

# ***Navodila za uporabo programskega okolja R***

## ***Ustvarjanje in upravljanje s podatki***

### **KAZALO**

1	USTVARJANJE PODATKOV .....	1
1.1	Vnos podatkov preko tipkovnice .....	1
1.2	Uvažanje podatkov iz besedilne datoteke .....	3
1.3	Uvažanje Excelovih datotek .....	4
1.4	Uvažanje SPSS datotek .....	5
2	OSNOVNO UPRAVLJANJE S PODATKI V R .....	7
2.1	Imena stolpcev .....	7
2.2	Preurejanje vrstnega reda stolpcev .....	7
2.3	Izločanje določenega dela podatkov .....	8
2.3.1	Izbiranje (ohranjanje) spremenljivk .....	8
2.3.2	Izključevanje (opuščanje) spremenljivk .....	9
2.3.3	Izbiranje določenega dela opazovanj .....	9
2.3.4	Hkratna izbira spremenljivk in opazovanj .....	10
2.4	Spajanje podatkovnih okvirjev .....	11
2.4.1	Dodajanje stolpcev (horizontalno spajanje) .....	11
2.4.2	Dodajanje vrstic (vertikalno spajanje) .....	11
2.5	Uporabne funkcije za delo s podatkovnimi objekti .....	12
3	DODAJANJE PRIPISOV PODATKOM .....	12
3.1	Oznake spremenljivk .....	13
3.2	Oznake vrednosti spremenljivk .....	13

### **KAZALO SLIK**

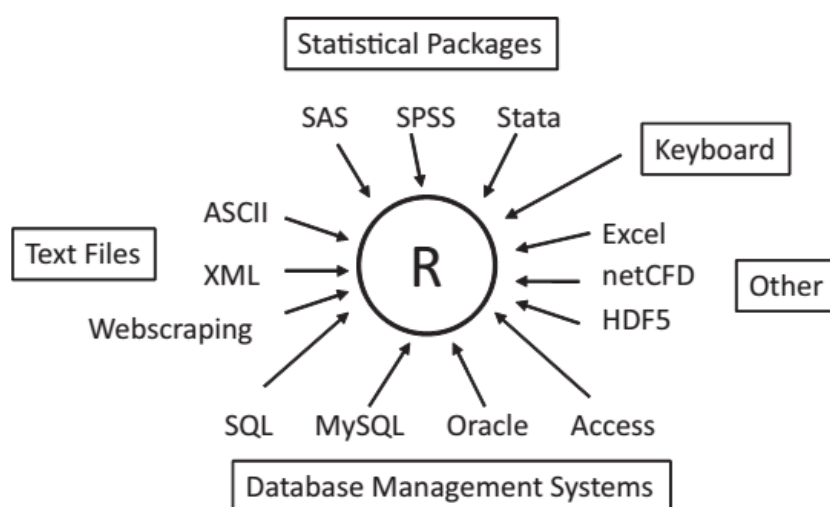
Slika 1 : Podatkovne strukture, ki jih lahko uvozimo v R .....	1
Slika 2: Urejevalnik podatkov (angl. Data editor) .....	2
Slika 3: Urejevalnik podatkov z nekaj vnesenimi podatki .....	2

# 1 USTVARJANJE PODATKOV

Kot smo že spoznali, v R lahko uporabljamo različne strukture podatkov, ki smo jih imenovali objekti (vektorji, matrike, sezname, podatkovni okviri). Podatke, ki jih zapisujemo v različnih objektih, navadno pridobimo iz različnih virov in v različnih formatih. Naša naloga pa je, da znamo te podatke uvoziti v orodje za njihovo analizo, jih analizirati in poročati o rezultatih analize.

R vključuje različna orodja za uvažanje podatkov<sup>1</sup>. Kot prikazuje Slika 1, lahko v R vnašamo podatke preko tipkovnice, lahko pa jih tudi uvažamo iz tekstovnih datotek, sistemov za upravljanje z bazami podatkov, drugih statističnih paketov in drugih.

