

Базе података у R-у

2. час

8. март 2016.

Базе података

То је листа сачињена од вектора и/или фактора једнаке дужине који су у таквој међусобној вези да подаци на истој позицији у сваком од њих потичу од исте експерименталне јединице.

Може се замислити и као матрица чије су врсте случајеви (опсервације), а колоне променљиве. Базе података могу се лако конструисати од (постојећих или нових) вектора коришћењем функције `data.frame()`.

```
> colors <- c("red","yellow","blue")
> numbers <- c(1, 2, 3)
> col.and.num<- data.frame(colors, numbers, more.numbers=4:6)
> col.and.num
```

```
  colors numbers more.numbers
1 red         1             4
2 yellow      2             5
3 blue        3             6
```

“Прилепљивање” нових колона, односно врста, на постојећу базу:

```
> exact <- c(T, F, F)
> col.and.num1 <- cbind(col.and.num, exact)
> col.and.num1
```

```
  colors numbers more.numbers  exact
1 red         1             4   TRUE
2 yellow      2             5  FALSE
3 blue        3             6  FALSE
```

Коришћење база података, које се налазе у оквиру R пакета:

```
> data(trees)
> d <- trees
> head(d)
```

	Girth	Height	Volume
1	8.3	70	10.3
2	8.6	65	10.3
3	8.8	63	10.2
4	10.5	72	16.4
5	10.7	81	18.8
6	10.8	83	19.7

```
> d$Height
[1] 70 65 63 72 81 83 66 75 80 75 79 76 76 69 75 74 85 86 71 64 78 80 74 72 77 81 82 80 80 80 87
> d$Height[1:10]
[1] 70 65 63 72 81 83 66 75 80 75
> d$Height[1]
[1] 70
```

Променљиве у бази података могу се учинити видљивим на глобалном нивоу у R изразима функцијом `attach()`. Овим се избегава гломазна `$`-нотација.

```
> attach(d)
> HG <- Height/Girth
> HG
```

```
[1] 8.433735 7.558140 7.159091 6.857143 7.570093 7.685185 6.000000 6.818182
[9] 7.207207 6.696429 6.991150 6.666667 6.666667 5.897436 6.250000 5.736434
[17] 6.589147 6.466165 5.182482 4.637681 5.571429 5.633803 5.103448 4.500000
[25] 4.723926 4.682081 4.685714 4.469274 4.444444 4.444444 4.223301
```

Функција `str()` враћа структуру прослеђеног објекта.

```
> str(HG)
num [1:31] 8.43 7.56 7.16 6.86 7.57 ...
> str(trees)
'data.frame': 31 obs. of 3 variables:
 $ Girth : num 8.3 8.6 8.8 10.5 10.7 10.8 11 11 11.1 11.2 ...
 $ Height: num 70 65 63 72 81 83 66 75 80 75 ...
 $ Volume: num 10.3 10.3 10.2 16.4 18.8 19.7 15.6 18.2 22.6 19.9 ...
```

Функције `subset()` и `transform()` користе се за издвајање подскупова података и креирање нових база са трансформисаним променљивим:

```
> (d.small <- subset(d, Volume < 18))
```

	Girth	Height	Volume
1	8.3	70	10.3
2	8.6	65	10.3
3	8.8	63	10.2
4	10.5	72	16.4
7	11.0	66	15.6

```
> (d.in_m3 <- transform(d, Volume.m=0.0283168466*Volume))
```

	Girth	Height	Volume	Volume.m
1	8.3	70	10.3	0.2916635
2	8.6	65	10.3	0.2916635
3	8.8	63	10.2	0.2888318
4	10.5	72	16.4	0.4643963
.	.	.	.	.
29	18.0	80	51.5	1.4583176
30	18.0	80	51.0	1.4441592
31	20.6	87	77.0	2.1803972

У ситуацији када су подаци за различите групе субјекта организовани у више база, могуће је спојити базе помоћу функције `merge()`. Базе се спајају по променљивој истог имена у обе базе. Та променљива је обично *ID*, променљива која чува идентификацију субјекта. Није препоручљиво коришћење функције `cbind()`.

```
> data1 <- data.frame(ID=1:5, x=letters[1:5])
> data2 <- data.frame(ID=1:5, y=letters[6:10])
> merge(data1, data2)
```

