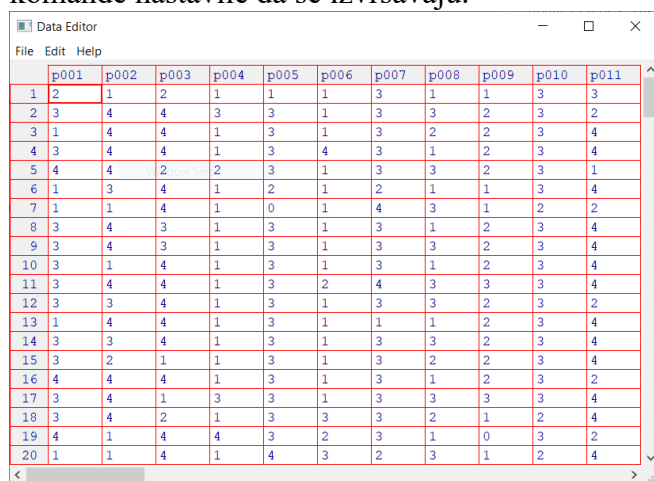


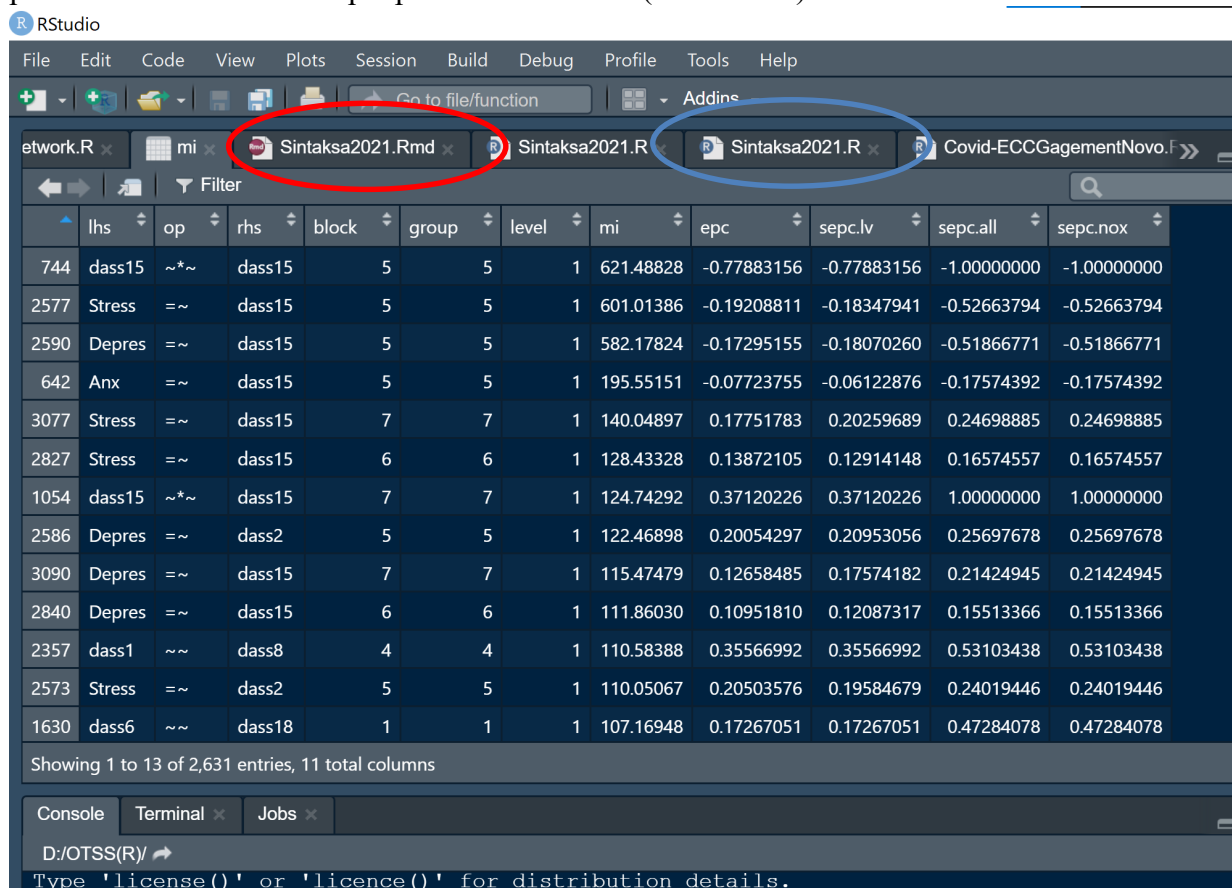
Par dodatnih objašnjenja...

