

ТЕОРИЈА УЗОРАКА час 2

12. март '14.

• Базе података (наставак)

Увоз података из екстерних докумената - опционални аргументи функције `read.table()`:

```
> read.table(file, header=F, sep="", quote="\\"", dec=".", row.names, col.names,  
+           as.is=F, na.strings="NA", colClasses=NA, nrows=-1, skip=0,  
+           check.names=T, fill=!blank.lines.skip, strip.white=F,  
+           blank.lines.skip=T, comment.char="#")
```

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

Варијанте функције `read.table()` су, такође, веома корисне јер имају различите *default* вредности.

```
> read.csv(file, header=T, sep=";", quote="\\"", dec=".", fill=T, comment.char="", ...)
> read.csv2(file, header=T, sep=";", quote="\\"", dec=";", fill=T, comment.char="", ...)
> read.delim(file, header=T, sep="\t", quote="\\"", dec=".", fill=T, comment.char="", ...)
> read.delim2(file, header=T, sep="\t", quote="\\"", dec=";", fill=T, comment.char="", ...)
> library("gdata", lib.loc="C:/Users/Lenchy/Documents/R/win-library/3.0")
> read.xls(xls, sheet=1, method=c("csv","tsv","tab"), perl="perl")
```

Алтернатива функцији `read.table()`, на изванан начин, јесте и функција `scan()`. Разлика између њих је у томе што се коришћењем функције `scan()` подаци “пакују” у вектор/листу. Ова функција је флексибилнија и ради брже (што је веома значајно код великих база). Код мањих база (са испод милион јединица података) тешко се уочава предност коришћења ове функције.

```
> my.data4 <- scan("C:/Users/Lenchy/Desktop/posao_fax/TU/VEZBE_1314/CAS2/ISIT.txt",
+ dec=".", what=list(0,0,0,0,0))
Read 789 records
> str(my.data4)
List of 5
 $ : num [1:789] 517 582 547 614 1068 ...
 $ : num [1:789] 28.7 27.9 23.4 18.3 12.4 ...
 $ : num [1:789] 1 1 1 1 1 1 1 1 1 1 ...
 $ : num [1:789] 50.2 50.2 50.2 50.2 50.2 ...
 $ : num [1:789] -14.5 -14.5 -14.5 -14.5 -14.5 ...
```

- Имплицитне петље (*implicit loops*)

Постоји фамилија функција типа 'apply', чијом се употребом избегава коришћење петљи.

- `apply()` примењује одређену функцију на врсте и/или колоне матрице; синтаксички она је облика

`apply(X, MARGIN, FUN, ...)`

X - матрица/база, MARGIN - да ли се ради са врстама (1), колонама (2) или са оба (`c(1, 2)`), FUN - функција (или чак оператор) која се примењује

```
> cm <- c(35, 14, 11, 1, 4, 11, 3, 0, 12, 9, 38, 4, 2, 5, 12, 2)
> dim(cm) <- rep(4,2)
> (rsum <- apply(cm, 1, sum))
[1] 53 39 64 7
> (cstd <- apply(cm, 2, sd))
[1] 14.291606 4.654747 15.195942 4.716991
> (rcchar <- apply(cm, c(1,2), as.character))
      [,1] [,2] [,3] [,4]
[1,] "35" "4"  "12" "2"
[2,] "14" "11" "9"  "5"
[3,] "11" "3"  "38" "12"
[4,] "1"  "0"  "4"  "2"
```

Ова функција враћа вектор или (вишедимензионалну) матрицу или листу.