	ID	x	y
1	1	a	f
2	2	b	g
3	3	c	h
4	4	d	i
5	5	e	j

У случају када постоји више променљивих истог имена (у обе базе), потребно је додати аргумент `'by='`.

```
> merge(data1, data2, by='ID', suffixes=c(1, 2))
```

	ID	x1	x2
1	1	a	f
2	2	b	g
3	3	c	h
4	4	d	i
5	5	e	j

Логичке и *NA* вредности у вектору

Хоћемо да израчунамо средњу вредност вектора *x*.

```
> (x <- c(10, 20, NA, 4, NA, 2))  
[1] 10 20 NA 4 NA 2  
> sum(x) / length(x)  
[1] NA  
> (i <- is.na(x))  
[1] FALSE FALSE TRUE FALSE TRUE FALSE  
> !i  
[1] TRUE TRUE FALSE TRUE FALSE TRUE  
> (y <- x[!i])  
[1] 10 20 4 2  
> sum(x[!i]) / length(x[!i])  
[1] 9
```

Исти резултат добијамо на следећи начин

```
> mean(x, na.rm = TRUE)  
[1] 9
```

Учитавање података

Учитавање табеле као базе података врши се помоћу функције `read.table("putanja/ime_fajla.txt",header=T)`.

Аргумент `header=T` означава да прву врсту не посматра као податке већ као `header`. На пример, прво правимо текстуални фајл `cars.txt` који сместимо у исти фолдер или у позиву функције `read.table` у имену фајла наводимо целу путању.

```
baza<-read.table('cars.txt',header=T)
```

Ова функција подразумева да је сепаратор празно поље. Могуће је да сепаратор буде и

- зарез, тада се користи функција `read.csv()`
- ТАВ, тада се користи функција `read.delim()`

Уколико се подаци налазе у *excel*-у, могуће их је учитати помоћу:

```
baza <- read.xls('putanja/ime_fajla.tx', header=T)
```


file	the name of the file (within "" or a variable of mode character), possibly with its path (the symbol \ is not allowed and must be replaced by /, even under Windows), or a remote access to a file of type URL (http://...)
header	a logical (FALSE or TRUE) indicating if the file contains the names of the variables on its first line
sep	the field separator used in the file, for instance <code>sep="\t"</code> if it is a tabulation
quote	the characters used to cite the variables of mode character
dec	the character used for the decimal point
row.names	a vector with the names of the lines which can be either a vector of mode character, or the number (or the name) of a variable of the file (by default: 1, 2, 3, ...)
col.names	a vector with the names of the variables (by default: V1, V2, V3, ...)
as.is	controls the conversion of character variables as factors (if FALSE) or keeps them as characters (TRUE); as.is can be a logical, numeric or character vector specifying the variables to be kept as character
na.strings	the value given to missing data (converted as NA)
colClasses	a vector of mode character giving the classes to attribute to the columns
nrows	the maximum number of lines to read (negative values are ignored)
skip	the number of lines to be skipped before reading the data
check.names	if TRUE, checks that the variable names are valid for R
fill	if TRUE and all lines do not have the same number of variables, "blanks" are added
strip.white	(conditional to sep) if TRUE, deletes extra spaces before and after the character variables
blank.lines.skip	if TRUE, ignores "blank" lines
comment.char	a character defining comments in the data file, the rest of the line after this character is ignored (to disable this argument, use <code>comment.char = ""</code>)

Пример

```
> baza <- read.csv('D:/SOFTVER_4/2.BAZE_POATAKA/baza.txt', sep='')  
> View(baza)
```

	Field.Name	Area	Slope	Vegetation	Soil.pH	Damp	Worm.density
1	Nashs.Field	3.6	11	Grassland	4.1	FALSE	4
2	Silwood.Bottom	5.1	2	Arable	5.2	FALSE	7
3	Nursery.Field	2.8	3	Grassland	4.3	FALSE	2
4	Rush.Meadow	2.4	5	Meadow	4.9	TRUE	5
5	Gunness.Thicket	3.8	0	Scrub	4.2	FALSE	6
6	Oak.Mead	3.1	2	Grassland	3.9	FALSE	2
7	Church.Field	3.5	3	Grassland	4.2	FALSE	3
8	Ashurst	2.1	0	Arable	4.8	FALSE	4
9	The.Orchard	1.9	0	Orchard	5.7	FALSE	9
10	Rookery.Slope	1.5	4	Grassland	5.0	TRUE	7
11	Garden.Wood	2.9	10	Scrub	5.2	FALSE	8
12	North.Gravel	3.3	1	Grassland	4.1	FALSE	1
13	South.Gravel	3.7	2	Grassland	4.0	FALSE	2

```
> attach(baza)  
> names(baza)  
> summary(baza)
```

