Ch-5

Graphs and Charts

Bar plots in R (Input can be vector or matrix)

- Simple Bar plot
- Horizontal Bar plot
- Stacked Bar plot
- Grouped bar plot

R Bar plot

* Created by using → barplot() function
* Inputs can be vector / matrix
* If we supply a vector, the plot will have bars with their height equal to the elements in the vector / matrix.

Eg :- temp = c(27, 26, 23, 24, 30)

barplot (temp)

Argument used

- main - used to give heading
  xlab - x-axis name
  ylab -> y axis name
  col - give colour to bar.
  horiz -> TRUE
  names.args -> name of each bar

Eg - temp = c(
  barplot (temp,
  main = "Max Temp in a wee
  xlab = "Degree celsius
  ylab = "Dry"
  col = "blue")

- density - give lines inside bars
- border - border to bars
- density = 20, border = "red", col = "green
- width -> size of bars (by default = 1)
- space -> space b/w bars.

> x ← c(1,1,2,2,2,3,3,1,1,2,2,3,
4,4,4)

table(x)

x	1	2	3	4
	4	5	3	3

Plotting of categorical data

> x ← c(1,1,2,2,2,3,3,3,1,1,2,2,
3,4,4,4)

y = table(x)
barplot(height = y, width =
c(3,4,5,6))

> x ← c(1,1,2,2,2,3,3,1,1,2,2,3,
4,4,4)

y = table(x)
barplot(height = y, space = 5)

> x ← c(1,1,2,2,2,3,3,1,1,2,2,
3,4,4,4)

y = table(x)
barplot(height = y, names.arg
= LETTERS[1:4

barplot(height = y, names.arg
= c("Student 1", "Student 2",
"Student 3", "Student 4"))

> x = c(1,1,1,1,2,1,2 2, 2, 3, 3, 3,
2, 2, 3, 3

y = table(x)
barplot(height = y, names.arg
= c("Student 1", "Student 2",
"Student 3"), legend.text = T)

legend.text is a vector of text
used to construct a .legend
for the plot, i.e., used to identify
what each bar represent.

> x = c(1,1,1,1,2,1,2,2,2,3,3,3,1,1
2,2,33)