Када је реч о функцији, то може бити функција коју корисник сам дефинише приликом позива функције `apply()` или се, пак, може искористити функција већ уграђена у неки од R пакета, њени евентуални аргументи наводе се на крају.

```
> apply(airquality, 2, function(x) mean(x, na.rm=T))
  Ozone  Solar.R   wind   Temp   Month   Day
42.129310 185.931507  9.957516  77.882353  6.993464 15.803922
> apply(airquality, 2, mean, na.rm=T)
  Ozone  Solar.R   wind   Temp   Month   Day
42.129310 185.931507  9.957516  77.882353  6.993464 15.803922
```

- `tapply()`, најједноставније речено, омогућава креирање табеле у којој су смештене вредности одређене функције примењене на подгрупе дефинисане другим аргументом; синтаксички она је облика

`tapply(X, INDEX, FUN, ...)`

X - вектор, INDEX - фактор/листа фактора, FUN - функција (или чак оператор) која се примењује

```
> class(iris[,"Species"])
[1] "factor"
> tapply(as.vector(iris[,4]), iris$Species, mean)
  setosa versicolor virginica
  0.246   1.326   2.026
```

Излаз је (вишедимензионална) матрица.

Алтернатива је функција `aggregate()`, која се разликује по томе што је X R објекат, а синтаксичка разлика је само у називу другог аргумента: уместо `INDEX` користи се `by`.

Ова функција враћа базу података.

```
> aggregate(iris[,1:4], list(iris$Species), mean)
  Group.1 Sepal.Length Sepal.Width Petal.Length Petal.Width
1  setosa      5.006      3.428      1.462      0.246
2 versicolor  5.936      2.770      4.260      1.326
3  virginica  6.588      2.974      5.552      2.026
```

- `lapply()` примењује одређену функцију на објекат типа листа (не би требало заборавити да је и база, такође, листа); излаз је листа.

```
> (my.list <- as.list(iris[1:3, 1:3]))
$Sepal.Length
[1] 5.1 4.9 4.7

$Sepal.Width
[1] 3.5 3.0 3.2

$Petal.Length
[1] 1.4 1.4 1.3
```

```
> lapply(my.list, min)
```

```
$Sepal.Length  
[1] 4.7
```

```
$Sepal.width  
[1] 3
```

```
$Petal.Length  
[1] 1.3
```

- `sapply()` је флексибилнија варијанта функције `lapply()`; када је год погодно она враћа најједноставнији могући објекат, типа вектор или матрица. Нарочито је корисна за добијање додатних информација о колонама базе података.

```
> sapply(my.list, min) #na izlazu se dobija numericki vektor umesto liste
```

```
Sepal.Length Sepal.Width Petal.Length  
4.7          3.0          1.3
```

```
> sapply(iris, is.factor)
```

```
Sepal.Length Sepal.Width Petal.Length Petal.width Species  
FALSE        FALSE        FALSE        FALSE        TRUE
```

```
> sapply(iris, function(x) if(!is.factor(x)) return(0) else length(levels(x)))
```

```
Sepal.Length Sepal.Width Petal.Length Petal.width Species  
0            0            0            0            3
```

- `vapply()`; `replicate()`; `mapply()`; `rapply()` итд.

- `by()` je objektno orijentisana varijanta funkcije `tapply()`, posebno pogodna za rad sa bazama podataka. Ona obezbeđuje razdvajanje podataka po nivoima jednog ili više faktora i primenu funkcije na tako dobijene podskupove podataka.

```
> attach(iris)
> by(Petal.Length, Species, mean)
Species: setosa
[1] 1.462
-----
Species: versicolor
[1] 4.26
-----
Species: virginica
[1] 5.552
> iris.m <- by(iris[, 1:4], Species, colMeans)
> class(iris.m)
[1] "by"
> str(iris.m)
List of 3
 $ setosa      : Named num [1:4] 5.006 3.428 1.462 0.246
  ..- attr(*, "names")= chr [1:4] "Sepal.Length" "Sepal.width" "Petal.Length" "Petal.width"
 $ versicolor: Named num [1:4] 5.94 2.77 4.26 1.33
  ..- attr(*, "names")= chr [1:4] "Sepal.Length" "Sepal.width" "Petal.Length" "Petal.width"
 $ virginica  : Named num [1:4] 6.59 2.97 5.55 2.03
  ..- attr(*, "names")= chr [1:4] "Sepal.Length" "Sepal.width" "Petal.Length" "Petal.width"
 - attr(*, "dim")= int 3
 - attr(*, "dimnames")=List of 1
  ..$ Species: chr [1:3] "setosa" "versicolor" "virginica"
 - attr(*, "call")= language by.data.frame(data = iris[, 1:4], INDICES = Species, FUN = colMeans)
 - attr(*, "class")= chr "by"
> iris.m$setosa
Sepal.Length Sepal.width Petal.Length Petal.width
      5.006      3.428         1.462         0.246
> iris.m$setosa[3] #isti izlaz daje i naredba iris.m$setosa["Petal.Length"]
Petal.Length
      1.462
```