Пример

> baza

	Field.Name	Area	Slope	Vegetation	Soil.pH	Damp	worm.density
1	Nashs.Field	3.6	11	Grassland	4.1	FALSE	4
2	Silwood.Bottom	5.1	2	Arable	5.2	FALSE	7
3	Nursery.Field	2.8	3	Grassland	4.3	FALSE	2
4	Rush.Meadow	2.4	5	Meadow	4.9	TRUE	5
5	Gunness.Thicket	3.8	0	Scrub	4.2	FALSE	6
6	Oak.Mead	3.1	2	Grassland	3.9	FALSE	2
7	Church.Field	3.5	3	Grassland	4.2	FALSE	3
8	Ashurst	2.1	0	Arable	4.8	FALSE	4
9	The.Orchard	1.9	0	Orchard	5.7	FALSE	9
10	Rookery.Slope	1.5	4	Grassland	5.0	TRUE	7
11	Garden.Wood	2.9	10	Scrub	5.2	FALSE	8
12	North.Gravel	3.3	1	Grassland	4.1	FALSE	1
13	South.Gravel	3.7	2	Grassland	4.0	FALSE	2
14	observatory.Ridge	1.8	6	Grassland	3.8	FALSE	0
15	Pond.Field	4.1	0	Meadow	5.0	TRUE	6
16	Water.Meadow	3.9	0	Meadow	4.9	TRUE	8
17	Cheapside	2.2	8	Scrub	4.7	TRUE	4
18	Pound.Hill	4.4	2	Arable	4.5	FALSE	5
19	Gravel.Pit	2.9	1	Grassland	3.5	FALSE	1
20	Farm.Wood	0.8	10	Scrub	5.1	TRUE	3

Пример - решение

```
> baza[3,5]
[1] 4.3
> baza[14:19,7]
[1] 0 6 8 4 5 1
> baza[1:5,2:3]
  Area Slope
1  3.6    11
2  5.1     2
3  2.8     3
4  2.4     5
5  3.8     0
> baza[3,]
  Field.Name Area Slope Vegetation Soil.pH Damp Worm.density
3 Nursery.Field  2.8    3  Grassland   4.3 FALSE              2
> baza[,3]
[1] 11  2  3  5  0  2  3  0  0  4 10  1  2  6  0  0  8  2  1 10
> baza[,c(1,5)]
  Field.Name Soil.pH
1  Nashs.Field   4.1
2  Silwood.Bottom 5.2
. . .
20 Farm.Wood    5.1
> baza[sample(1:20,8),]
```

Сортирање базе података

За сортирање вектора најједноставније је користити функцију `sort()`. За сортирање базе података користи се функција `order()`, именовањем једног или више поља на основу којих би требало извршити сортирање. Сортирање на основу већег броја критеријума врши се прослеђивањем вишеструких аргумената функцији `order()`. Друга променљива се користи када се о уређењу не може одлучити на основу прве променљиве.

```
> d[order(d$Height, d$Girth),]
```

	Girth	Height	Volume
3	8.8	63	10.2
20	13.8	64	24.9
...			
16	12.9	74	22.2
23	14.5	74	36.3
8	11.0	75	18.2
10	11.2	75	19.9
...			
18	13.3	86	27.4
31	20.6	87	77.0

Сортирање се по default-у врши у растућем поретку. Поредак сортирања може се променити, тј. може се вршити сортирање у опадајућем поретку, додавањем знака минус испред променљиве.