Slika 1 : Podatkovne strukture, ki jih lahko uvozimo v R



Vir: Kabacoff, R. 2011. *R in Action*. Slika 2.1, str. 34.

## 1.1 VNOS PODATKOV PREKO TIPKOVNICE

Najenostavnejša metoda vnašanja podatkov je uporaba tipkovnice. To storimo v dveh korakih:

1. Izdelamo prazen podatkovni okvir (ali matriko) z imeni spremenljivk in njihovimi tipi.
2. Odpremo urejevalnik besedila na tem podatkovnem objektu, vnesemo podatke ter shranimo rezultat v podatkovni objekt.

Primer: Ustvarimo podatkovni okvir `mojipodatki` s tremi spremenljivkami:

`starost` (številka), `spol` (opisna) in `teža` (številka):

```
mojipodatki<-
data.frame(starost=numeric(0), spol=character(0), teza=numeric(0))
```

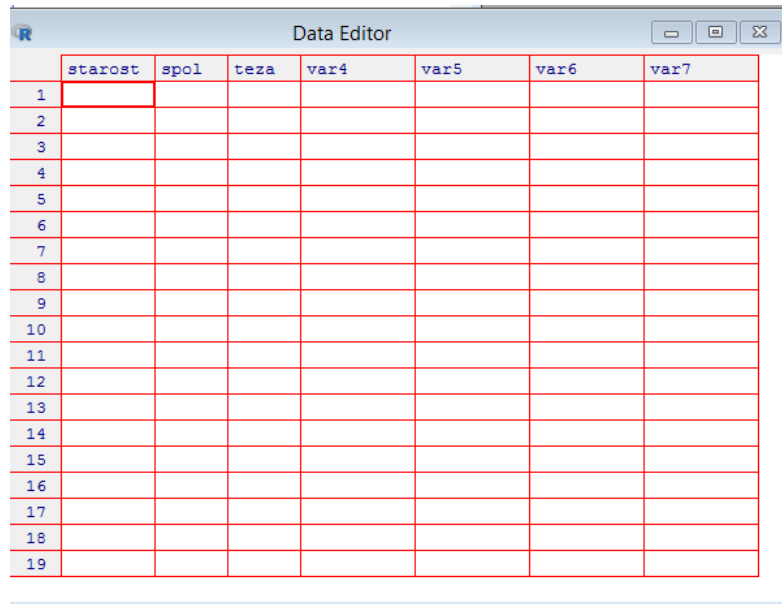
<sup>1</sup> Glej npr. <https://cran.r-project.org/doc/manuals/R-data.pdf>

Za priklic urejevalnika uporabimo funkcijo `edit()` :

```
edit(mojipodatki)
```

Odpre se okno, ki ga prikazuje Slika 2:

*Slika 2: Urejevalnik podatkov (angl. Data editor)*

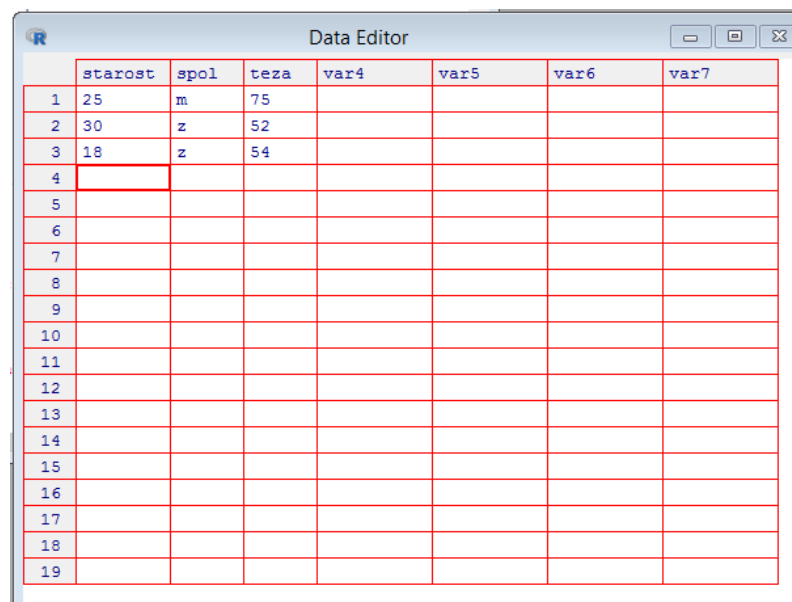


The screenshot shows the R Data Editor window. The title bar says "Data Editor". The table has 8 columns: "starost", "spol", "teza", "var4", "var5", "var6", and "var7". The rows are numbered 1 to 19. The first row (row 1) has a red border around the "starost" cell, which is currently empty.

	starost	spol	teza	var4	var5	var6	var7
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							

V to okno lahko sedaj vpisujemo podatke (Slika 3):

*Slika 3: Urejevalnik podatkov z nekaj vnesenimi podatki*



The screenshot shows the R Data Editor window with the same table as in Slika 2. The first three rows (rows 1, 2, and 3) are filled with data. The first row (row 1) has "25" in the "starost" column, "m" in the "spol" column, and "75" in the "teza" column. The second row (row 2) has "30" in the "starost" column, "z" in the "spol" column, and "52" in the "teza" column. The third row (row 3) has "18" in the "starost" column, "z" in the "spol" column, and "54" in the "teza" column. The fourth row (row 4) has a red border around the "starost" cell, which is currently empty.

	starost	spol	teza	var4	var5	var6	var7
1	25	m	75				
2	30	z	52				
3	18	z	54				
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							

## Pomembno!

Če uporabimo le zgornji ukaz, bomo ob zaprtju urejevalnika podatkov izgubili vse vnesene podatke. Zato je potrebno pred tem funkciji `edit()` prirediti neko ime, v katero se bodo shranili podatki. Npr.:

```
mojipodatki<- edit(mojipodatki)
```

Ta metoda vnosa podatkov je seveda uporabna le v primeru majhnega števila podatkov. Za večje podatkovne tabele pa navadno uporabimo katerega izmed načinov uvažanja podatkov, ki jih opisujemo v nadaljevanju.

Pogosto pa se namreč dogaja, da imamo podatke že shranjene v datoteki, ki je poljubnega formata. V nadaljevanju si bomo ogledali načine, na katere lahko v R uvozimo podatke različnih formatov (pogledali si bomo uvažanje podatkov iz najpogostejših formatov).

## 1.2 UVAŽANJE PODATKOV IZ BESEDILNE DATOTEKE

V besedilnih datotekah so podatki med seboj navadno ločeni s tabulatorji (.txt) ali vejicami (.csv). Takšne podatke lahko uvozimo s pomočjo funkcije `read.table()`. Ta funkcija prebere datoteko v obliki tabele in jo shrani kot podatkovni okvir. Osnovna oblika funkcije je naslednja:

```
mojipodatki<-  
read.table(file,header=logical_value,sep='delimiter',row.names='name')
```

pri tem je: *file* tekstovna datoteka,

za *header* opredelimo logično vrednost (TRUE ali FALSE), s čimer določimo, ali datoteka, ki jo uvažamo, v prvi vrstici vključuje imena spremenljivk,

*sep* določa ločilo, ki ločuje podatkovne vrednosti,

*row.names* pa je opcijski parameter, ki določa eno ali več spremenljivk, ki predstavljajo identifikatorje vrstic

Primer 1: Uvozimo podatke iz besedilne \*.txt datoteke:

```
podatki1<-read.table('POD1.txt',header=TRUE)
```

Rezultat prikličemo s pomočjo imena podatkov `podatki1`. Rezultat je:

	Price	Floor	Area	Rooms	Age	Cent.heat
1	52.00	2	830	5	6.2	no
2	54.75	3	710	5	7.5	no
3	57.50	1	1000	5	4.2	no
4	57.50	3	690	6	8.8	no
5	59.75	1	900	5	1.9	yes

Primer 2: Uvozimo podatke iz besedilne \*.csv datoteke:

```
podatki2<-read.table('POD1.csv',header=TRUE,sep=';')
```

