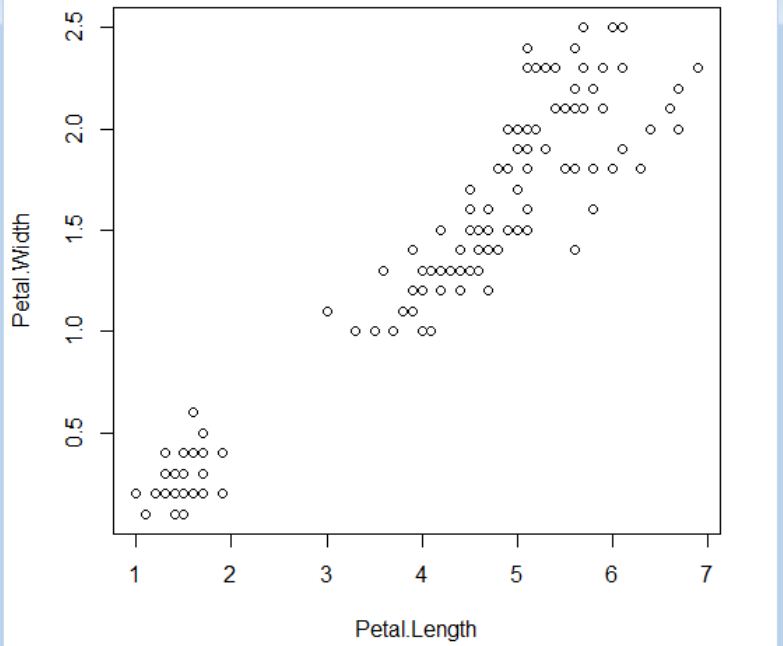
RGui

File History Resize Windows

R Console

```
> data(iris)
> attach(iris)
>
> str(iris)
'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  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 ...
> plot(Petal.Length, Petal.Width)
>
```

R Graphics: Device 2 (ACTIVE)



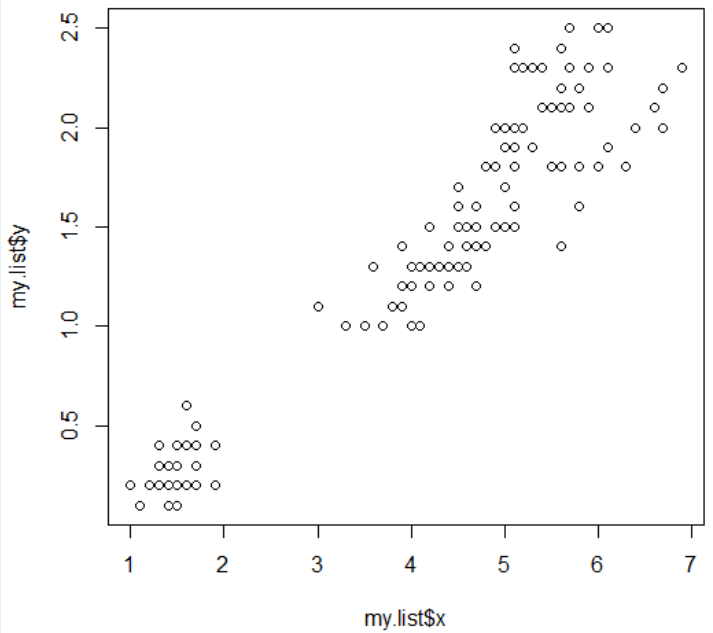
RGui

File Edit View Misc Packages Windows Help

R Console

```
> data(iris)
> my.list<-as.list(iris[,c(3,4)])
> names(my.list)<-c("x","y")
> plot(my.list)
>
```

R Graphics: Device 2 (ACTIVE)



RGui

File Edit View Misc Packages Windows Help

R Console

```
> attach(iris)
> plot(Petal.Length)
>
```

R Graphics: Device 2 (ACTIVE)