Пример - наставак

```
> baza[order(Slope),]  
> baza[rev(order(Slope)),]  
> baza[order(Vegetation,Worm.density),]  
> baza[order(Vegetation,Worm.density,Soil.pH),]  
> baza[order(Vegetation,Worm.density),c(4,7,5,3)]  
> baza[order(Vegetation,Worm.density),  
  c('Vegetation', 'Worm.density', 'Soil.pH', 'Slope')]
```

Употреба логичких оператора за издвајање делова базе.

```
> baza[Damp == T,]  
> baza[Worm.density > median(Worm.density) & Soil.pH < 5.2,]  
> baza[-(6:15),]  
> baza[!(Vegetation=='Grassland'),]
```

Табеле

Код експерименталних података често се појављују категоричке променљиве, које указују на неку потподелу података у групе. Њих у R-у означавамо као факторе (factors). Исправност многих анализа у R-у зависи од коректног препознавања фактора, нпр. функција `table()` ради само са категоричким променљивим. Фактори се дефинишу функцијом `factor()`, односно `ordered()`. Могу се добити и конверзијом знаковних или нумеричких вектора функцијама за промену типа: `as.factor()` и `as.ordered()`. Нивои постојећег фактора издвајају се функцијом `levels()`.

```
> table(trees$Height)
```

```
63 64 65 66 69 70 71 72 74 75 76 77 78 79 80 81 82 83 85 86 87
 1  1  1  1  1  1  1  2  2  3  2  1  1  1  5  2  1  1  1  1  1
```

Иако сличне, табеле нису у потпуности еквивалентне матрицама. Постоји класа објеката "table" за коју постоје специфични методи.

Графички приказ табела

Графичке могућности су важна и изузетно разноврсна компонента R-овог окружења. Обезбеђене су функције за генерисање различитих статистичких графика од података, као и флексибилно окружење за модификовање детаља на графицима. Могу се “градити” и потпуно нове врсте графика, прилагођене конкретном проблему. Брз увид у обиље R-ових графика стиче се покретањем програма за демонстрацију:

```
> demo(graphics)
> demo(image)
> demo(lattice)
```

Наредбе за креирање графова деле се у три основне групе:

- функције за графике високог нивоа креирају нови графикон, могуће са осама, ознакама, насловима итд.
- функције за графике ниског нивоа додају више информација на постојећи графикон, нпр. додатне тачке, линије и ознаке
- функције за интерактивне графике омогућавају да се интерактивно дода информација на постојећи графикон или из њега “извуче” информација, нпр. помоћу компјутерског миша

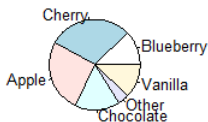
Питице (the pie chart)

```
> pie.sales
```

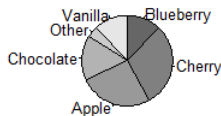
Blueberry	Cherry	Apple	Chocolate	Other	Vanilla
0.12	0.30	0.26	0.16	0.04	0.12

```
> pie(pie.sales, main = "Obicna pitica")  
> pie(pie.sales, col = gray(seq(0.4, 0.9, length = 6)), clockwise = TRUE, main = "Nijanse sive")  
> pie(pie.sales, col = rainbow(6), clockwise = TRUE, main = "Boje duge")  
> library(plotrix)  
> pie3D(pie.sales, explode = 0.1, main = "3D pitica", labelcex = 0.6)
```

Obicna pitica



Nijanse sive



Boje duge



3D pitica

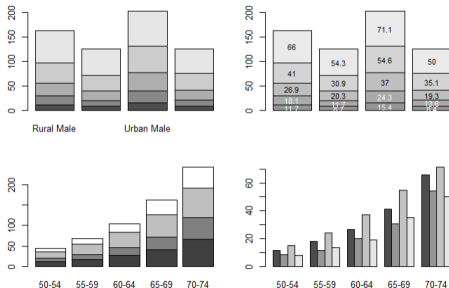


Барови (the bar chart)

```
> VADeaths
```

	Rural Male	Rural Female	Urban Male	Urban Female
50-54	11.7	8.7	15.4	8.4
55-59	18.1	11.7	24.3	13.6
60-64	26.9	20.3	37.0	19.3
65-69	41.0	30.9	54.6	35.1
70-74	66.0	54.3	71.1	50.0

```
> barplot(VADeaths)
> barplot(VADeaths, col=gray(0.5 + 1:5/12), names=rep("", 4))
> text(rep(midpts, each=5), apply(VADeaths, 2, cumsum) - VADeaths/2, VADeaths, col=rep(c("white", "black"),
+      times=2:3), cex=0.8)
> barplot(t(VADeaths), col = gray(c(0.25,0.5,0.75, 1)))
> barplot(t(VADeaths), beside = TRUE)
```