Nekada je potrebno da zatvorite prozor data editora (otvoren komandama fix ili edit) da bi komande nastavile da se izvršavaju.



	p001	p002	p003	p004	p005	p006	p007	p008	p009	p010	p011
1	2	1	2	1	1	1	3	1	1	3	3
2	3	4	4	3	3	1	3	3	2	3	2
3	1	4	4	1	3	1	3	2	2	3	4
4	3	4	4	1	3	4	3	1	2	3	4
5	4	4	2	2	3	1	3	3	2	3	1
6	1	3	4	1	2	1	2	1	1	3	4
7	1	1	4	1	0	1	4	3	1	2	2
8	3	4	3	1	3	1	3	1	2	3	4
9	3	4	3	1	3	1	3	3	2	3	4
10	3	1	4	1	3	1	3	1	2	3	4
11	3	4	4	1	3	2	4	3	3	3	4
12	3	3	4	1	3	1	3	3	2	3	2
13	1	4	4	1	3	1	1	1	2	3	4
14	3	3	4	1	3	1	3	3	2	3	4
15	3	2	1	1	3	1	3	2	2	3	4
16	4	4	4	1	3	1	3	1	2	3	2
17	3	4	1	3	3	1	3	3	3	3	4
18	3	4	2	1	3	3	3	2	1	2	4
19	4	1	4	4	3	2	3	1	0	3	2
20	1	1	4	1	4	3	2	3	1	2	4

Takođe, ako skript koji ste pokrenuli sadrži komandu View, može se desiti da vam se u panelu Source umesto skripta prikaže data.frame (vidite sliku):



RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

etwork.R x mi x Sintaksa2021.Rmd x Sintaksa2021.R x Covid-ECCGagementNovo.f

Filter

	lhs	op	rhs	block	group	level	mi	epc	sepc.lv	sepc.all	sepc.nox
744	dass15	~*	dass15	5	5	1	621.48828	-0.77883156	-0.77883156	-1.00000000	-1.00000000
2577	Stress	=~	dass15	5	5	1	601.01386	-0.19208811	-0.18347941	-0.52663794	-0.52663794
2590	Depres	=~	dass15	5	5	1	582.17824	-0.17295155	-0.18070260	-0.51866771	-0.51866771
642	Anx	=~	dass15	5	5	1	195.55151	-0.07723755	-0.06122876	-0.17574392	-0.17574392
3077	Stress	=~	dass15	7	7	1	140.04897	0.17751783	0.20259689	0.24698885	0.24698885
2827	Stress	=~	dass15	6	6	1	128.43328	0.13872105	0.12914148	0.16574557	0.16574557
1054	dass15	~*	dass15	7	7	1	124.74292	0.37120226	0.37120226	1.00000000	1.00000000
2586	Depres	=~	dass2	5	5	1	122.46898	0.20054297	0.20953056	0.25697678	0.25697678
3090	Depres	=~	dass15	7	7	1	115.47479	0.12658485	0.17574182	0.21424945	0.21424945
2840	Depres	=~	dass15	6	6	1	111.86030	0.10951810	0.12087317	0.15513366	0.15513366
2357	dass1	~*	dass8	4	4	1	110.58388	0.35566992	0.35566992	0.53103438	0.53103438
2573	Stress	=~	dass2	5	5	1	110.05067	0.20503576	0.19584679	0.24019446	0.24019446
1630	dass6	~*	dass18	1	1	1	107.16948	0.17267051	0.17267051	0.47284078	0.47284078

Showing 1 to 13 of 2,631 entries, 11 total columns

Console Terminal x Jobs x

D:/OTSS(R) ↗

Type 'license()' or 'licence()' for distribution details.