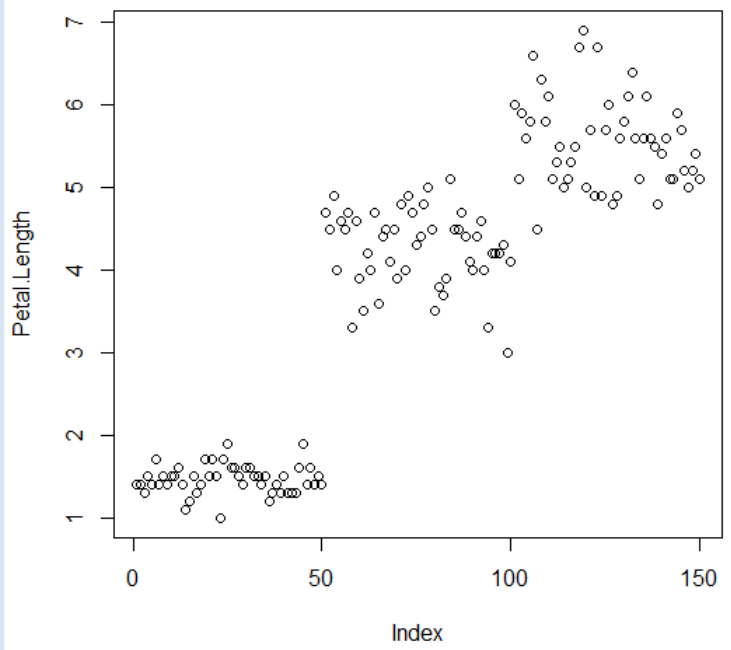


Figure 7: Plotting symbols

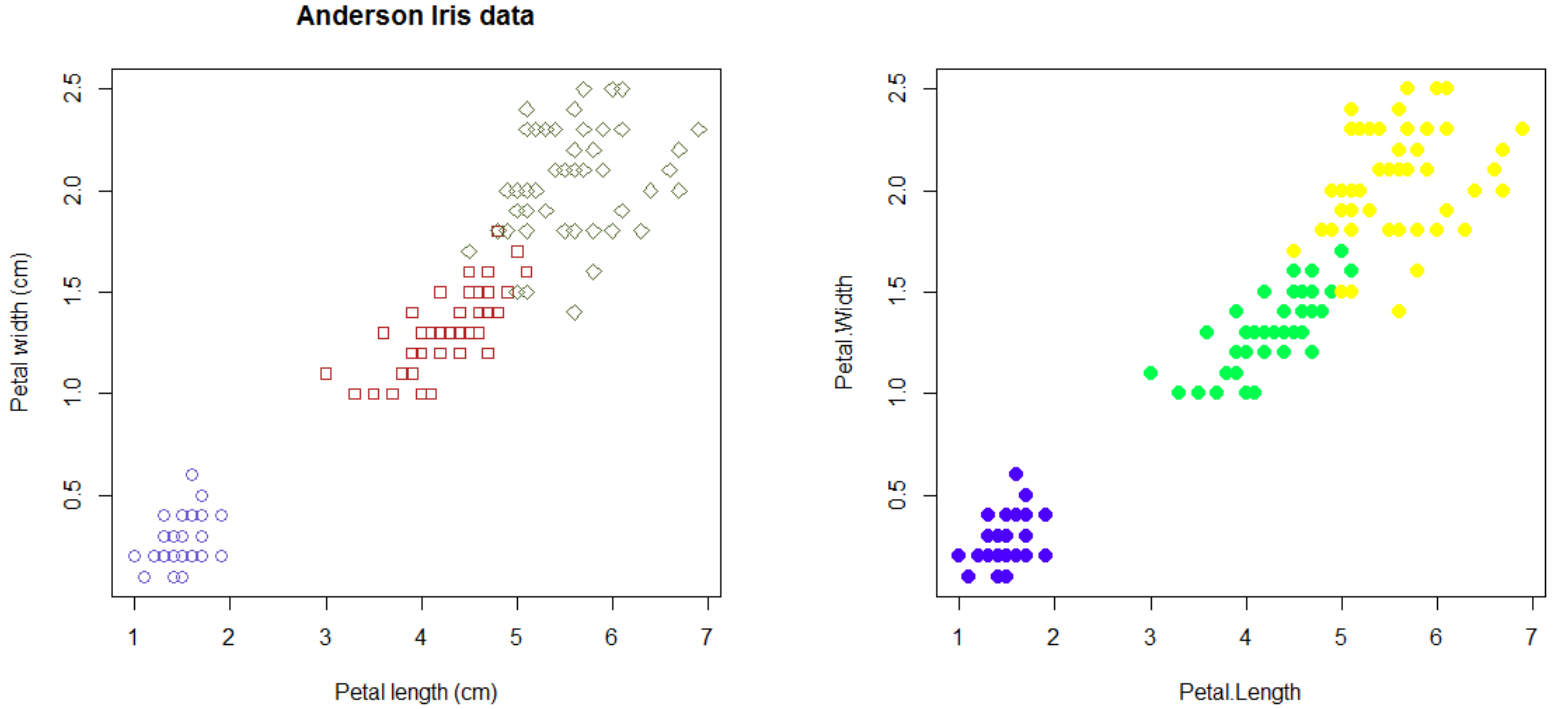
abline	Add a Straight Line
arrows	Add Arrows
axis	Add an Axis
box	Draw a framing Box
grid	Add a Grid
legend	Add Legends
lines	Add Connected Line Segments
mtext	Write Text into the Margins
points	Add Points
polygon	Draw Polygons
rect	Draw Rectangles
rug	Add a Rug
segments	Add Line Segments
symbols	Draw Symbols
text	Add Text
title	Plot Annotation

Table 1: Methods for adding to an existing base graphics plot

RGui - [R Console]

File Edit View Misc Packages Windows Help

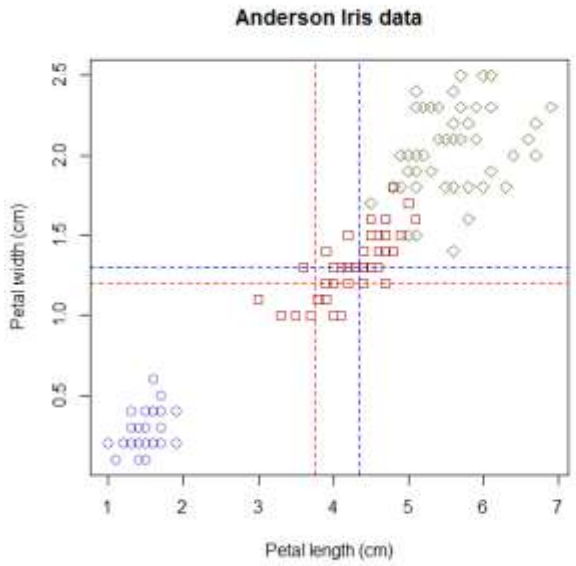
```
> plot(Petal.Length, Petal.Width, pch=(21:23)[as.numeric(Species)], cex=1.2,
+ xlab="Petal length (cm)", ylab="Petal width (cm)",
+ main="Anderson Iris data",
+ col=c("slateblue", "firebrick", "darkolivegreen")[as.numeric(Species)])
>
>
> plot(Petal.Length, Petal.Width, pch=20,cex=2,
+ col=topo.colors(3)[as.numeric(Species)])
>
```

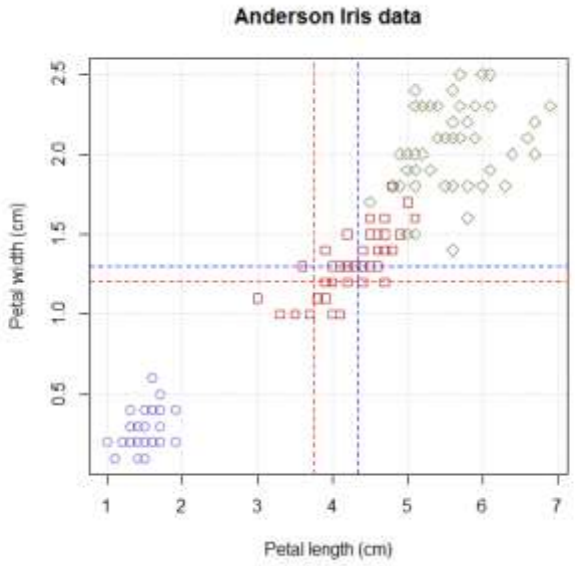


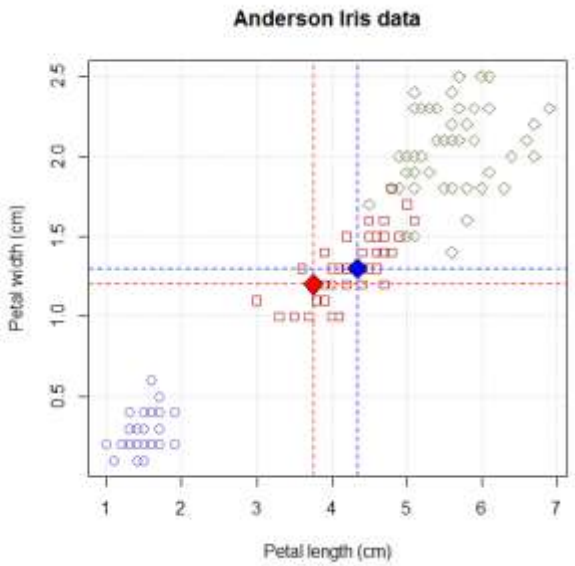
RGui - [R Console]

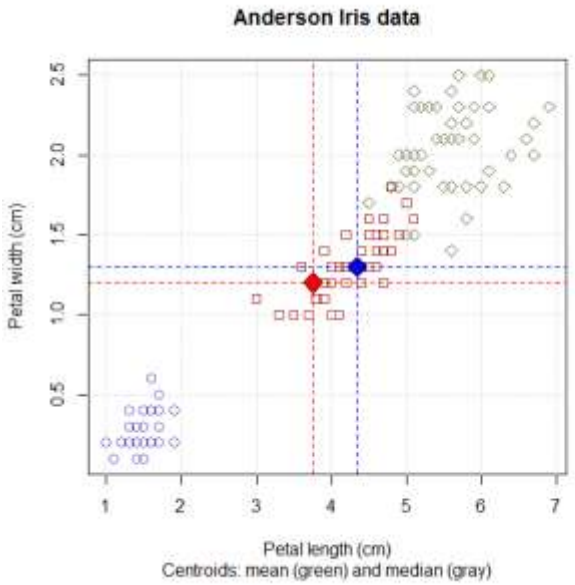
File Edit View Misc Packages Windows Help