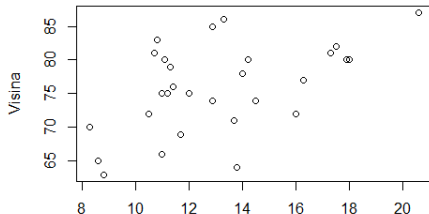
Функција `plot()`

Једна је од R-ових најчешће коришћених функција за графике; спада међу генеричке функције, тј. врста графика који креира дијаграм у зависности од врсте/класе првог аргумента.

- **`plot(x, y)`, `plot(xy)`**

Ако су x и y вектори, `plot(x, y)` за резултат има тачкасти график (`scatterplot`) y -а у зависности од x -а. Исти учинак може се произвести прослеђивањем само једног аргумента xy функцији `plot()`, који је, у том случају, листа са два вектора или матрица са две колоне.

> `plot(cbind(trees$Girth, trees$Height), xlab="Obim", ylab="Visina")`

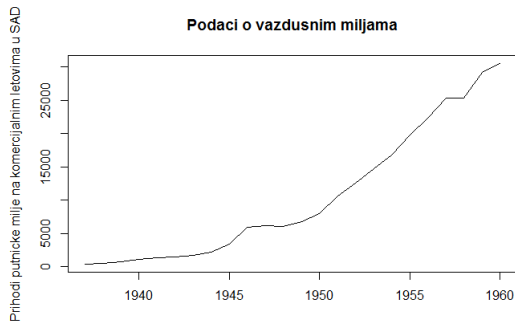


Функција plot()

- **plot(x)**

Ако је x временска серија (time series), резултат је граф временске серије; ако је x нумерички вектор на излазу се добија график вредности тог вектора у зависности од индекса у вектору.

```
> plot(airmiles, main="Podaci o vazdusnim miljama", xlab="vreme", ylab="Prihodi putnicke  
milje na komercijalnim letovima u SAD")
```



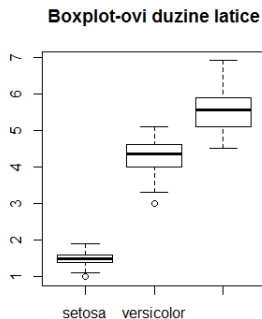
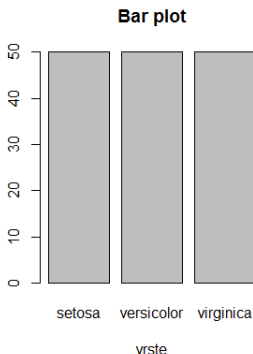
Функција plot()

- **plot(f), plot(f, y)**

Ако је *f* је фактор, *y* је нумерички вектор; првом наредбом генерише се график *f*-а у облику bar-ова, док се као резултат друге наредбе добијају boxplot-ови *y*-а за сваки ниво *f*-а.

> plot(iris\$Species, xlab="vrste", main="Bar plot")

> plot(iris\$Species, iris\$Petal.Length, main="Boxplot-ovi duzine latice")

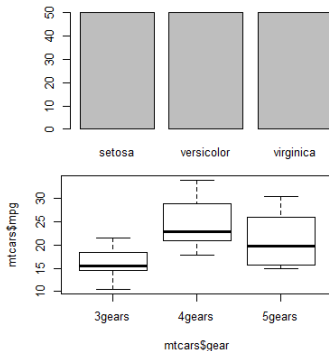
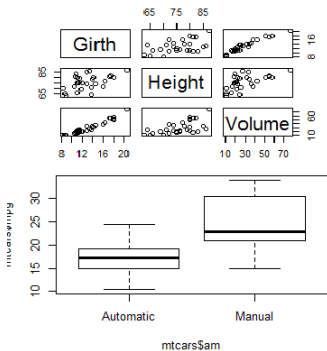


Функција plot()

- **plot(df), plot(~ expr), plot(y ~ expr)**

df је база података, y је било који објекат, expr је списак назива објеката одвојених помоћу '+'; првим двама наредбама, редом, генеришу се distribution plots променљивих у бази, односно distribution plots броја именованих објеката; излаз треће наредбе је низ графика y-а у односу на сваки објекат именован у expr.

```
> plot(trees)
> plot(~ iris$Species)
> plot(mtcars$mpg ~ mtcars$am + mtcars$gear)
```



У R постоје и две веома корисне функције за презентовање вишеструких података.