U tom slučaju samo kliknite na jezičak sa skriptom (zaokruženo crvenim ili plavim u zavisnosti od datoteke koje koristite), a jezičak sa podacima možete i ugaziti (kada vidite šta vas je zanimalo).

Kroz skript se možete kretati i preko markera u dnu panela Source:

```
1241 Kada je specifikovana biva zadržana
1242 ```{r}
1243 novipodaci3.long <- melt(podaci2.wide, id.vars=c("ispitanik","pol",
1244 "neponovljeno"), measure.vars=c("merenje1", "merenje2", "merenje3" ),
1245 variable.name="merenje")
1246 head(novipodaci3.long)
1247 ```
1248 ### SLAJD 63 ###
1249 ## Vraćanje u WIDE
1250 Transformacija funkcijom dcast() iz paketa maditr.
1251 ```{r}
1252 p_load(maditr)
1253 novipodaci3.wide<-dcast(novipodaci3.long, ispitanik + pol ~ merenje,
1254 value.var="value")
1255 # Nije specifikovana varijabla neponovljeno pa je izbačena
1256 head(novipodaci3.wide)
1257 ```
1246:1 # SLAJD 63 ###
```

Kliknete na marker (na slici „SLAJD 63“ i sa liste koja se pojavi odaberete željeni marker (lokaciju u sintaksi).

Tekst sa PowerPoint slajdova se u skriptu nalazi u obliku komentara. Prezentacija vam verovatno neće biti ni potrebna, pošto više teksta ima u komentarima nego na prezentaciji. Ako imate pitanja ili neki problem obratite mi se na mail: janicic@ff.uns.ac.rs. Srećno!

Definisanje sopstvenih funkcija

Na primeru funkcije za normalizaciju varijable biće opisano kreiranje funkcija.

Funkciju kreiramo tako što je smestimo u objekat sa željenim imenom:

```
normalizuj<-function (nenormalna) {
  qnorm((rank(nenormalna, na.last = "keep", ties.method =
"min"))/length(nenormalna))
}
```

normalizuj je naziv funkcije (objekat)

Sa „<-function“ kažemo da će objekat „normalizuj“ biti funkcija.

U zagradama () stoji naziv varijable ili više njih koje će funkcija obrađivati – ovde je to varijabla „nenormalna“ (misli se na varijablu čija distribucija odstupa od normalne).

U vitičastim zagradama {} se definiše šta će funkcija da radi. Tu ide spisak naredbi koje će funkcija izvršavati.

Razbićemo po fazama šta funkcija normalizuj radi:

```
1. qnorm((rank(nenormalna, na.last = "keep", ties.method =
"min"))/length(nenormalna))
```

```
rank(nenormalna, na.last = "keep", ties.method = "min")
```

Ovaj deo će prvo rangirati rezultate na varijabli „nenormalna“ tako da niže vrednosti dobiju niže rangove (1, 2, 3...)

Nedostajući podaci biće zadržani, ali bez ranga (odnosno biće „na“) `na.last = "keep"`

Ako ima jednakih vrednosti one će dobiti isti rang i to najniži. Npr. imamo vrednosti: 11, 15, 16, 16, 18, one će dobiti sledeće rangove 11=1, 15=2, 16=3, 16=3, a 18=5. 4 se preskače jer pre vrednosti 18 već ima 4 niža rezultata, pa je rezultat 18 peti po redu. `ties.method =`

```
"min"
```

```
2. qnorm((rank(nenormalna, na.last = "keep", ties.method =  
"min"))/length(nenormalna))
```

Da bismo rangove pretvorili u proporciju rezultata do pripadajuće vrednosti, potrebno je svaki rang podeliti ukupnim brojem rezultata što radi funkcija `length(nenormalna)`

3. Rezultat ovog dela `rank(nenormalna, na.last = "keep", ties.method = "min"))/length(nenormalna)` funkcije koju kreiramo je, dakle proporcija rezultata do određene vrednosti.