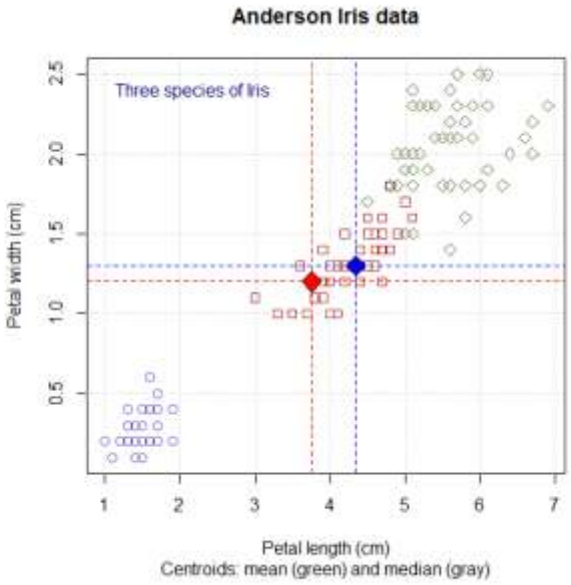
```
> plot(Petal.Length, Petal.Width, pch=(21:23)[as.numeric(Species)], cex=1.2,
+ xlab="Petal length (cm)", ylab="Petal width (cm)",
+ main="Anderson Iris data",
+ col=c("slateblue", "firebrick", "darkolivegreen")[as.numeric(Species)])
>
> abline(v=mean(Petal.Length), lty=2, col="red")
> abline(h=mean(Petal.Width), lty=2, col="red")
> abline(v=median(Petal.Length), lty=2, col="blue")
> abline(h=median(Petal.Width), lty=2, col="blue")
> #lty je argument kojim se odredjuje tip linije, ovde je postavljen na 2 odnosno "dashed"
>
> grid()
> #dodaje svetlo sivu mrezu, koja odgovara podeocima na osama
>
> points(mean(Petal.Length), mean(Petal.Width),
+ cex=2, pch=23, col="black", bg="red")
> points(median(Petal.Length), median(Petal.Width),
+ cex=2, pch=23, col="black", bg="blue")
> #dodaju se tacke sa odredjenim koordinatama, koje su naglasene svojim izgledom
>
> title(sub="Centroids: mean (green) and median (gray)")
>
> text(1, 2.4, "Three species of Iris", pos=4, col="navyblue")
>
> legend(1, 2.4, levels(Species), pch=21:23, bty="n",
+ col=c("slateblue", "firebrick", "darkolivegreen"))
>
```

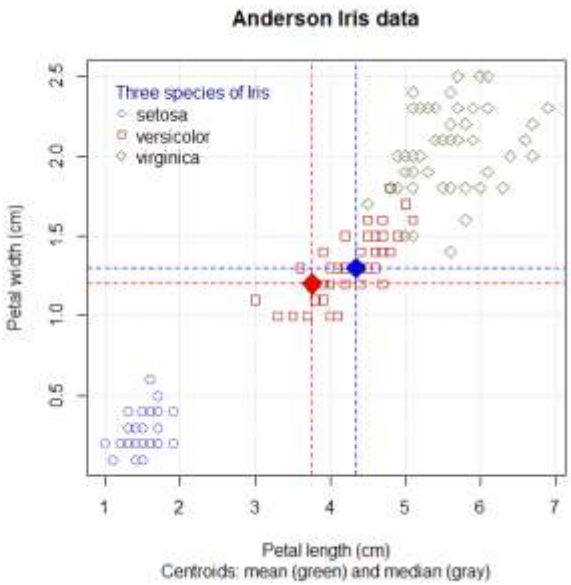

Anderson Iris data
Petal width (cm)
Petal length (cm)


Anderson Iris data
Petal width (cm)
Petal length (cm)


Anderson Iris data
Petal width (cm)
Petal length (cm)


Anderson Iris data
Petal width (cm)
Petal length (cm)
Centroids: mean (green) and median (gray)


Anderson Iris data
Three species of Iris
Petal width (cm)
Petal length (cm)
Centroids: mean (green) and median (gray)


Anderson Iris data
Three species of Iris
setosa
versicolor
virginica
Petal width (cm)
Petal length (cm)
Centroids: mean (green) and median (gray)

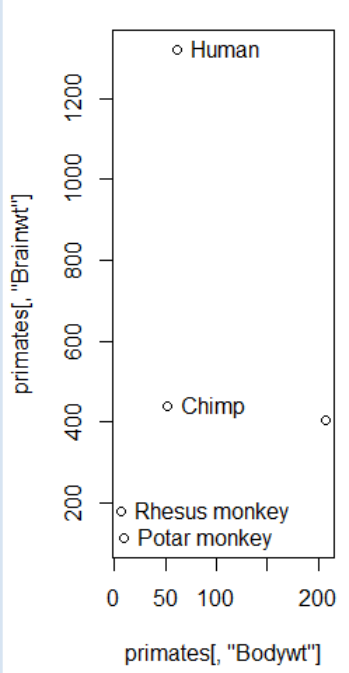
RGui

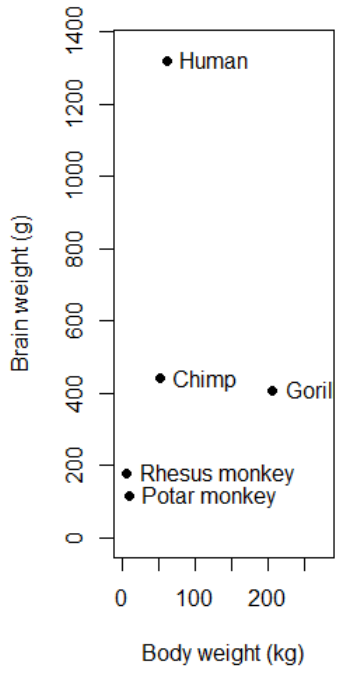
File Edit View Misc Packages Windows Help

R Console

```
> par(mfrow=c(1,2))
> primates<-matrix(c(10,115,207,406,62,1320,6.8,179,52.2,440),ncol=2,byrow=T,
+ dimnames=list(c("Potar monkey","Gorilla","Human","Rhesus monkey","Chimp"),
+ c("Bodywt","Brainwt")))
> plot(primates[,"Bodywt"],primates[,"Brainwt"])
> text(primates[,"Bodywt"],primates[,"Brainwt"],labels=row.names(primates),pos=4)
> plot(x=primates[,"Bodywt"], y=primates[,"Brainwt"], pch=16,
+ xlab="Body weight (kg)", ylab="Brain weight (g)",
+ xlim=c(0,280), ylim=c(0,1350))
> text(primates[,"Bodywt"],primates[,"Brainwt"],labels=row.names(primates),pos=4)
>
> #pos=4 rezultira pozicioniranjem teksta sa desne strane tacke
> #za pozicioniranje teksta sa leve strane tacke treba postaviti pos=2
>
>
> par(mfrow=c(1,1)) #da bi se buduci grafici iscrtavali po jedan u prozoru - reset
>
```