Rezultat prikličemo s pomočjo imena podatkov `podatki2`. Rezultat je:

	Price	Floor	Area	Rooms	Age	Cent.heat
1	52	2	830	5	6,2	no
2	54,75	3	710	5	7,5	no
3	57,5	1	1000	5	4,2	no
4	57,5	3	690	6	8,8	no
5	59,75	1	900	5	1,9	yes

Kot vidimo, se v podatkih kot ločilo pojavlja vejica. Vemo, da R za decimalno ločilo uporablja piko, zato uvoženih podatkov ne bo znal uporabljati. K sreči lahko tudi ta problem odpravimo takoj ob uvažanju podatkov tako, da v funkciji uporabimo tudi argument `dec=`:

```
podatki2<-  
read.table('C:/POD1.csv',header=TRUE,sep=';',dec=",")
```

In rezultat je:

	Price	Floor	Area	Rooms	Age	Cent.heat
1	52.00	2	830	5	6.2	no
2	54.75	3	710	5	7.5	no
3	57.50	1	1000	5	4.2	no
4	57.50	3	690	6	8.8	no
5	59.75	1	900	5	1.9	yes

### 1.3 UVAŽANJE EXCELOVIH DATOTEK

V R lahko uvozimo tudi podatke, ki jih imamo shranjene v Excelovem (.xlsx) formatu. Za to moramo najprej v R naložiti in namestiti paket `xlsx`:

```
install.packages('xlsx')  
library(xlsx)
```

V nameščenem paketu uporabimo funkcijo `read.xlsx()`, ki ima naslednjo osnovno obliko:

```
mojipodatki<-read.xlsx(file,n)
```

Pri tem je: `file` pot do xlsx datoteke

`n` je številka delovnega zvezka, ki ga želimo uvoziti.

Primer: Uvozimo podatke iz Excelove datoteke:

```
library(xlsx)
podatki3<-read.xlsx('POD1.xlsx',1)
```

Rezultat prikličemo s pomočjo imena podatkov `podatki3`. Rezultat je:

	Price	Floor	Area	Rooms	Age	Cent.heat
1	52.00	2	830	5	6.2	no
2	54.75	3	710	5	7.5	no
3	57.50	1	1000	5	4.2	no
4	57.50	3	690	6	8.8	no
5	59.75	1	900	5	1.9	yes

## 1.4 UVAŽANJE SPSS DATOTEK

Nabore podatkov iz SPSS (končnica \*.sav) lahko uvozimo v R s pomočjo:

- funkcije `read.spss()`, ki se nahaja v paketu `foreign` ali
- funkcije `spss.get()`, ki se nahaja v paketu `Hmisc`.

Funkcija `spss.get()` je ohlapnejša, saj avtomatsko namesto nas določi več parametrov, ki jih je potrebno določiti v primeru funkcije `read.spss()`. S tem je transfer podatkov olajšan in tudi bolj konsistenten z vsemi določili, ki so zapisana v \*.sav formatu podatkov.

Pred uporabo moramo tudi naložiti in namestiti izbran paket:

```
install.packages('Hmisc')
library(Hmisc)

ali

install.packages('foreign')
library(foreign)
```

Splošna oblika funkcije `spss.get()` je naslednja:

```
mojipodatki<-spss.get(file,use.value.labels=logical_value)
```

Pri tem je: `file` podatkovna \*.sav datoteka, ki jo želimo uvoziti;

opcija `use.value.labels=logical_value` pove, da naj funkcija pretvori oznake vrednosti spremenljivk v R faktorje (**Opomba:** v našem primeru te opcije ne uporabimo, torej opredelimo `FALSE`!);

Primer: Uvozimo podatke iz SPSS datoteke:

```
library(Hmisc)
podatki4<-spss.get('POD1.sav',use.value.labels=TRUE)
```

Rezultat prikličemo s pomočjo imena podatkov `podatki4`. Rezultat je:

	Price	Floor	Area	Rooms	Age	Cent.heat
1	52.00	2	860	5	6.2	no
2	54.75	3	710	5	7.5	no
3	57.50	1	1000	5	4.2	no
4	57.50	3	690	6	8.8	no
5	59.75	1	900	5	1.9	yes

Uporabimo še pri argumentu `use.value.labels` logično vrednost `FALSE`:

```
podatki4<-spss.get('POD1.sav',use.value.labels=FALSE)
```

Rezultat je:

	Price	Floor	Area	Rooms	Age	Cent.heat
1	52.00	2	860	5	6.2	1
2	54.75	3	710	5	7.5	1
3	57.50	1	1000	5	4.2	1
4	57.50	3	690	6	8.8	1
5	59.75	1	900	5	1.9	2

## 2 OSNOVNO UPRAVLJANJE S PODATKI V R

Ko podatke uvozimo v R, moramo iz njih ustvariti novo podatkovno datoteko, ki bo zajemala spremenljivke, ki smo jih izbrali za analiziranje. V nadaljevanju so opisani določeni postopki v R, ki nam pomagajo pri urejanju podatkov.

### 2.1 IMENA STOLPCEV

Imena stolpcev zajemajo imena posameznih spremenljivk. Do imen stolpcev dostopamo s pomočjo funkcije `names()`.

```
names(podatki)
```

Ker bomo pri obdelavi podatkov in izvajanju analiz večkrat uporabljali imena stolpcev (spremenljivk), je priporočeno, da so ta imena kratka ter vsebinsko povezana s spremenljivko.

Imena lahko zamenjamo na naslednje načine:

- zamenjamo le eno ime glede na številko stolpca:

Npr. ime spremenljivke `gndr`, ki se nahaja v 2. stolpcu, spremenimo v ime `spol`:

```
names(podatki)[2]='spol'
```

- zamenjamo več imen hkrati:

Npr. imena spremenljivk `cndry`, `gndr`, `ageg` in `yrbrn` ki se nahajajo v 1, 2., 3. in 5. stolpcu, spremenimo v imena `drz`, `spol`, `star`, `letroj`:

```
names(podatki)[c(1:3,5)]=c('drz','spol','star','letroj')
```

- zamenjamo vsa imena hkrati:

```
names(podatki)=c('ime1','ime2','ime3','ime4','ime5')
```

### 2.2 PREUREJANJE VRSTNEGA REDA STOLPCEV

Včasih želimo iz takšnega ali drugačnega razloga stolpce prerazporediti. To lahko storimo na naslednja načina:

- vrstni red stolpcev preuredimo glede na zaporedno številko:

```
podatki.1=podatki[c(2,6:8,1,3:5)]
```

```
names(podatki.1)
```

- vrstni red stolpcev preuredimo glede na imena stolpcev:

```
podatki.2=podatki.1[c('spol','letroj','drz')]
```

```
names(podatki.2)
```



## 2.3 IZLOČANJE DOLOČENEGA DELA PODATKOV

Navadno moramo iz večjega podatkovnega okvira izbrati le določene spremenljivke in ustvariti nov podatkovni okvir.

### 2.3.1 Izbiranje (ohranjanje) spremenljivk

Stolpec z izbrano spremenljivko lahko izberemo na dva načina:

- uporabimo lahko zaporedno številko stolpca, v katerem se vrednosti spremenljivke nahajajo:

```
podatki.3=podatki[,2]
```

Zgornji ukaz iz podatkovnega okvira `podatki` izbere spremenljivko, ki se nahaja v 2. stolpcu, in jo shrani v podatkovni okvir `podatki.3`.

- uporabimo lahko znak \$, pri čemer levo od znaka vpišemo ime podatkov, desno od znaka pa ime spremenljivke, ki jo želimo izločiti:

```
podatki.4=podatki$spol
```

Zgornji ukaz iz podatkovnega okvira `podatki` izbere spremenljivko 'spol' in jo shrani v podatkovni okvir `podatki.4`.