- Splitting on a factor

Честа операција за којом постоји потреба при раду са базама података, јесте “разбијање” базе на више група података, дефинисаних одређеним фактором.

```
> iris.s <- split(iris, species); str(iris.s)
```

```
List of 3
 $ setosa      :'data.frame':   50 obs. of  5 variables:
  ..$ Sepal.Length: num [1:50] 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
  ..$ Sepal.width : num [1:50] 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
  ..$ Petal.Length: num [1:50] 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
  ..$ Petal.width : num [1:50] 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
  ..$ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1
  ...
 $ versicolor:'data.frame':   50 obs. of  5 variables:
  ..$ Sepal.Length: num [1:50] 7 6.4 6.9 5.5 6.5 5.7 6.3 4.9 6.6 5.2 ...
  ..$ Sepal.width : num [1:50] 3.2 3.2 3.1 2.3 2.8 2.8 3.3 2.4 2.9 2.7 ...
  ..$ Petal.Length: num [1:50] 4.7 4.5 4.9 4 4.6 4.5 4.7 3.3 4.6 3.9 ...
  ..$ Petal.width : num [1:50] 1.4 1.5 1.5 1.3 1.5 1.3 1.6 1 1.3 1.4 ...
  ..$ Species     : Factor w/ 3 levels "setosa","versicolor",..: 2 2 2 2 2 2 2 2 2 2
  ...
 $ virginica  :'data.frame':   50 obs. of  5 variables:
  ..$ Sepal.Length: num [1:50] 6.3 5.8 7.1 6.3 6.5 7.6 4.9 7.3 6.7 7.2 ...
  ..$ Sepal.width : num [1:50] 3.3 2.7 3 2.9 3 3 2.5 2.9 2.5 3.6 ...
  ..$ Petal.Length: num [1:50] 6 5.1 5.9 5.6 5.8 6.6 4.5 6.3 5.8 6.1 ...
  ..$ Petal.width : num [1:50] 2.5 1.9 2.1 1.8 2.2 2.1 1.7 1.8 1.8 2.5 ...
  ..$ Species     : Factor w/ 3 levels "setosa","versicolor",..: 3 3 3 3 3 3 3 3 3 3
  ...
```

```
> summary(iris.s$setosa$Petal.Length)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1.000	1.400	1.500	1.462	1.575	1.900

Функција `unsplit()` има супротан ефекат од функције `split()`. Она комбинује групе података и од њих саставља јединствену базу.

Заправо, при статистичкој обради података, а зависно од потреба, уобичајен је поступак:

split → *modify* → *unsplit*.

```
> iris.s2 <- lapply(iris.s, transform, Petal.Width=Petal.Width*10)
> #jedinica merenja sirine latice je mm, umesto cm kako stoji u originalnoj bazi
> ir.s <- unsplit(iris.s2, species)
> str(ir.s)
'data.frame': 150 obs. of 5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  2 2 2 2 2 4 3 2 2 1 ...
 $ Species      : Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...
```

- “Преобликовање” базе (*reshaping*)

У ситуацијама када постоје вишеструка мерења забележена за исти експериментални субјекат (нпр. услед праћења субјекта током времена или коришћења различитих техника), подаци могу бити организовани у:

- “широком” формату
- “дугачком” формату

Широк формат подразумева да за сваког субјекта постоји тачно једна (опсервациона) врста, а свако мерење је смештено у засебну променљиву. Дугачак формат подразумева да постоји по једна (опсервациона) врста за свако мерење (дакле, истом субјекту кореспондира већи број врста).

Различите функције захтевају различите формате, па се услед тога јавља потреба за “преобликовањем” база.