y = table(x)
barplot(height = y, las = 1)
barplot(height =

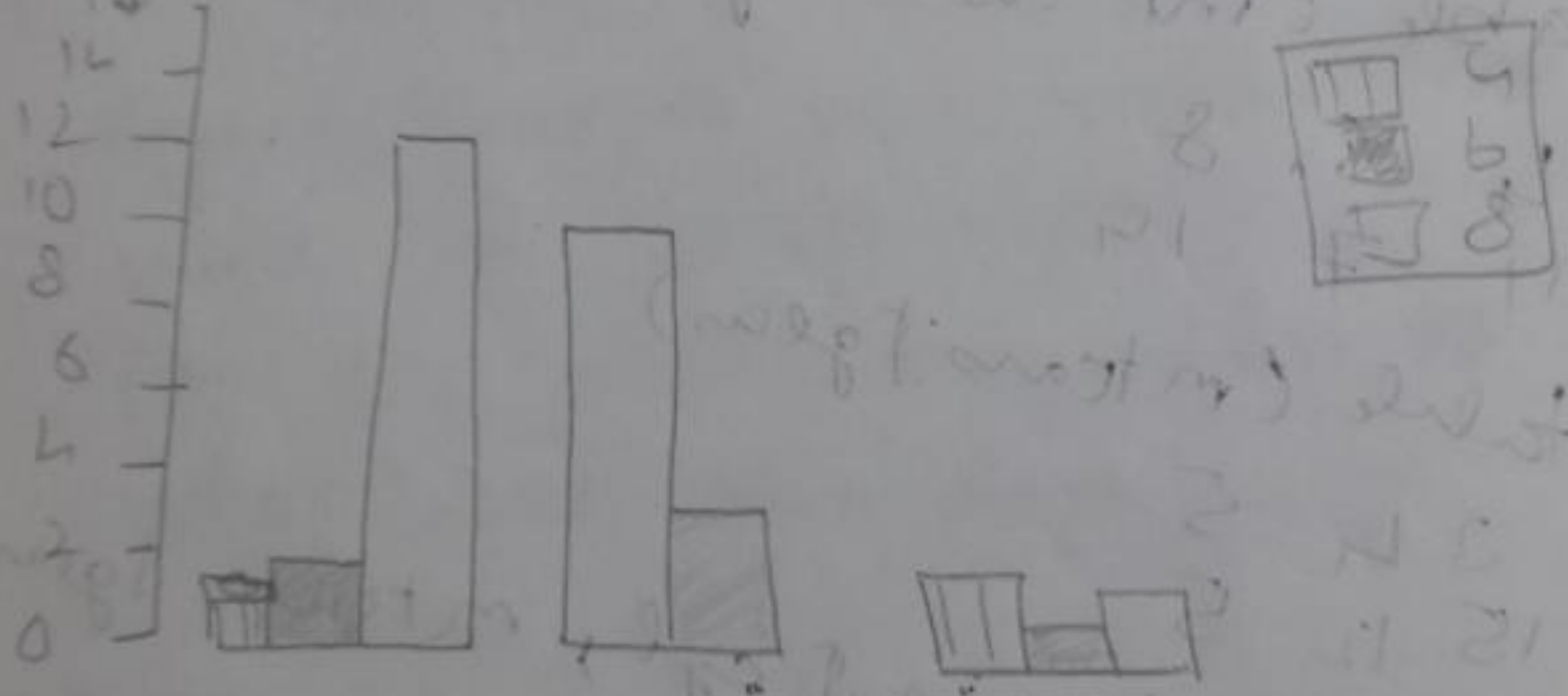
Stacked Bar plots

The plot drawn when matrix is given as input

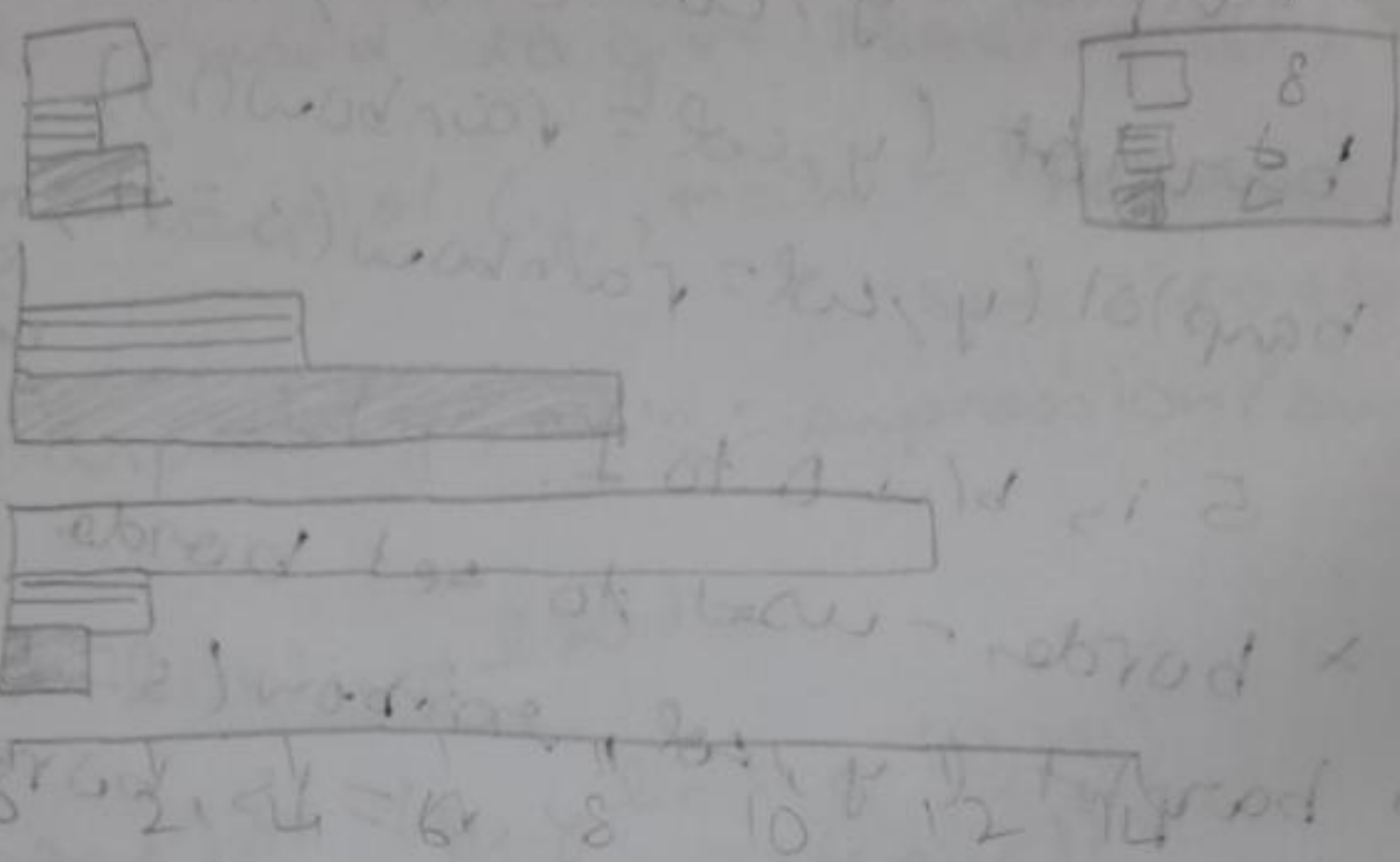
```
> data(mtcars)
> names(mtcars)
[1] "mpg"  "cyl"  "disp" "hp"   "drat"
"wt"   "qsec" "vs"   "am"   "gear"
"carb"
> mtcars$cyl
[1] ...
> table(mtcars$cyl)

 4  6  8
11  7 14
> table(mtcars$gear)

 3  4  5
15 12  5
> table(mtcars$cyl, mtcars$gear)

   3  4  5
4  1  8  2
6  2  4  1
8 12  0  2
```