Če želimo v podatkovnem okviru izbrati več kot eno spremenljivko, lahko to storimo na naslednje načine:

- uporabimo zaporedne številke stolpcev, v katerih se nahajajo izbrane spremenljivke:

```
podatki.5=podatki[,c(2:6)]
```

Ukaz iz podatkovnega okvira `podatki` izbere spremenljivke, ki se nahajajo v 2., 3., 4., 5. in 6. stolpcu, ter jih shrani v podatkovni okvir `podatki.5`.

- ustvarimo lahko tudi vektor z imeni spremenljivk z ukazom:

```
spr=c('spr1','spr2','spr3','spr4','spr5')
```

S tem v vektor `spr` shranimo imena spremenljivk, ki jih želimo izbrati.

Te spremenljivke iz podatkovnega okvira nato izberemo z ukazom:

```
podatki.6=podatki[spr]
```

Kombiniramo lahko tudi znaka \$ in []: z znakom \$ izločimo vektor (stolpec, spremenljivko) in tega indeksiramo z uporabo []:

```
podatki.7=podatki$spol[50:55]
```

Ukaz nam torej izpiše vrednosti izbrane spremenljivke, ki se nahajajo od 50. do 55. zaporednega mesta (torej za 50., 51., 52., 53., 54. in 55. zaporedno enoto), in jih shrani v nov podatkovni okvir `podatki.6`.

### 2.3.2 Izključevanje (opuščanje) spremenljivk

Če želimo iz podatkov odstraniti posamezen stolpec (spremenljivko), lahko to zopet naredimo na različne načine:

- ustvarimo lahko vektor z imeni spremenljivk, ki jih želimo izbrisati iz podatkovnega okvira, z ukazom:

```
spr=names(podatki) %in% c('spr1','spr2')
```

S tem v vektor `spr` shranimo imena spremenljivk, ki jih želimo odstraniti.

Te spremenljivke iz podatkovnega okvira nato odstranimo z ukazom:

```
podatki.7=podatki[!spr]
```

- uporabimo zaporedne številke stolpcev, v katerih se nahajajo spremenljivke, ki jih želimo odstraniti, in znak – (minus):

```
podatki.8=podatki[c(-2,-6)]
```

Ukaz torej iz podatkovnega okvira `podatki` odstrani spremenljivki, ki se nahajata v 2. in 6. stolpcu. Podatke brez izločenih spremenljivk pa shrani v podatkovni okvir `podatki.8`.

- uporabimo lahko tudi znak \$ ter oznako NULL:

```
podatki$spr1=NULL
```

S tem ukazom določimo stolpcu z imenom 'spr1' vrednost `NULL` (nedefinirano).

### 2.3.3 Izbiranje določenega dela opazovanj

Izbira ali izključitev opazovanj (torej, vrstic) je navadno ključni vidik uspešne priprave in analize podatkov.

Opazovanja (vrstice) lahko v podatkovnem okviru izberemo na različne načine:

- določimo lahko razpon vrstic, ki jih želimo izbrati:

```
podatki.9=podatki[1:36,]
```

S tem ukazom izberemo podatke za spremenljivke, ki se nahajajo v vrsticah od 1 do 36 (torej, izberemo prvih 36 opazovanj) in jih shranimo v nov podatkovni okvir `podatki.9`.

- Opazovanja lahko iz podatkovnega okvira izločimo tudi glede na vrednosti spremenljivk, ki jih enote zavzemajo na spremenljivkah:

```
podatki.10=podatki[which(podatki$spol=='M' & podatki$star>30),]
```

Ukaz nam iz podatkovnega okvira `podatki` izbere tista opazovanja, ki imajo na spremenljivki 'spol' vrednost 'M' (torej, osebe moškega spola) in na spremenljivki 'star' vrednost, ki je večja od 30 (torej, starejši od 30 let) ter jih shrani v nov podatkovni okvir `podatki.10`. V novem podatkovnem okviru imamo torej podatke za vse spremenljivke za moške, starejše od 30 let.