```
> hsb2 <- read.table('http://www.ats.ucla.edu/stat/r/modules/hsb2.csv', header=T, sep=",")
```

```
> hsb2[1:10,]
```

	id	female	race	ses	schtyp	prog	read	write	math	science	socst
1	70	0	4	1	1	1	57	52	41	47	57
2	121	1	4	2	1	3	68	59	53	63	61
3	86	0	4	3	1	1	44	33	54	58	31
4	141	0	4	3	1	3	63	44	47	53	56
5	172	0	4	2	1	2	47	52	57	53	61
6	113	0	4	2	1	2	44	52	51	63	61
7	50	0	3	2	1	1	50	59	42	53	61
8	11	0	1	2	1	2	34	46	45	39	36
9	84	0	4	2	1	1	63	57	54	58	51
10	48	0	3	2	1	2	57	55	52	50	51

```
> #siroki u dugacki format
```

```
> l <- reshape(hsb2, varying = c("read", "write", "math", "science", "socst"),
```

```
+ v.names = "score", timevar = "subj",
```

```
+ times = c("read", "write", "math", "science", "socst"),
```

```
+ new.row.names = 1:1000,
```

```
+ direction = "long")
```

```
> l.sort <- l[order(l$id),]
```

```
> l.sort[1:10,]
```

	id	female	race	ses	schtyp	prog	subj	score
99	1	1	1	1	1	3	read	34
299	1	1	1	1	1	3	write	44
499	1	1	1	1	1	3	math	40
699	1	1	1	1	1	3	science	39
899	1	1	1	1	1	3	socst	41
139	2	1	1	2	1	3	read	39
339	2	1	1	2	1	3	write	41
539	2	1	1	2	1	3	math	33
739	2	1	1	2	1	3	science	42
939	2	1	1	2	1	3	socst	41

```
> #dugacak u široki format
> w <- reshape(l.sort, timevar = "subj",
+             idvar = c("id", "female", "race", "ses", "schtyp", "prog"),
+             direction = "wide")
> w[1:10,]
```

	id	female	race	ses	schtyp	prog	score.read	score.write	score.math	score.science	score.socst
99	1	1	1	1	1	3	34	44	40	39	41
139	2	1	1	2	1	3	39	41	33	42	41
84	3	0	1	1	1	2	63	65	48	63	56
112	4	1	1	1	1	2	44	50	41	39	51
76	5	0	1	1	1	2	47	40	43	45	31
149	6	1	1	1	1	2	47	41	46	40	41
50	7	0	1	2	1	2	57	54	59	47	51
94	8	1	1	1	1	2	39	44	52	44	48
60	9	0	1	2	1	3	48	49	52	44	51
154	10	1	1	2	1	1	47	54	49	53	61

Преуређење базе података може се вршити и помоћу функције `stack()` и њој инверзне функције `unstack()`. Функција `stack()` надовезује више вектора из различитих колона базе у један вектор упоредан фактору који указује одакле потиче која опсервација.

```
> data <- data.frame(A=sample(1:6, 2), B=sample(1:6, 2), C=sample(1:6, 2))
```

```
> data
  A B C
1 3 5 5
2 4 3 4
> stack(data)
  values ind
1      3   A
2      4   A
3      5   B
4      3   B
5      5   C
6      4   C
```

```

> stack(data, select=c("A", "B")) #selektuje se koje kolone bi trebalo "sloziti"
  values ind
1      3   A
2      4   A
3      5   B
4      3   B

> data(PlantGrowth); str(PlantGrowth)
'data.frame':  30 obs. of  2 variables:
 $ weight: num  4.17 5.58 5.18 6.11 4.5 4.61 5.17 4.53 5.33 5.14 ...
 $ group : Factor w/ 3 levels "ctrl","trt1",..: 1 1 1 1 1 1 1 1 1 1 ...
> max(PlantGrowth$weight[PlantGrowth$group=="ctrl"])
[1] 6.11