y1 = table(mtcars$cyl, mtcars$gear)
barplot(y1, gert.text=T)

> barplot(y1, legend.text=T, beside=T)



barplots(y1, legend.text=T, beside=T, horiz=T)



* density: used to give lines inside bars

Eg - x = c(1,1,1,2,2,2,3,3,3,3)

y = table(x)

barplot(y, legend.text=T, las=1, density=c(5,10,15))

* Angle: used to give angle to lines inside bars
* colours: give colours to bars
* barplot(y, col="red")

par(mfrows=c(1,1))
> barplot(y, col=c(1,2,3))
barplot(y, col=rainbow(1))
barplot(y, col=rainbow(s=.2, n=2
                        &
                        no. of col

5 is b/w 0 to 1.

* border - used to set borde to ba

> barplot(y, col, rainbow(s=.5,
                    n=15, borde=

> par(mf rows=c(1,2)
> barplot(y, col=rainbow(s=5,
                    n=13, border=F)
> barplot(y, col=rainbow(s=5
                    n=13, border=T)
> ~~barplot(y col=~~
> ba(mf rows=c(1,1)

* main : used to give heading to to the particula bar plot.

sub - used to give heading at bottom

Eg - barplot (y, main = "heade",
sub = "foote")

barplot (y; main = expression(sum()))

* xlimit, ylimit

barplot (y, ylim = c(0, 10))
barplot (y, xlim = c(0, 5))

Piechart

Diagramatic representation of values

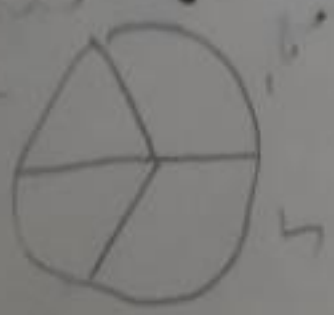
x = c(1, 1, 1, 2, 2, 3, 3, 4, 4, 4)

pie(x)

x = c(1, 1, 2, 2, 3, 3, 4, 4, 4)

y = table(x)

pie(y)



pie (y, main = "my first plot")

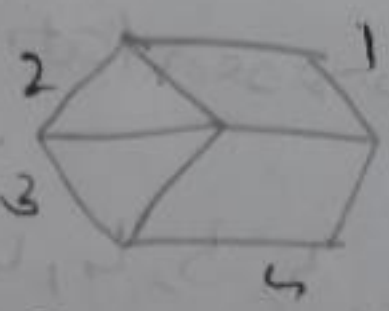
x = a vector of non negative numerical quantities)

eg1 :- pie (y, ~~main~~ labels = LETTERS[1:4])

Eg: pie (y, labels = c("red", blue, green, orange)

* labels give name of each slices)

Pie (y, edges = 10)



→ pie (y, radius = .5)

* pie (y, clockwise = T)

* pie (y, density = c(10, 20, 30, 40)

Density = used to give shading to each slide)

colour = F[101]

* pie(y, col = rainbow(15))
  pie(y, col = 1:h)

Border:

Used to set border it can either T or F

pie(y, col = 1:h, border = F)