R Graphics: Device 2 (ACTIVE)





adj	controls text justification with respect to the left border of the text so that 0 is left-justified, 0.5 is centred, 1 is right-justified, values > 1 move the text further to the left, and negative values further to the right; if two values are given (e.g., c(0, 0)) the second one controls vertical justification with respect to the text baseline
bg	specifies the colour of the background (e.g., bg="red", bg="blue"; the list of the 657 available colours is displayed with colors())
bty	controls the type of box drawn around the plot, allowed values are: "o", "l", "7", "c", "u" ou "]" (the box looks like the corresponding character); if bty="n" the box is not drawn
cex	a value controlling the size of texts and symbols with respect to the default; the following parameters have the same control for numbers on the axes, cex.axis, the axis labels, cex.lab, the title, cex.main, and the sub-title, cex.sub
col	controls the colour of symbols; as for cex there are: col.axis, col.lab, col.main, col.sub
font	an integer which controls the style of text (1: normal, 2: italics, 3: bold, 4: bold italics); as for cex there are: font.axis, font.lab, font.main, font.sub
las	an integer which controls the orientation of the axis labels (0: parallel to the axes, 1: horizontal, 2: perpendicular to the axes, 3: vertical)
lty	controls the type of lines, can be an integer (1: solid, 2: dashed, 3: dotted, 4: dotdash, 5: longdash, 6: twodash), or a string of up to eight characters (between "0" and "9") which specifies alternatively the length, in points or pixels, of the drawn elements and the blanks, for example lty="44" will have the same effet than lty=2
lwd	a numeric which controls the width of lines
mar	a vector of 4 numeric values which control the space between the axes and the border of the graph of the form c(bottom, left, top, right), the default values are c(5.1, 4.1, 4.1, 2.1)
mfcol	a vector of the form c(nr,nc) which partitions the graphic window as a matrix of nr lines and nc columns, the plots are then drawn in columns (see section 4.1.2)
mfrow	id. but the plots are then drawn in line (see section 4.1.2)
pch	controls the type of symbol, either an integer between 1 and 25, or any single character within "" (Fig. 2)
ps	an integer which controls the size in points of texts and symbols
pty	a character which specifies the type of the plotting region, "s": square, "m": maximal
tck	a value which specifies the length of tick-marks on the axes as a fraction of the smallest of the width or height of the plot; if tck=1 a grid is drawn
tcl	id. but as a fraction of the height of a line of text (by default tcl=-0.5)
xaxt	if xaxt="n" the x-axis is set but not drawn (useful in conjunction with axis(side=1, ...))
yaxt	if yaxt="n" the y-axis is set but not drawn (useful in conjunction with axis(side=2, ...))

assocplot	Association Plots
barplot	Bar Plots
boxplot	Box Plots
contour	Contour Plots
coplot	Conditioning Plots
dotchart	Cleveland Dot Plots
filled.contour	Level (Contour) Plots
fourfoldplot	Fourfold Plots
hist	Histograms
image	Display a Colour Image
matplot	Plot Columns of Matrices
mosaicplot	Mosaic Plots
pairs	Scatterplot Matrices
persp	Perspective (3D) Plots
plot	Generic X-Y Plotting
stars	Star (Spider/Radar) Plots
stem	Stem-and-Leaf Plots
stripchart	1-D Scatter Plots
sunflowerplot	Sunflower Scatter Plots

Table 2: Base graphics plot types

For each function, the options may be found with the on-line help in R. Some of these options are identical for several graphical functions; here are the main ones (with their possible default values):

`add=FALSE`	if `TRUE` superposes the plot on the previous one (if it exists)
`axes=TRUE`	if `FALSE` does not draw the axes and the box
`type="p"`	specifies the type of plot, `"p"`: points, `"l"`: lines, `"b"`: points connected by lines, `"o"`: id. but the lines are over the points, `"h"`: vertical lines, `"s"`: steps, the data are represented by the top of the vertical lines, `"S"`: id. but the data are represented by the bottom of the vertical lines
`xlim=, ylim=`	specifies the lower and upper limits of the axes, for example with `xlim=c(1, 10)` or `xlim=range(x)`
`xlab=, ylab=`	annotates the axes, must be variables of mode character
`main=`	main title, must be a variable of mode character
`sub=`	sub-title (written in a smaller font)

`plot(x)`	plot of the values of `x` (on the y-axis) ordered on the x-axis
`plot(x, y)`	bivariate plot of `x` (on the x-axis) and `y` (on the y-axis)
`sunflowerplot(x, y)`	id. but the points with similar coordinates are drawn as a flower which petal number represents the number of points
`pie(x)`	circular pie-chart
`boxplot(x)`	"box-and-whiskers" plot
`stripchart(x)`	plot of the values of `x` on a line (an alternative to `boxplot()` for small sample sizes)
`coplot(x~y | z)`	bivariate plot of `x` and `y` for each value (or interval of values) of `z`
`interaction.plot (f1, f2, y)`	if `f1` and `f2` are factors, plots the means of `y` (on the y-axis) with respect to the values of `f1` (on the x-axis) and of `f2` (different curves); the option `fun` allows to choose the summary statistic of `y` (by default `fun=mean`)
`matplot(x,y)`	bivariate plot of the first column of `x` *vs.* the first one of `y`, the second one of `x` *vs.* the second one of `y`, etc.
`dotchart(x)`	if `x` is a data frame, plots a Cleveland dot plot (stacked plots line-by-line and column-by-column)
`fourfoldplot(x)`	visualizes, with quarters of circles, the association between two dichotomous variables for different populations (`x` must be an array with `dim=c(2, 2, k)`, or a matrix with `dim=c(2, 2)` if $k = 1$)
`assocplot(x)`	Cohen–Friendly graph showing the deviations from independence of rows and columns in a two dimensional contingency table
`mosaicplot(x)`	'mosaic' graph of the residuals from a log-linear regression of a contingency table
`pairs(x)`	if `x` is a matrix or a data frame, draws all possible bivariate plots between the columns of `x`
`plot.ts(x)`	if `x` is an object of class `"ts"`, plot of `x` with respect to time, `x` may be multivariate but the series must have the same frequency and dates
`ts.plot(x)`	id. but if `x` is multivariate the series may have different dates and must have the same frequency
`hist(x)`	histogram of the frequencies of `x`
`barplot(x)`	histogram of the values of `x`
`qqnorm(x)`	quantiles of `x` with respect to the values expected under a normal law
`qqplot(x, y)`	quantiles of `y` with respect to the quantiles of `x`
`contour(x, y, z)`	contour plot (data are interpolated to draw the curves), `x` and `y` must be vectors and `z` must be a matrix so that `dim(z)=c(length(x), length(y))` (`x` and `y` may be omitted)
`filled.contour (x, y, z)`	id. but the areas between the contours are coloured, and a legend of the colours is drawn as well
`image(x, y, z)`	id. but the actual data are represented with colours
`persp(x, y, z)`	id. but in perspective
`stars(x)`	if `x` is a matrix or a data frame, draws a graph with segments or a star where each row of `x` is represented by a star and the columns are the lengths of the segments
`symbols(x, y, ...)`	draws, at the coordinates given by `x` and `y`, symbols (circles, squares, rectangles, stars, thermometres or "boxplots") which sizes, colours, etc, are specified by supplementary arguments
`termplot(mod.obj)`	plot of the (partial) effects of a regression model (`mod.obj`)

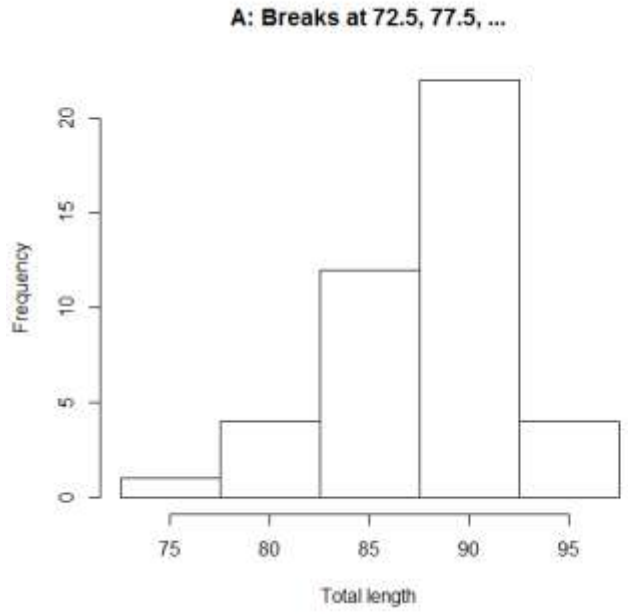
R has a set of graphical functions which affect an already existing graph: they are called *low-level plotting commands*. Here are the main ones:

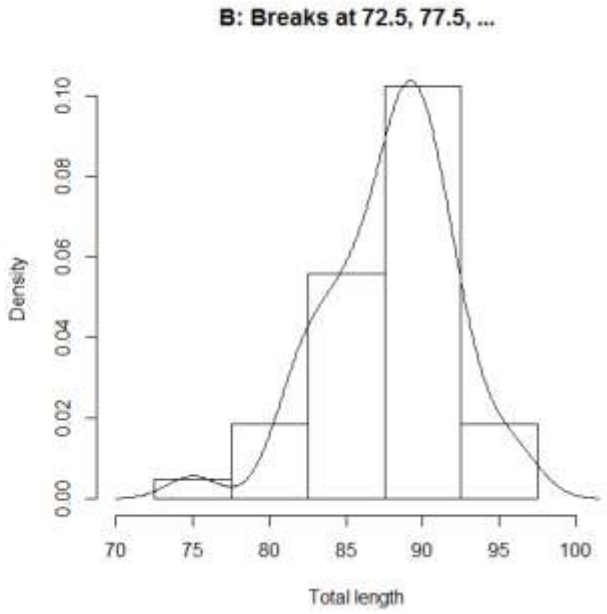
`points(x, y)`	adds points (the option `type=` can be used)
`lines(x, y)`	id. but with lines
`text(x, y, labels, ...)`	adds text given by `labels` at coordinates (x,y); a typical use is: `plot(x, y, type="n"); text(x, y, names)`
`mtext(text, side=3, line=0, ...)`	adds text given by `text` in the margin specified by `side` (see `axis()` below); `line` specifies the line from the plotting area
`segments(x0, y0, x1, y1)`	draws lines from points (x0,y0) to points (x1,y1)
`arrows(x0, y0, x1, y1, angle= 30, code=2)`	id. with arrows at points (x0,y0) if `code=2`, at points (x1,y1) if `code=1`, or both if `code=3`; `angle` controls the angle from the shaft of the arrow to the edge of the arrow head
`abline(a,b)`	draws a line of slope `b` and intercept `a`
`abline(h=y)`	draws a horizontal line at ordinate `y`
`abline(v=x)`	draws a vertical line at abcissa `x`
`abline(lm.obj)`	draws the regression line given by `lm.obj` (see section 5)
`rect(x1, y1, x2, y2)`	draws a rectangle which left, right, bottom, and top limits are `x1`, `x2`, `y1`, and `y2`, respectively
`polygon(x, y)`	draws a polygon linking the points with coordinates given by `x` and `y`
`legend(x, y, legend)`	adds the legend at the point (x,y) with the symbols given by `legend`
`title()`	adds a title and optionally a sub-title
`axis(side, vect)`	adds an axis at the bottom (`side=1`), on the left (2), at the top (3), or on the right (4); `vect` (optional) gives the abcissa (or ordinates) where tick-marks are drawn
`box()`	adds a box around the current plot
`rug(x)`	draws the data `x` on the x-axis as small vertical lines
`locator(n, type="n", ...)`	returns the coordinates (x, y) after the user has clicked `n` times on the plot with the mouse; also draws symbols (`type="p"`) or lines (`type="l"`) with respect to optional graphic parameters (...); by default nothing is drawn (`type="n"`)

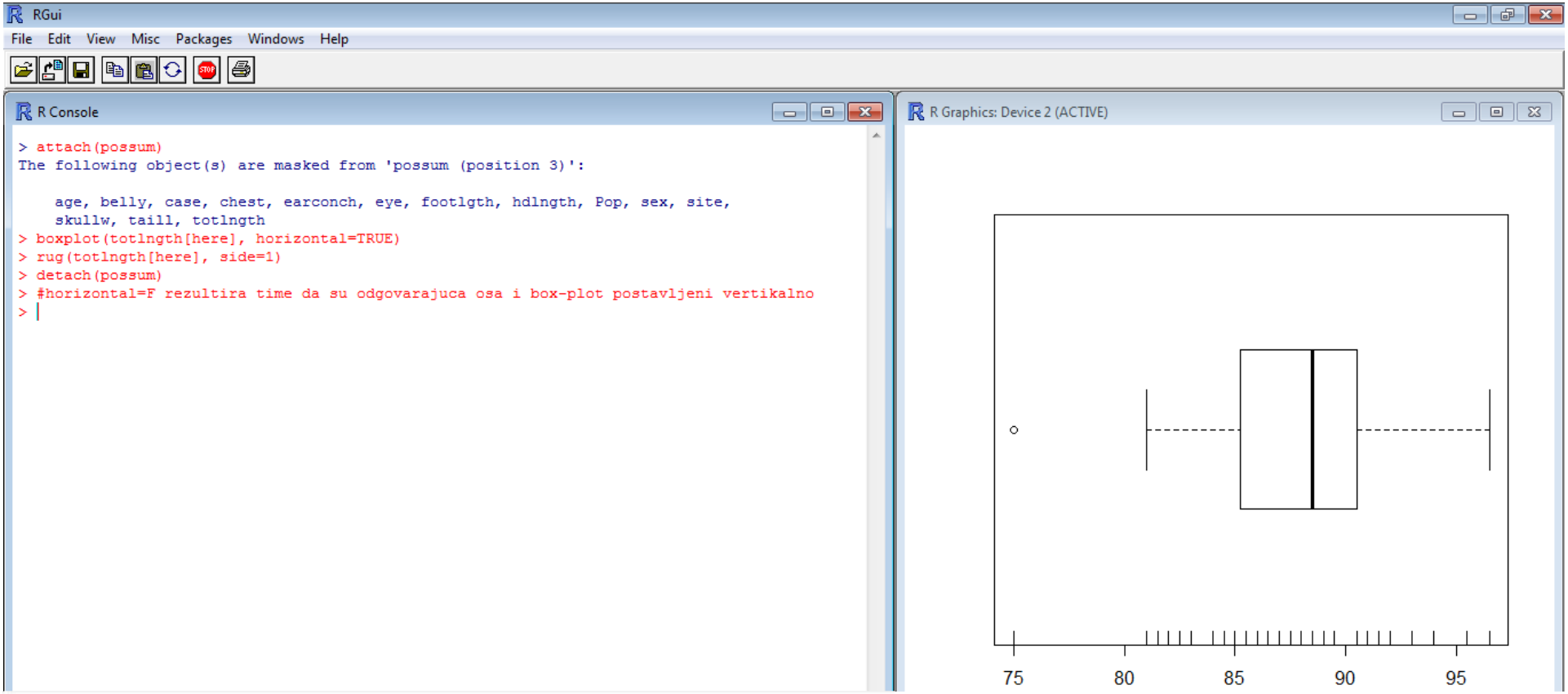
RGui - [R Console]