> pg <- unstack(PlantGrowth, weight ~ group); str(pg)
'data.frame':  10 obs. of  3 variables:
 $ ctrl: num  4.17 5.58 5.18 6.11 4.5 4.61 5.17 4.53 5.33 5.14
 $ trt1: num  4.81 4.17 4.41 3.59 5.87 3.83 6.03 4.89 4.32 4.69
 $ trt2: num  6.31 5.12 5.54 5.5 5.37 5.29 4.92 6.15 5.8 5.26
> max(pg$ctrl)
[1] 6.11

```

Може се приметити да је први аргумент обе функције објекат (база) коју би требало преуредити. Други аргумент функције `unstack()` је формула:

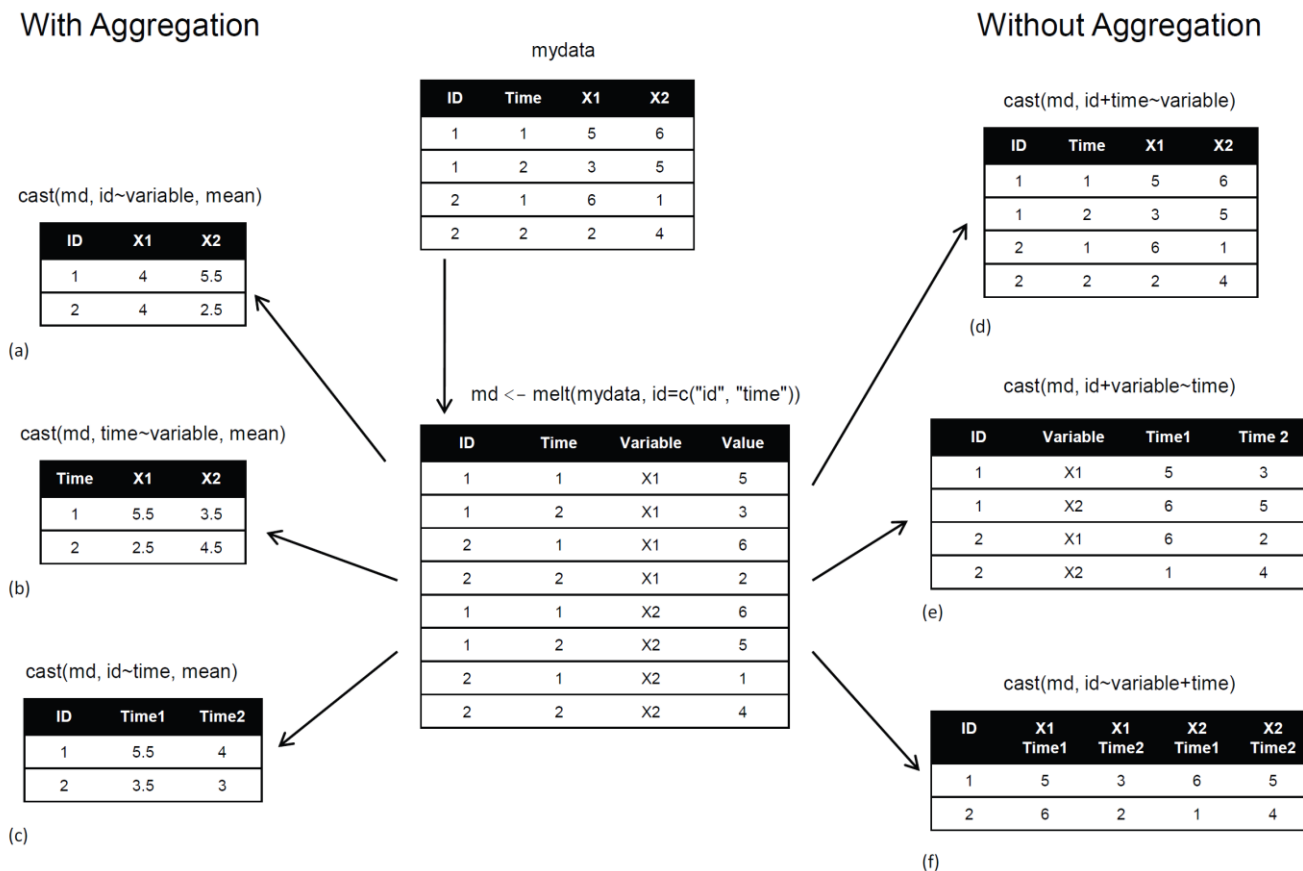
values to unstack ~ groups to create.

Иначе, нотација облика $y \sim x$ се интерпретира као “у описано коришћењем x ”. То је први и основни пример формуле (*model formula*).

Напредније могућности и свестранији приступ реструктурирању и здруживању података пружа пакет `reshape`.

Оно што се суштински дешава јесте да корисник “топи” (`melt()`) податке, да би их, након тога, “излио” (`cast()`) у жељени калуп, и, при томе, може да врши здруживање података на жељени начин.

Reshaping a Dataset



- Табеле (*table*)

Код експерименталних података често се појављују категоричке променљиве, које указују на неку потподелу података у групе. Њих би у R-у требало назначити као факторе (*factors*), при чему се разликују два типа: неуређени (номинални) и уређени (ординални).

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()`.

- Функција `table()` даје табеларни приказ фреквенције за сваки ниво (или упоредо - нивое) одређене категоричке променљиве/променљивих.