4. Ostalo je samo da dobijene proporcije pretvorimo u z vrednosti koje odgovaraju datim proporcijama. To radi funkcija `qnorm()` odnosno
`qnorm((rank(nenormalna, na.last = "keep", ties.method =
"min"))/length(nenormalna))`

Da bi vam novokreirana funkcija bila dostupna za izvršavanje, morate je prvo pokrenuti (svaki put kad pokrenete R, a za to je potrebno da sintaksu/skript sačuvate).

Pokrenete (pozicionirate se u prvu liniju funkcije i sa ENTER pokrenete u R-u ili

Ctrl+ENTER ili ikonicom Run u RStudio):

```
normalizuj<-function (nenormalna) {  
  qnorm((rank(nenormalna, na.last = "keep", ties.method =  
"min"))/length(nenormalna))  
}
```

Dalje je izvršavate tako što pozovete ime funkcije, a u zagradi stavite naziv varijable koju želite da normalizujete.

Npr. ako želite da normalizujete varijablu "visina" onda biste kucali :

```
normalizuj(visina)
```

Interna varijabla « nenormalna » će dobiti vrednost varijable « visina » i funkcija će biti izvršena na njoj.

Rezultat funkcije će biti vektor vrednosti koji odgovara normalizovanim vrednostima varijable « visina ».

Ako želite da rezultat funkcije upišete u neki objekat, recimo varijablu n.visina to možete uraditi na sledeći način :

```
n.visina<-normalizuj(visina)
```

LDF I WDF

LDF ili Long Data Format je format podataka u kojem postoje ponovljena merenja nekih varijabli, ali su data u okviru jedne kolone. Mora postojati identifikacija ispitanika i i merenja. Na primer:

	ispitanik	pol	merenje	mereno
1	1	M	merenje1	7.9
2	1	M	merenje2	12.3
3	1	M	merenje3	10.7
4	2	Z	merenje1	6.3
5	2	Z	merenje2	10.6
6	2	Z	merenje3	11.1
7	3	Z	merenje1	9.5
8	3	Z	merenje2	13.1
9	3	Z	merenje3	13.8
10	4	M	merenje1	11.5
11	4	M	merenje2	13.4
12	4	M	merenje3	12.9

Svaki ispitanik ima podatke u onoliko redova koliko ima ponovljenih merenja. Na varijablama koje su merenje jednom vrednosti se ponavljaju (npr. Pol). Postoji identifikacija merenja (varijabla Merenje). Ovakav tip matrice podataka često se koristi u R-u u linearnim modelima sa ponovljenim merenjima.

WDF ili Wide Data Format je format podataka u kojem postoje ponovljena merenja nekih varijabli, a ponovljena merenja iste varijable data su u posebnim kolonama. Svi podaci za jednog ispitanika dati su u jednom redu. Ovaj tip matrica podataka se obično koristi u komercijalnom statističkom softveru kao što su SPSS i Statistica. Na primer :

ispitanik	pol	merenje1	merenje2	merenje3
1	M	7.9	12.3	10.7
2	Z	6.3	10.6	11.1
3	Z	9.5	13.1	13.8
4	M	11.5	13.4	12.9

Opet malo o kreiranju funkcija.

Kreiramo funkciju koja računa korekciju za atenuaciju (procenu korelacije između dve varijable kada bi one bile izmerene savršeno pouzdano).

```
Aten<-function(kor, rtt1=1, rtt2=1) {
  if (kor<(-1) | kor>1 | !is.numeric(kor)) {
    stop("Neispravna vrednost korelacije!")
  }
  if (rtt1<0 | rtt1>1 | !is.numeric(rtt1) | rtt2< 0 | rtt2>1 |
!is.numeric(rtt2)) {
    stop("Neispravna vrednost pouzdanosti!")
  }
  cat("Korelacija sa korekcijom za atenuaciju iznosi:",
round(kor/sqrt(rtt1*rtt2), 3))
}
aten(0.2, 0.62, 0.7) #pozivanje funkcije
```

Obratite pažnju da funkcija uzima tri varijable, gde je **kor** korelacija dva testa, a **rtt1** i **rtt2** su njihove pouzdanosti.

rtt1=1, rtt2=1 u funkciji označava da će ukoliko se ne navedu vrednosti pouzdanosti testova funkcija podrazumevati da je njihova pouzdanost savršena, odnosno jednaka 1 (takozvana *default* vrednost je 1).