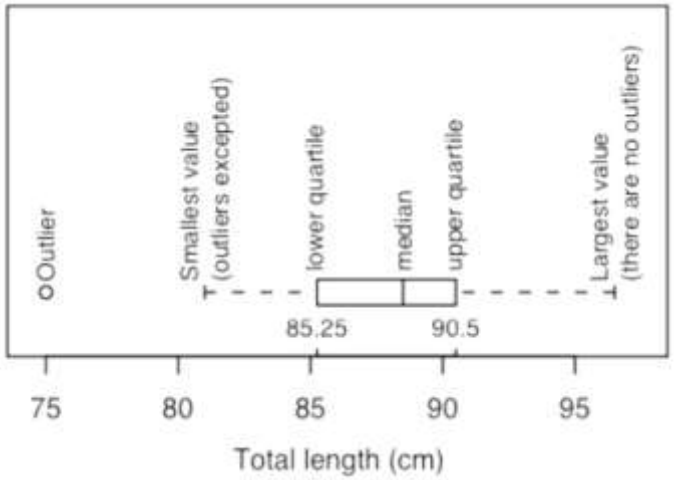
File Edit View Misc Packages Windows Help

```
> library(DAAG)
> data(possum)
> attach(possum)
> here <- sex == "f"
> dens <- density(totlngth[here])
> dens

Call:
        density.default(x = totlngth[here])

Data: totlngth[here] (43 obs.); Bandwidth 'bw' = 1.662

       x                y
 Min.   : 70.01   Min.   :6.275e-05
 1st Qu.: 77.88   1st Qu.:3.684e-03
 Median : 85.75   Median :1.641e-02
 Mean   : 85.75   Mean   :3.174e-02
 3rd Qu.: 93.62   3rd Qu.:5.544e-02
 Max.   :101.49   Max.   :1.039e-01
> xlim <- range(dens$x)
> ylim <- range(dens$y)
> totlngth[here]
 [1] 91.5 95.5 92.0 85.5 90.5 91.0 91.5 89.5 89.5 92.0 89.5 90.5 89.0 96.5 89.0 85.0 88.0 84.0 94.0 82.5 75.0 84.5 83.0 81.0 88.0 85.0 91.0 93.0 88.0 90.5 88.5 89.5
[33] 88.5 86.0 87.0 83.0 82.0 81.5 86.5 87.5 86.5 89.0 89.0
> hist(totlngth[here], breaks = 72.5 + (0:5) * 5, xlab="Total length", main="A: Breaks at 72.5, 77.5, ...")
> hist(totlngth[here], breaks = 72.5 + (0:5) * 5, probability=T, xlim=xlim, ylim=ylim, xlab="Total length", main="B: Breaks at 72.5, 77.5, ...")
> lines(dens)
> detach(possum)
>
```









Inter–quartile range
= 90.5 – 85.25
= 5.25

Compare
0.75 x Inter–quartile range
= 3.9
with standard deviation
= 4.2

Figure 10: Boxplot of female possum lengths, with additional labelling information.

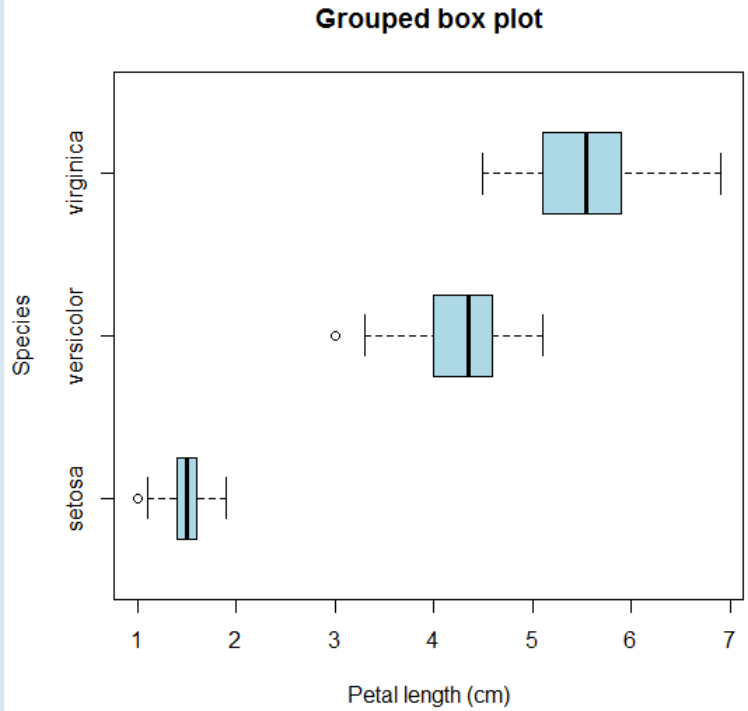
RGui

File Edit View Misc Packages Windows Help

R Console

```
> attach(iris)
> boxplot(Petal.Length ~ Species, horizontal=T,
+ col="lightblue", boxwex=.5,
+ xlab="Petal length (cm)", ylab="Species",
+ main="Grouped box plot")
>
```

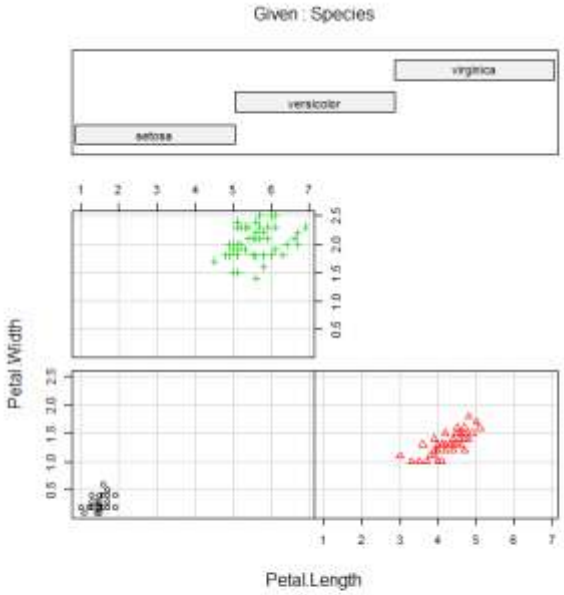
R Graphics: Device 2 (ACTIVE)

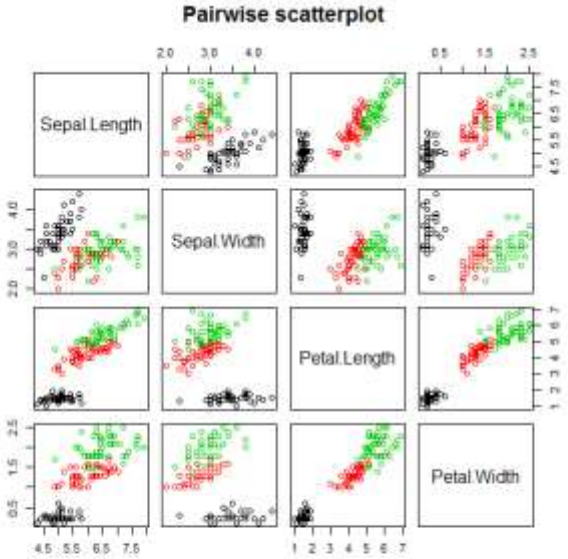


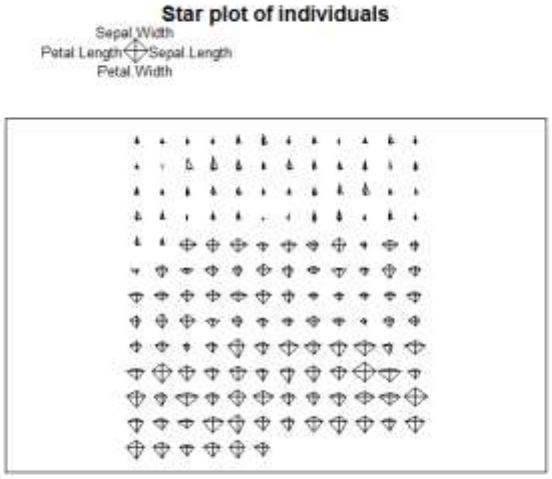
RGui - [R Console]