Histogram

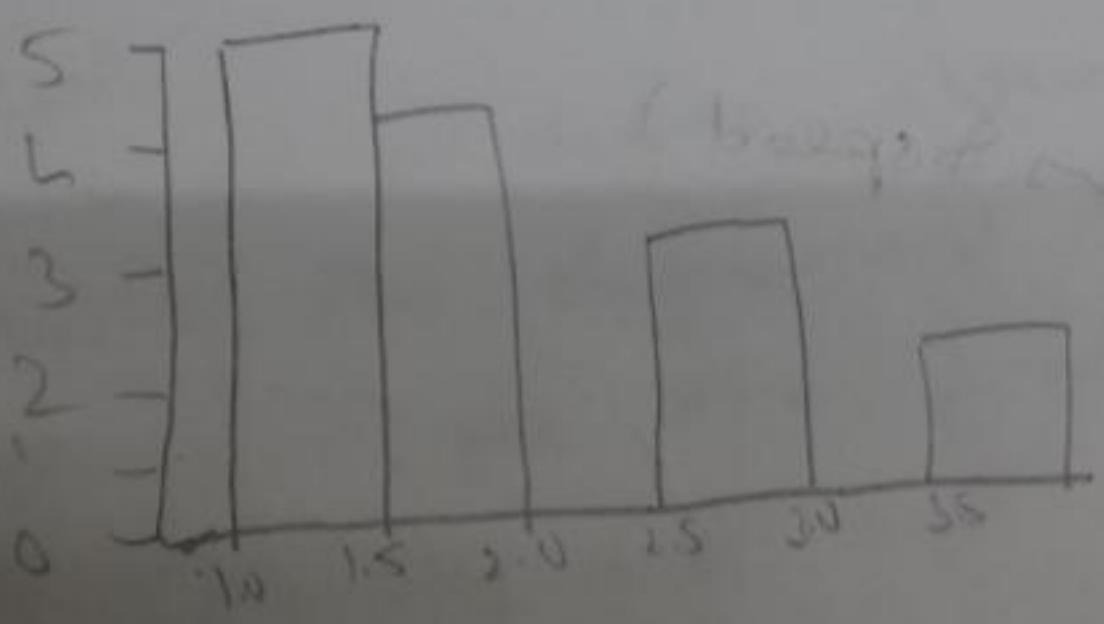
used to plot quantitative data

function : Hist()

inputs are vector inputs

x = c(1,1,1,1,1,2,2,2,2,3,3,3,4,4)

Hist(x)



To viewing the grouping arrangement use the function, cut()

```
> cut(x,6)
> dataframe(x, cut(x,6))
  data("cars")
> head(cars)
```

	speed	dis
1	4	2
2	4	10
3	7	4
4	7	22
5	8	16
6	9	10

```
> cars$speed  # To display data of speed
```

[1]

```
> hist(cars$speed)
```

Arguments

* breaks:
  > hist(cars$speed, breaks=20)
* main :- used to give title
> hist(cars$speed, xlab="dist",
ylab="No of times")

Eg :- > airquality
> head(airquality).
> temp = airquality$Temp
> hist(temp
> str(airquality)
str - used to display structure

* xlim, ylim:
used to provide range of values
* col:
used to define colour.
* With the argument freq=FALSE we can get the frequency probability distribution instead of probability

Return values of hist()

Display the value in hist()

* breaks - place where the breaks occurs
* counts - The no. of observation falling in that cell.
* mids - The midpoint of cells.
* xname - The x argument name
* equidist - A logical value indicating if the bar breaks are equally spaced or not

Eg - h = hist(temp)

> h

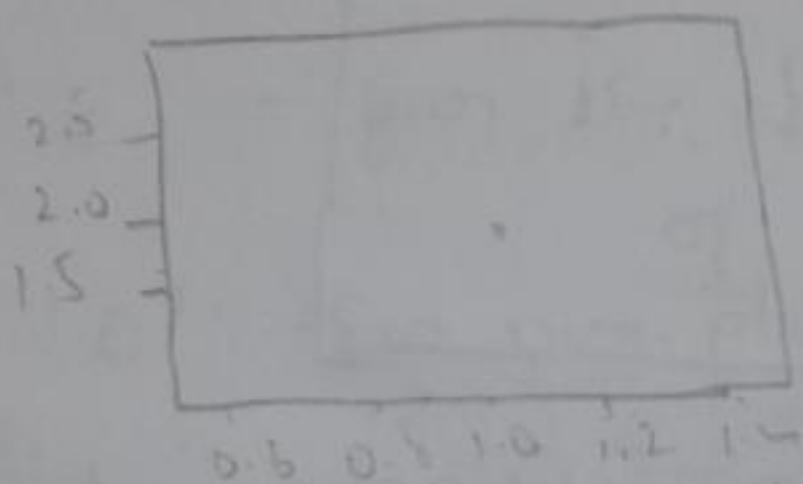
* With the breaks argument we can specify the no. of cells we want in the histogram.
* We can also define breakpoints b/w the cells as a vector, this makes it possible to plot a histogram with unequal intervals

