentă scorul probei de matematică și ca variabile independente: scorul probei de citire, genul, care e variabila factor, de referință fiind categoria- băieți și etnia, care e variabilă factor, categoria de referință fiind grupul A.

```
Model2<-lm(math_score~
reading_score+factor(race_ethnicity)+factor(gender), data = stud)
summary(model2)
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)      6.02356    1.13153    5.323 1.26e-07 ***
reading_score      0.93285    0.01443   64.626 < 2e-16 ***
factor(race_ethnicity)group B  0.99398    0.82248    1.209  0.2271
factor(race_ethnicity)group C  0.56943    0.76957    0.740  0.4595
factor(race_ethnicity)group D  1.76328    0.78709    2.240  0.0253 *
factor(race_ethnicity)group E  5.43113    0.87342    6.218 7.40e-10 ***
factor(gender)female    -11.68284    0.41877  -27.898 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.385 on 993 degrees of freedom
Multiple R-squared:  0.8238,    Adjusted R-squared:  0.8227
F-statistic: 773.7 on 6 and 993 DF,  p-value: < 2.2e-16
```

Conform rezultatelor afișate, studenții din grupa E au media scorului pentru proba de matematică mai mare decât cei din grupa etnică A. Coeficienții pentru grupele B și C nu sunt semnificativi, deoarece probabilitatea este foarte mare, respectiv, ipoteza nulă se acceptă, deci, scorul mediu la matematică nu se diferențiază foarte mult între aceste grupe față de grupa de referință.

Pentru a alege între modele, am comparat MSE-urile și RMSE\_urile.

```
mape(model1, stud)
rmse(model1, stud)
```

Rezultatele obținute pentru modelul 1 sunt : MSE= 43.12562 și RMSE= 6.56701. Pentru modelul 2, MSE= 40.47653 și RMSE= 6.362117. Modelul 2 are un MSE și RMSE mai mic, prin urmare, este mai potrivit decât primul model.

Am salvat predicțiile și erorile în setul de date.

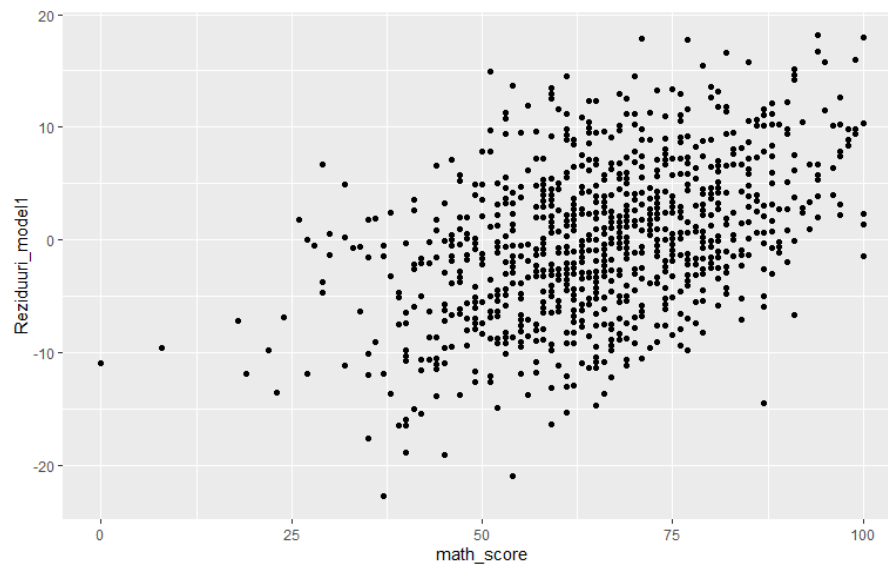
```
stud<- stud %>%
  add_predictions(model1, var = "Predictii_model1") %>%
  add_residuals(model1, var = "Reziduuri_model1")
stud<- stud %>%
  add_predictions(model2, var = "Predictii_model2") %>%
  add_residuals(model2, var = "Reziduuri_model2")
```

### Analiza comportamentului erorilor de predicție:

Pentru ca modelul să fie adecvat punctele pe grafic trebuie să fie aleator distribuite, astfel încât să demonstreze că nu mai există informații importante în erori ce trebuie incluse în model.

Astfel, se poate observa că pentru ambele modele punctele sunt puțin grupate în jurul unei drepte. Astfel, putem concluziona că ambele modele ar putea fi îmbunătățite.

```
ggplot(stud, aes(math_score, Reziduuri_model1)) + geom_point()
```



```
ggplot(stud, aes(math_score, Reziduuri_model2)) + geom_point()
```