- **pairs(X)**

где је X нумеричка матрица или база података. Овом наредбом се добија матрица тачкастих графика парова променљивих дефинисаних колонама X -а, тј. свака колона X -а приказана је у зависности од сваке друге колоне X -а.

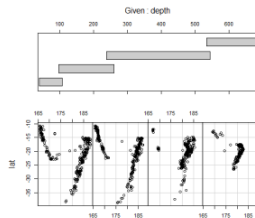
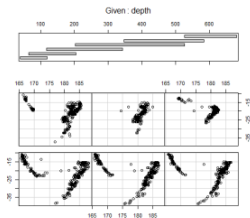
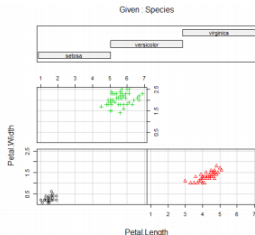
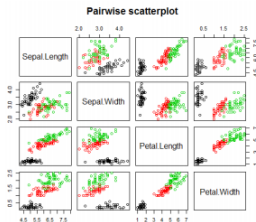
- **coplot(a ~ b | c)**

где су a и b нумерички вектори, а c вектор или фактор (претпоставља се да су сви једнаке дужине). Излаз је низ графика на којима је приказана зависност a од b за дате вредности c -а. Ако је c фактор, то значи да је a приказан у односу на b за сваки ниво c -а. Ако је, пак, c нумерички вектор, он је издељен на неколико “условних” интервала и за сваки интервал a је приказан у односу на b за вредности c -а унутар тог интервала.

```

> pairs(iris[,1:4], col=as.numeric(Species), main="Pairwise scatterplot")
> coplot(Petal.Width ~ Petal.Length | Species, col=as.numeric(Species), pch=as.numeric(Species))
> coplot(lat ~ long | depth, data = quakes)
> given.depth <- co.intervals(quakes$depth, number = 4, overlap = .1)
> coplot(lat ~ long | depth, data = quakes, given.v = given.depth, rows = 1)

```

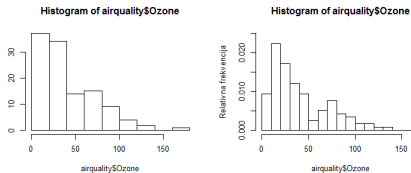


Хистограм

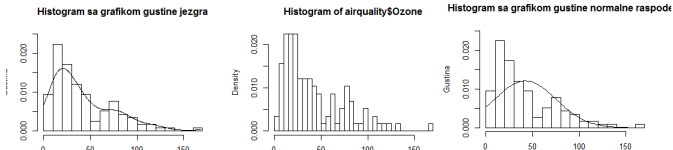
Хистограм је најједноставнија непараметарска “оцена” густине расподеле. Функција **hist(x)**, где је x нумерички вектор је наредба за исцртавање хистограма. То је посебна врста графика са bar-овима, која се користи за приказ расподеле фреквенција низа (непрекидних) нумеричких вредности. Од опционалних аргумената важно је издвојити:

- `nclass`
постављањем на одређену вредност корисник може да препоручи жељени (разуман) број bar-ова
- `breaks` (`breaks= "Sturges"` / `breaks="Scott"` / `breaks="Freedman-Diaconis"`)
могу се тачно задати број bar-ова, односно граничне тачке bar-ова
- `prob=T`
вредности на у-оси представљају релативне, а не апсолутне фреквенције

```
> hist(airquality$Ozone)
> hist(airquality$Ozone, breaks=15, prob=T, ylim=c(0, 0.025), ylab="Relativna frekvencija")
```