- Uporabimo lahko tudi funkciji `attach()` in `detach()`, s čimer nam ni potrebno uporabljati znaka `$`:

```
attach(podatki)
```

```
podatki.11=podatki[which(spol=='M' & star>30),]
```

```
detach(podatki)
```

Ukaz naredi enako kot prejšnji, le zapis je drugačen. V podatkovnem okviru `podatki.11` imamo torej podatke za vse spremenljivke za moške, starejše od 30 let.

### 2.3.4 Hkratna izbira spremenljivk in opazovanj

Tu gre v bistvu za bližnjico pri izbiranju spremenljivk in/ali opazovanj iz podatkovnega okvira. To nam omogoča funkcija `subset()`, ki je najverjetneje eden od najlažjih načinov za izbiranje spremenljivk in opazovanj v podatkih.

Sledita dva primera uporabe funkcije `subset()`:

#### 1. primer:

```
podatki.12=subset(podatki, star>=35|star<24, select=c(spr1, spr2, spr3))
```

V tem primeru smo iz podatkovnega okvira `podatki` izbrali tiste vrstice, ki imajo na spremenljivki 'star' vrednost večjo ali enako 35 **ali** vrednost manjšo od 24. Tako smo v nov podatkovni okvir `podatki.12` shranili podatke za enote, ki so stari 35 in več ali pa so mlajši od 24 let, in spremenljivke `spr1`, `spr2` in `spr3`.

#### 2. primer:

```
podatki.13=subset(podatki, spol=='M'&star>25, select=spol:spr3)
```

V tem primeru smo iz podatkovnega okvira `podatki` izbrali tiste vrstice, ki imajo na spremenljivki `spol` vrednost 'M' **in** na spremenljivki 'star' vrednost večjo od 25. Tako smo v nov podatkovni okvir `podatki.13` shranili podatke za moške, starejše od 25 let, in spremenljivke `spol`, `spr1`, `spr2` in `spr3`.

## 2.4 SPAJANJE PODATKOVNIH OKVIRJEV

Kadar imamo podatke shranjene na večih lokacijah (v različnih dokumentih), jih moramo spojit, združiti predno pričnemo z njihovo obdelavo. To lahko storimo tako, da spajamo (dodajamo) stolpce ali vrstice.

### 2.4.1 Dodajanje stolpcev (horizontalno spajanje)

Horizontalno spajanje je navadno uporabljeno za dodajanje spremenljivk v obstoječi podatkovni okvir.

Za horizontalno spajanje dveh podatkovnih okvirjev uporabimo funkcijo `merge()`. V večini primerov sta dva podatkovna okvira združena na osnovi ene ali več skupnih spremenljivk (npr. ID):

```
podatki.3=merge(podatkiA,podatkiB,by='ID')
```

Ukaz torej združi podatkovni okvir `podatkiA` in podatkovni okvir `podatkiB` na osnovi spremenljivke 'ID'.

Določimo lahko tudi več kot eno skupno spremenljivko, na osnovi katere spajamo podatke:

```
podatki.3=merge(podatkiA,podatkiB,by=c('ID','Country'))
```

Kadar spajamo matrike podatkov ali podatkovne okvire horizontalno, pri tem pa nam ni potrebno določiti skupnega ključa (skupne spremenljivke, na osnovi katere podatke spajamo), lahko uporabimo tudi funkcijo `cbind()`:

```
podatki.4=cbind(podatkiA,podatkiB)
```

Ta funkcija nam horizontalno združi objekta A in B. Da bo funkcija delovala pravilno, morata imeti oba objekta enako število vrstic ter biti urejena v enakem vrstnem redu.

### 2.4.2 Dodajanje vrstic (vertikalno spajanje)

Vertikalno spajanje je navadno uporabljeno pri dodajanju opazovanj v podatkovni okvir.

Za vertikalno spajanje dveh podatkovnih okvirjev uporabimo funkcijo `rbind()`:

```
podatki.5=rbind(podatkiA,podatkiB)
```

Podatkovna okvira, ki ju spajamo s to funkcijo, morata vsebovati iste spremenljivke, ni pa potrebno, da sta urejena v enakem vrstnem redu.