```
> Deer <- read.table("C:/Users/Lenchy/Desktop/posao_fax/TU/VEZBE_1314/CAS2/Deer.txt", header=T)
```

```
> names(Deer)
```

```
[1] "Farm" "Month" "Year" "Sex" "clas1_4" "LCT" "KFI" "Ecervi" "Tb"
```

```
> str(Deer)
```

```
'data.frame': 1182 obs. of 9 variables:
 $ Farm : Factor w/ 27 levels "AL","AU","BA",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ Month : int 10 10 10 10 10 10 10 10 10 10 ...
 $ Year : int 0 0 0 0 0 0 0 0 0 0 ...
 $ Sex : int 1 1 1 1 1 1 1 1 1 1 ...
 $ clas1_4: int 4 4 3 4 4 4 4 4 4 4 ...
 $ LCT : num 191 180 192 196 204 190 196 200 197 208 ...
 $ KFI : num 20.4 16.4 15.9 17.3 NA ...
 $ Ecervi : num 0 0 2.38 0 0 0 1.21 0 0.8 0 ...
 $ Tb : int 0 0 0 0 NA 0 NA 1 0 0 ...
```

```
> table(Deer$Farm) #dobija se tabela sa brojevima uzorkovanih zivotinja sa svake od farmi
```

AL	AU	BA	BE	CB	CRC	HB	LCV	LN	MAN	MB	MO	NC	NV	PA	PN	QM	RF
15	37	98	19	93	16	35	2	34	76	41	278	32	35	11	45	75	34
RÑ	RO	SAL	SAU	SE	TI	TN	VISO	VY									
25	44	1	3	26	21	31	15	40									

```
> my.deer <- table(Deer$Sex, Deer$Year) #dobija se tabela sa brojevima uzorkovanih zivotinja
odredjenog pola za svaku godinu
```

```
> colnames(my.deer) <- c("2000.", "2001.", "2002.", "2003.", "2004.", "2005.", "1999.")
```

```
> rownames(my.deer) <- c("F", "M")
```

```
> names(dimnames(my.deer)) <- c("pol", "godina")
```

```
> my.deer
```

	godina						
pol	2000.	2001.	2002.	2003.	2004.	2005.	1999.
F	115	85	154	75	78	34	21
M	76	40	197	123	60	35	0

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

Важно је поменути један опционални аргумент функције `table()`, а то је `useNA`, који може бити постављен на једну од вредности `"no"`, `"ifany"`, `"always"`.

- Функцијом `margin.table()` формирају се маргиналне табеле, које садрже збирове фреквенција по једној или другој димензији табеле прослеђене као први аргумент функције. Други аргумент је маргинални индекс: 1 даје укупне збирове по врстама, а 2 по колонама.