Slično, **kor=0** znači da ako zaboravite da unesete korelaciju dve varijable, funkcija će podrazumevati vrednost 0.

Izrazi **if {...}** testiraju da li su unete vrednosti korelacije i pouzdanosti moguće i numeričke i ako nisu, prekidaju izvršenje funkcije `stop("Neispravna vrednost korelacije/pouzdanosti!")` ispisujući poruku. Primetite da je korišćen operator **|** (ili) što znači da je dovoljno da je bilo koji uslov zadovoljen da se funkcija prekine (uslovi: da je korelacija koja je uneta manja od -1, veća od 1 ili nije broj, da je bilo koja od dve unete pouzdanosti manja od 0 ili veća od 1 ili nije broj).

Funkcija **cat()** služi za ispisivanje teksta i spaja *argumente* odvojene zarezima u tekstualni vektor koji se ispisuje na ekranu. Argumenti mogu biti i numerički. Ovde su argumenti: tekst **"Korelacija sa korekcijom za atenuaciju iznosi:"** i rezultat funkcije **round(kor/sqrt(rtt1*rtt2), 3)** – što je numerička vrednost.

Funkciju **cat()** možete koristiti za pisanje naslova u ispisu i na grafikonima. Prelom reda (dodavanje novog reda) u okviru funkcije se pravi sa **\n**. Tekstualni deo mora biti pod navodnicima.

Funkcija **round()** zaokružuje numeričku vrednost na određeni broj decimala. Argumenti su **voj** : neka numerička vrednost i željeni broj decimala. Numerička vrednost koju smo prosledili ovoj funkciji je rezultat izračunavanja **kor/sqrt(rtt1*rtt2)**. Praktično, korelaciju dva testa podelili smo kvadratnim korenom (**sqrt**) proizvoda njihovih pouzdanosti (odnosno geometrijskom sredinom njihovih pouzdanosti) i na taj način dobili procenu korelacije dva merena konstrukta kada bi testovi kojima smo ih merili bili savršeno pouzdani.

Petlja For

```
for (i in 1:10) {  
  podaci2[,i+2]=as.numeric(podaci2[,i+2]);  
}
```

Petlja `for` ponavlja naredbu sve dok je `i` u zadatom okviru (ovde od 1 do 10). Svakim ponavljanjem `i` postaje za 1 veći.

U ovom slučaju pretvara varijable od 3 do 12 iz `data.frame` `podaci2` u numerički tip `as.numeric()`.

`i` može uzeti vrednosti od 1 do 10. U prvom izvršavanju `i` je 1. To znači da komanda

```
podaci2[,i+2]=as.numeric(podaci2[,i+2]);
```

praktično znači

```
podaci2[,1+2]=as.numeric(podaci2[,1+2]);
```

odnosno

```
podaci2[,3]=as.numeric(podaci2[,3]);
```

To znači: Neka 3. kolona `data.frame`a `podaci2` postane numerička

U drugom ponavljanju `i` postaje 2, a komanda praktično znači

```
podaci2[,2+2]=as.numeric(podaci2[,2+2]);
```

odnosno

```
podaci2[,4]=as.numeric(podaci2[,4]);
```

To znači: Neka 4. kolona `data.frame`a `podaci2` postane numerička

Naredba će se ponavljati dok `i` ne postane 10 i posle toga će stati. Gornju petlju smo mogli napisati i ovako:

```
for (i in 3:12) {  
  podaci2[,i]=as.numeric(podaci2[,i]);  
}
```

a rezultat bi bio isti, samo što bi `i` uzimalo vrednosti od 3 do 12.