Če podatkovni okvir A vključuje spremenljivke, ki jih v podatkovnem okviru B ni, potem je pred spajanjem potrebno:

- izbrisati dodatne spremenljivke v podatkovnem okviru A

ali

- ustvariti dodatne spremenljivke v podatkovnem okviru B in njihove vrednosti nastaviti na NA (manjkajoče vrednosti).

## 2.5 UPORABNE FUNKCIJE ZA DELO S PODATKOVNIMI OBJEKTI

Funkcija	Namen
<code>length(object)</code>	število elementov / komponent
<code>dim(object)</code>	dimenzija (razsežnost) objekta
<code>str(object)</code>	struktura objekta
<code>class(object)</code>	razred oziroma tip objekta
<code>mode(object)</code>	na kakšen način je objekt shranjen
<code>names(object)</code>	imena komponent objekta
<code>c(object, object, ...)</code>	združi objekte v vektor
<code>cbind(object, object, ...)</code>	združi objekte po stolpcih
<code>rbind(object, object, ...)</code>	združi objekte po vrsticah
<code>object</code>	izpiše objekt
<code>head(object)</code>	prikaže prvi del objekta
<code>tail(object)</code>	prikaže zadnji del objekta
<code>ls()</code>	izpiše trenutne objekte, ki so na voljo
<code>rm(object, object, ...)</code>	izbriše enega ali več objektov
<code>newobject&lt;-edit(object)</code>	uredi objekt in ga shrani kot newobject
<code>fix(object)</code>	uredi na mestu

## 3 DODAJANJE PRIPISOV PODATKOM

Navadno so nam pri analiziranju podatkov in interpretiranju rezultatov v precejšnjo pomoč imena, ki so pripisana tako spremenljivkam, kot tudi njihovim vrednostim. Navadno se pripisi

torej nanašajo na dodajanje opisnih oznak imenom spremenljivk in opisnih oznak vrednostim kategoričnih (opisnih) spremenljivk.

### 3.1 OZNAKE SPREMENLJIVK

Sposobnost R-ja za ravnanje z oznakami spremenljivk je precej omejena. Pomagamo si lahko le tako, da uporabimo opisno oznako spremenljivke kot ime spremenljivke. V funkcijah, ko uporabljamo spremenljivko, pa uporabljamo kar indeks njenega položaja v podatkovni tabeli (torej, št. stolpca).

Če se spomnimo primera, ko smo obravnavali način, na katerega podatke v R vnašamo preko tipkovnice, denimo, da želimo za spremenljivko `starost` uporabiti opisno oznako »Starost ob hospitalizaciji (v letih)«.

To lahko naredimo s pomočjo funkcije `names()` na naslednji način:

```
names(mojipodatki)[1]<-'Starost ob hospitalizaciji (v letih)'
```

Rezultat je:

	Starost ob hospitalizaciji (v letih)	spol	teza
1	25	m	75
2	30	z	52
3	18	z	15

Seveda je to novo ime predolgo, da bi ga uporabljali vsakič, ko kličemo to spremenljivko v različnih funkcijah. Zato se v teh primerih raje sklicujemo na to spremenljivko kot:

```
mojipodatki[1],
```

opisna oznaka `Starost ob hospitalizaciji (v letih)` pa se bo izpisala v vsakem izpisu, v katerem se pojavlja ta spremenljivka.

### 3.2 OZNAKE VREDNOSTI SPREMENLJIVK

Če želimo opisnim oz. kategoričnim spremenljivkam pripisati opisne oznake vrednosti, lahko uporabimo funkcijo `factor()`.

V primeru spremenljivke `spol` želimo vrednosti `m` pripisati oznako 'moški', vrednosti `z` pa oznako 'ženska'. Opisne oznake vrednosti lahko pripravimo na naslednji način:

```
mojipodatki$spol<-  
factor(mojipodatki$spol, levels=c('m','z'), labels=c('moski','zenska'))
```

Rezultat je:

	starost	spol	teza
1	25	moski	75
2	30	zenska	52
3	18	zenska	15