```
> my.deer.marg <- margin.table(my.deer, 2)
> my.deer.marg
godina
2000. 2001. 2002. 2003. 2004. 2005. 1999.
191   125   351   198   138   69   21
```

- Функција `prop.table()` даје приказ релативних фреквенција, у смислу да се свака вредност из табеле прослеђене као први аргумент функције, скалира тако што се подели укупним збиром по врсти, односно колони којој припада.

```
> my.deer.prop <- prop.table(my.deer, 1)
> my.deer.prop
godina
pol 2000. 2001. 2002. 2003. 2004. 2005. 1999.
F 0.20462633 0.15124555 0.27402135 0.13345196 0.13879004 0.06049822 0.03736655
M 0.14312618 0.07532957 0.37099812 0.23163842 0.11299435 0.06591337 0.00000000
```

- Функција summary()

Спада у генеричке функције (функције које реагују на објекат, прослеђен као аргумент функције, на начин који је осетљив на његову класу).

То је најчешће коришћена функција за издвајање сумарних резултата анализе променљиве, који су основни, али довољно детаљни.

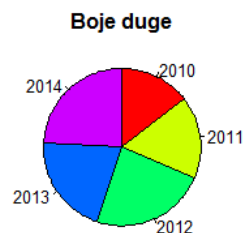
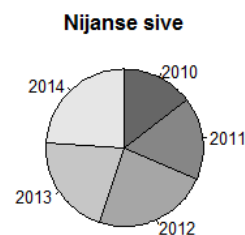
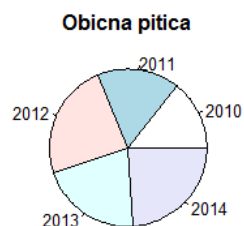
```
> (s.stat <- summary(Deer))
```

Farm	Month	Year	Sex	clas1_4
MO : 278	Min. : 1.000	Min. : 0.000	Min. : 1.000	Min. : 1.00
BA : 98	1st Qu.: 2.000	1st Qu.: 1.000	1st Qu.: 1.000	1st Qu.: 3.00
CB : 93	Median : 9.000	Median : 2.000	Median : 1.000	Median : 3.00
MAN : 76	Mean : 6.838	Mean : 3.959	Mean : 1.478	Mean : 3.15
QM : 75	3rd Qu.: 11.000	3rd Qu.: 3.000	3rd Qu.: 2.000	3rd Qu.: 4.00
PN : 45	Max. : 12.000	Max. : 99.000	Max. : 2.000	Max. : 4.00
(Other): 517	NA's : 61	NA's : 60	NA's : 30	NA's : 54
LCT	KFI	Ecervi	Tb	
Min. : 75.0	Min. : 2.42	Min. : 0.000	Min. : 0.000	
1st Qu.: 151.0	1st Qu.: 14.02	1st Qu.: 0.000	1st Qu.: 0.000	
Median : 163.0	Median : 22.91	Median : 6.495	Median : 0.000	
Mean : 161.5	Mean : 34.80	Mean : 46.232	Mean : 0.123	
3rd Qu.: 174.0	3rd Qu.: 43.17	3rd Qu.: 36.602	3rd Qu.: 0.000	
Max. : 216.0	Max. : 294.32	Max. : 2186.600	Max. : 1.000	
NA's : 104	NA's : 335	NA's : 318	NA's : 288	

• Графички приказ табела (*graphical display of tables*)

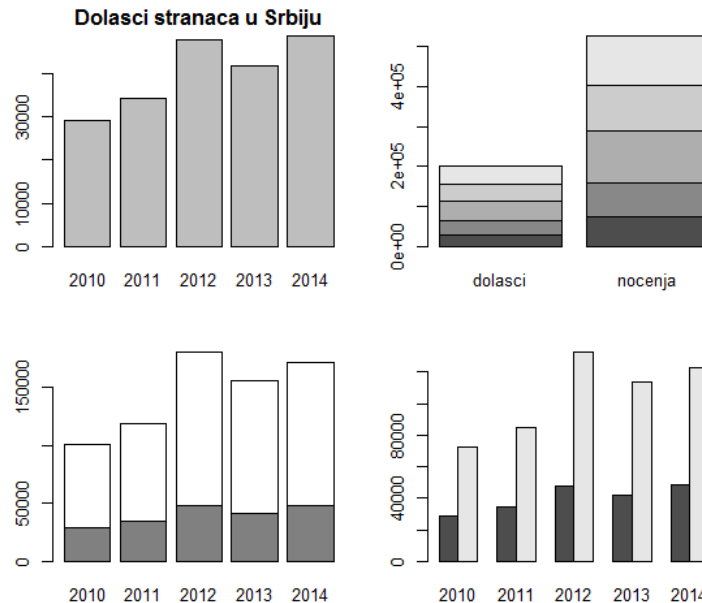
▫ **Питице** (*the pie chart*)