File Edit View Misc Packages Windows Help

```
> coplot(Petal.Width ~ Petal.Length | Species,
+ col=as.numeric(Species), pch=as.numeric(Species))
> #prethodnom f-jom kreiran je conditioning plot
>
> pairs(iris[,1:4], col=as.numeric(Species),
+ main="Pairwise scatterplot")
> #prethodnom f-jom kreiran je pairwise scatterplot
>
> stars(iris[,1:4], key.loc=c(2,35), mar=c(2, 2, 10, 2),
+ main="Star plot of individuals", frame=T)
> #prethodnom f-jom kreiran je star plot
>
```



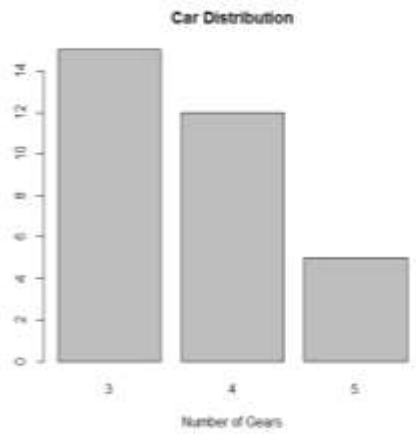


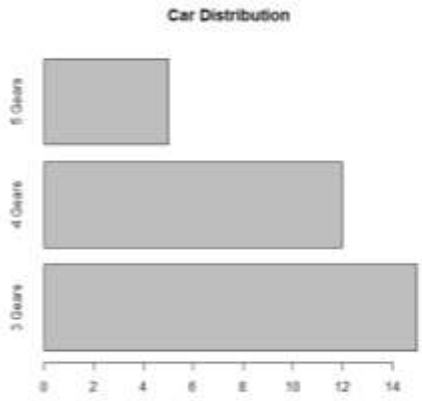


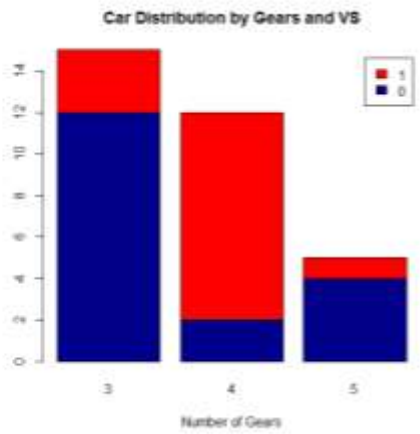
RGui - [R Console]

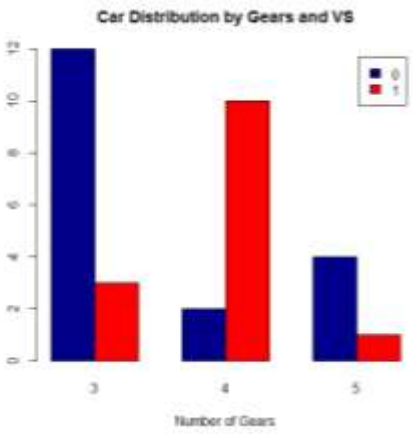
File Edit View Misc Packages Windows Help

```
> #najjednostavniji bar-plot
> counts <- table(mtcars$gear)
> counts

 3  4  5
15 12  5
> barplot(counts, main="Car Distribution",xlab="Number of Gears")
>
> #jednostavan horizontalan bar-plot sa dodatim nazivima barova
> counts <- table(mtcars$gear)
> barplot(counts, main="Car Distribution", horiz=TRUE,
+   names.arg=c("3 Gears", "4 Gears", "5 Gears"))
>
> #'prilepljen' bar-plot u boji i sa legendom
> counts <- table(mtcars$vs, mtcars$gear)
> counts

    3  4  5
  0 12  2  4
  1  3 10  1
> barplot(counts, main="Car Distribution by Gears and VS",xlab="Number of Gears",
+ col=c("darkblue","red"),legend = rownames(counts))
>
> #grupisan bar-plot
> counts <- table(mtcars$vs, mtcars$gear)
> barplot(counts, main="Car Distribution by Gears and VS",xlab="Number of Gears",
+ col=c("darkblue","red"),legend = rownames(counts), beside=TRUE)
>
```





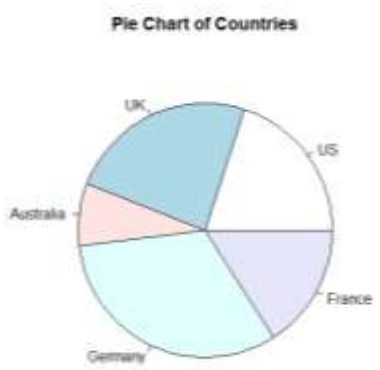


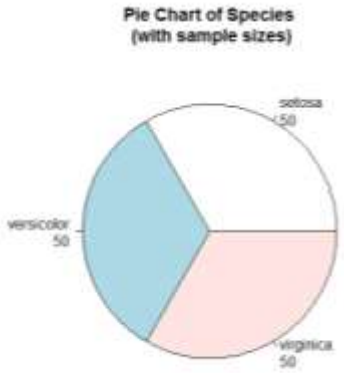


RGui - [R Console]

File Edit View Misc Packages Windows Help

```
> #najjednostavniji pie-chart
> slices <- c(10, 12,4, 16, 8)
> lbls <- c("US", "UK", "Australia", "Germany", "France")
> pie(slices, labels = lbls, main="Pie Chart of Countries")
>
> #pie-chart sa naznacenim procentima
> slices <- c(10, 12, 4, 16, 8)
> lbls <- c("US", "UK", "Australia", "Germany", "France")
> pct <- round(slices/sum(slices)*100)
> lbls <- paste(lbls, pct) #pored imena delova pitice ispisuju se i odgovarajuci procenti
> lbls <- paste(lbls,"%",sep="") #dodaje se i znak za %
> pie(slices,labels = lbls, col=rainbow(length(lbls)),main="Pie Chart of Countries")
>
> #pie-chart u 3D
> library(plotrix)
> slices <- c(10, 12, 4, 16, 8)
> lbls <- c("US", "UK", "Australia", "Germany", "France")
> pie3D(slices,labels=lbls,explode=0.1,main="Pie Chart of Countries")
>
> mytable <- table(iris$Species)
> mytable

    setosa versicolor  virginica
        50         50         50
> lbls <- paste(names(mytable), "\n", mytable, sep="")
> pie(mytable, labels = lbls,main="Pie Chart of Species\n (with sample sizes)")
>
```