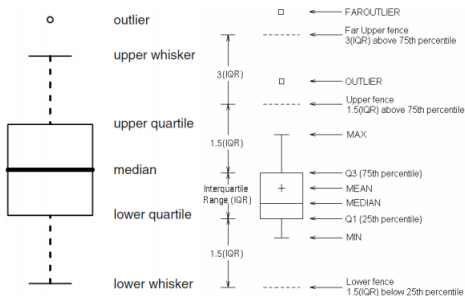
```
> max.ozone <- max(airquality$Ozone, na.rm=T)
> hist(airquality$Ozone, prob=T, breaks=15, xlab='Ozone (ppb)', ylim=c(0, 0.025),
+ ylab='Gustina', main='Histogram sa grafikom gustine jezgra')
> lines(density(airquality$Ozone, na.rm=T, from=0, to=max.ozone))
> mean.ozone <- mean(airquality$Ozone, na.rm=T)
> sd.ozone <- sd(airquality$Ozone, na.rm=T)
> ozone.histogram <- hist(airquality$Ozone, breaks=50, prob=T)
> hist(airquality$Ozone, breaks=15, prob=T, ylim=c(0, 0.025), xlab='Ozone (ppb)',
+ ylab='Gustina', main='Histogram sa grafikom gustine normalne raspodele')
> curve(dnorm(x, mean=mean.ozone, sd=sd.ozone), add = T)
```



Boxplot

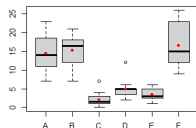
Функција **boxplot(x)**, где је x нумерички вектор је наредба за исцртавање правоугаоног дијаграма. Приликом статистичке анализе требало би га обавезно користити за приказивање (непрекидне) нумеричке променљиве, чије су вредности измерене на подгрупама субјеката, одређеним категоричком променљивом.

Висина правоугаоника (боха) дефинисана је распоном између квантила (IQR). The lower whisker (доња “ограда”) повлачи се са доње стране боха и налази се на нивоу који одговара већој од две вредности: узорачки минимум и први квантил умањен за 1.5IQR . Слично, the upper whisker (горња “ограда”) повлачи се са горње стране боха и налази се на нивоу који одговара мањој од две вредности: узорачки максимум и трећи квантил увећан за 1.5IQR .



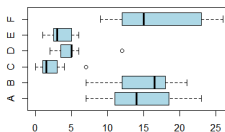
Boxplot

```
> boxplot(count ~ spray, data = InsectSprays, col = "lightgray")  
> means <- tapply(InsectSprays$count, InsectSprays$spray, mean)  
> points(means, col="red", pch=18)
```



Уместо вертикалне оријентације boxplot-ова, навођењем опционалног аргумента `horizontal=T`, добијају се хоризонтално постављени boxplot-ови.

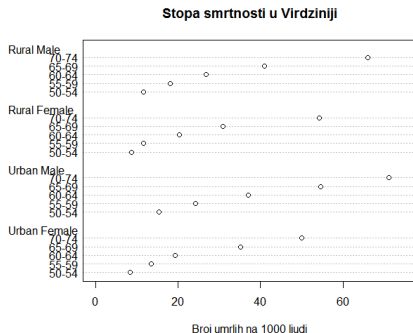
```
> boxplot(count ~ spray, data = InsectSprays, horizontal=T, col = "lightgray")
```



Dotchart

Функција **dotchart()** је функција за генерисање тачкастог графика. Посебно је погодан за визуални приказ свих унесених података с вредностима које леже у одређеним распонима.

```
> dotchart(VADeaths, xlim=c(0,75), xlab="Broj umrlih na 1000 ljudi",  
           main="Stopa smrtnosti u Virdziniji")
```



Q-Q графици

Q-Q графици су врста тачкастих графика, који се користе да би се упоредиле расподеле две групе података или узорачка са (неком) референтном расподелом.

- **qqnorm(x)**
исцртава нумерички вектор x у односу на очекивану нормалну расподелу
- **qqline(x)**
додаје праву на претходни график која пролази кроз квантиле расподеле и података
- **qqplot(x, y)**
исцртава квантиле x -а у односу на квантиле y -а ради поређења њихових припадних расподела

Q-Q графици

```
> X <- rnorm(1000); A <- rnorm(1000); B <- rnorm(1000, mean=3, sd=2)
> C <- rt(1000, df=2); D <- exp(rnorm(1000))
> par(mfrow=c(2, 2), mar=c(3, 3, 2, 1))
> qqplot(X, A, main="A i X su iz iste raspodele")
> qqplot(X, B, main="B je reskaliran X")
> qqplot(X, C, main="C ima tezi rep")
> qqplot(X, D, main="D je pomerena u desno")
```