```
> rzs.dolasci <- read.table("C:/Users/Lenchy/Desktop/posao_fax/TU/dolasci.txt", header=T)
> names(rzs.dolasci)
[1] "Godina" "Bgd_d" "Bgd_s" "vjvd_d" "vjvd_s" "Smd_Z_d" "Smd_Z_s" "JI_d"
[9] "JI_s"
> dolasci <- rowSums(rzs.dolasci[, seq(3,9,by=2)])
> names(dolasci) <- rzs.dolasci[, 1]
> dolasci
 2010 2011 2012 2013 2014
29180 34503 48113 41919 48763
> par(mfrow = c(2, 2), mar = c(3, 3, 2, 1))
> pie(dolasci, main = "Obicna pitica")
> pie(dolasci, col = gray(seq(0.4, 0.9, length = 5)), clockwise = TRUE, main = "Nijanse sive")
> pie(dolasci, col = rainbow(5), clockwise = TRUE, main = "Boje duge")
> library(plotrix)
> pie3D(dolasci, labels = names(Cases), explode = 0.1, main = "3D pitica", labelcex = 0.6)
```



▫ Барови (*the bar chart*)

```
> rzs.nocenja <- read.table("C:/Users/Lenchy/Desktop/posao_fax/TU/nocenja.txt", header=T)
> nocenja <- rowSums(rzs.nocenja[, seq(3,9,by=2)])
> names(nocenja) <- rzs.nocenja[, 1]
> nocenja
  2010  2011  2012  2013  2014
72191 84888 132648 113608 122802
> par(mfrow = c(2, 2), mar = c(3, 3, 2, 1))
> barplot(dolasci , main = "Dolasci stranaca u Srbiju")
> br <- cbind(dolasci, nocenja)
> barplot(br)
> barplot(t(br), col = gray(c(0.5, 1)))
> barplot(t(br), beside = TRUE)
```



У презентацији коришћене су следеће базе података:

- ISIT.txt - садржи експерименталне податке забележене приликом истраживања пелагичних биолуминесцентних организама који настањују велике дубине Атлантика. Подаци су прикупљени током 2001/2002. године за време трајања серије од четири крстарења брода 'Royal Research Ship Discovery'.
- hsb2.csv - назив је скраћеница од *high school and beyond*. Садржи 200 опсервација из узорка средњошколаца, при чему су бележени демографски подаци о ученицима (пол, социјално-економски статус, етничка припадност). Такође, регистровани су и бројеви поена које су испитаници остварили на пет стандардизованих тестова.
- Deer.txt - садржи епидемиолошке податке. Опсервације се односе на дивљу свињу и јелена, који се узгајају на бројним поседима широм Шпаније. База садржи податке о туберкулози код обе врсте и паразиту *Elaphostrongylus cervi*, који инфицира само јелена.