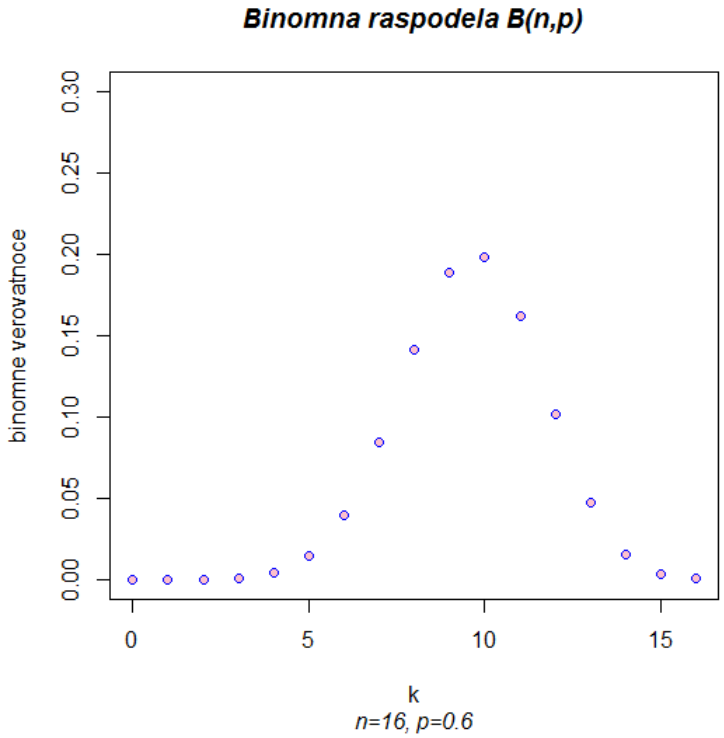


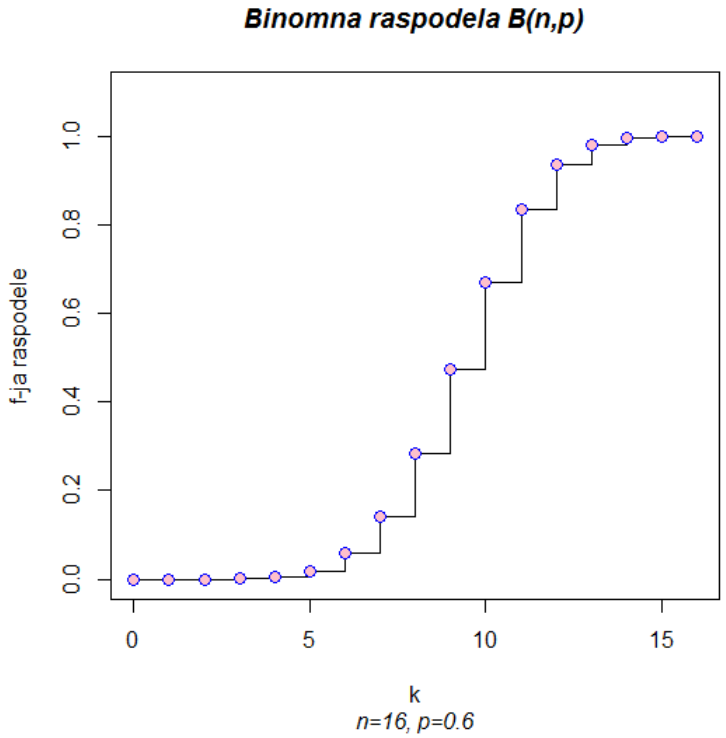


RGui - [R Console]

File Edit View Misc Packages Windows Help

```
> size<-16
> prob<-0.6
> k<-0:size
> p<-dbinom(k,size,prob) #generise se vektor koji cuva verovatnoce iz zakona raspodele binomne B(16,0.6) raspodele
> plot(k,p,main="Binomna raspodela B(n,p)",sub="n=16, p=0.6",xlab="k",ylab="binomne verovatnoce",ylim=c(0,0.3),
+ type="p",pch=21,col="blue",bg="pink",font.main=4,font.sub=3,font.lab=1)
>
> plot(k,cumsum(p),main="Binomna raspodela B(n,p)",sub="n=16, p=0.6",xlab="k",ylab="f-ja raspodele",ylim=c(0,1.1),
+ type="s",font.main=4,font.sub=3,font.lab=1)
> points(k,cumsum(p),pch=21,col="blue",bg="pink",cex=1.25)
>
```





Вежбања:

1. Учитати базу података `LakeHuron` и испитати какве податке садржи. Нацртати график одговарајући график, који приказује ниво језера Хјурон (мерен у стопама) по годинама за које постоје забележени подаци. Означити тачке које одговарају најмањем и највећем нивоу језера и дописати поред њих min, max, редом.

2. Учитати базу података `morley` и испитати какве податке садржи. Нацртати хистограм за обележје: брзина светлости. Шта се, на основу хистограма, може закључити о подацима? Има ли аутлајера? Нацртати и бокс-плот.

```
### function to visualise the variability of small random samples
##  required arguments:
##     n : sample size
##  arguments with reasonable defaults:
##     rows,cols : dimensions of display
##     mu, sd : mean, s.d. of normal distribution to sample
##     bsd : histogram bins to represent each s.d.
##     sdd : +/- number of s.d. to display
plot.normals <- function(n, rows=2, cols=2, mu=0, sd=1,
                         bsd = 2, sdd=3.5) {
                              # set up graphic display
par(mfrow=c(rows, cols))
                              # number of random samples
reps <- rows*cols
                              # histogram bin width
bin.width=sd/bsd
                              # scale x-axis
x.min <- mu-(sdd*sd); x.max <- mu+(sdd*sd)
                              # scale y-axis; max. dnorm(1,0)=0.3989
                              #  adjust to sample and bin sizes
                              #  and normalize by s.d.
                              #  and leave room for higher bars
y.max <- n*0.5*bin.width/sd
                              # compute and display each graph
for (i in 1:reps) {
  v <- rnorm(n, mu, sd)
  hist(v, xlim=c(x.min,x.max), ylim=c(0,y.max),
       breaks = seq(mu-5*sd, mu+5*sd, by=bin.width),
       main=paste("mu =", mu, ", sigma =", sd),
       xlab=paste("Sample",i), col="lightblue", border="gray",
       freq=TRUE)
  x <- seq(x.min,x.max,length=120)
                              # true normal distribution
  points(x,dnorm(x, mu, sd)*(n*bin.width),
         type="l", col="blue", lty=1, lwd=1.8)
                              # normal dist. estimated from sample
  points(x,dnorm(x, mean(v), sd(v))*(n*bin.width),
         type="l", col="red", lty=2, lwd=1.8)
                              # print sample params.
                              # and Pr(Type I error)
  text(x.min, 0.9*y.max, paste("mean:", round(mean(v),2)),pos=4)
  text(x.min, 0.8*y.max, paste("sdev:", round(sd(v),2)),pos=4)
  text(x.min, 0.7*y.max,
       paste("Pr(t):", round((t.test(v, mu=mu))$p.value,2)),pos=4)
}
                              # clean up
par(mfrow=c(1,1))
}
```

3. Написати ф-ју која генерише n (нпр. по default-у $n = 1000$) сл. бројева из дискретне униформне расподеле, којој се може мењати параметар/параметри. Затим се, у прозору за цртање добија хистограм релативних фреквенција елемената тог узорка и истовремено, са његове десне стране, график ф-је расподеле те униформне расподеле.

4. Проучити код ф-је:

и извршити позиве ф-је:

```
> plot.normals(60)
> plot.normals(60, mu=100, sd=15)
> plot.normals(60, rows=3, cols=3, mu=100, sd=15)
```