hist (temp, border = "blue"; breaks
= c(55, 60, 70, 75, 80, 100)

scatter plot

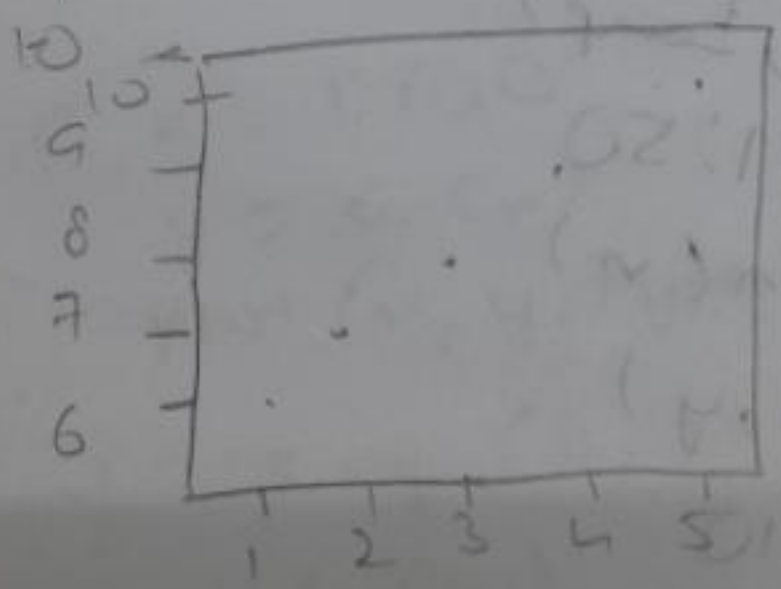
created by using plot() function

Eg - plot(x)



Eg - plot(c(5,6,7,8))

x = 1:5
y = 6:10
plot(x, y)



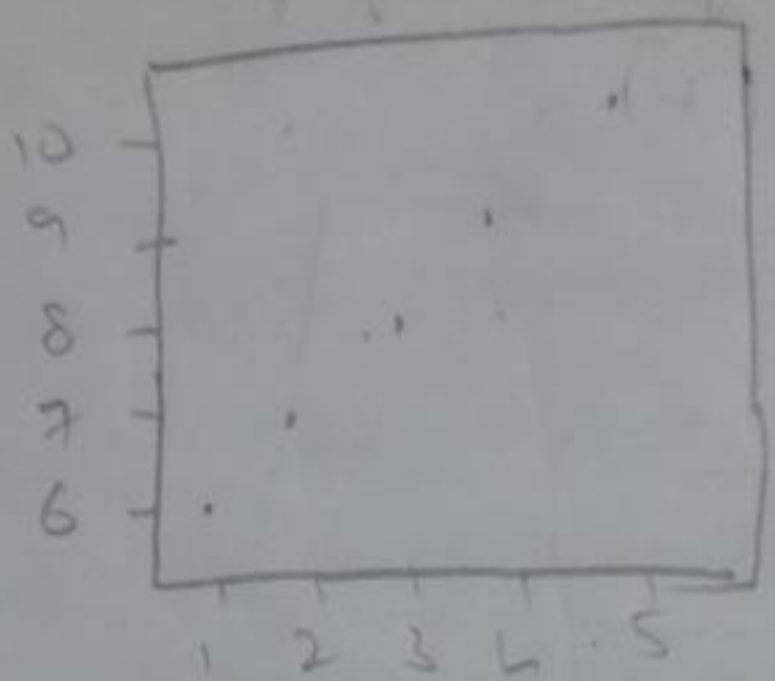
plot(c(5,6,7,8))

x = 1:5

y = 6:10

plot(x,y)

plot(x,y)



Eg - airquality

> head(air, quality)

> day = airquality $day

> temp - airquality $temp

> plot (day, temp)

Eg ⇒ x1 = 1:50

> y = sin(x)

> plot(x, y)

Eg - x = 1:10

y = 21:30

plot (x, y, main = "scatter plot", xlab = "x-values", ylab = "y values", col = 1:10)

Type

Specifies what type of plot should be drawn possible types are

"p" -> for points
"l" -> for lines
"b" - for both, i.e combination of points and lines
"c" - for the lines part alone of b(i.e, dashed lines.
"o" - for overplotted
"h" - for histogram
"s" - for scala stair
"n" -> no plotting

Eg - plot(x, y, main = "scatter plot", type = "c")

x = 1:100
y = sin(x)
plot(x, y, type = "l")

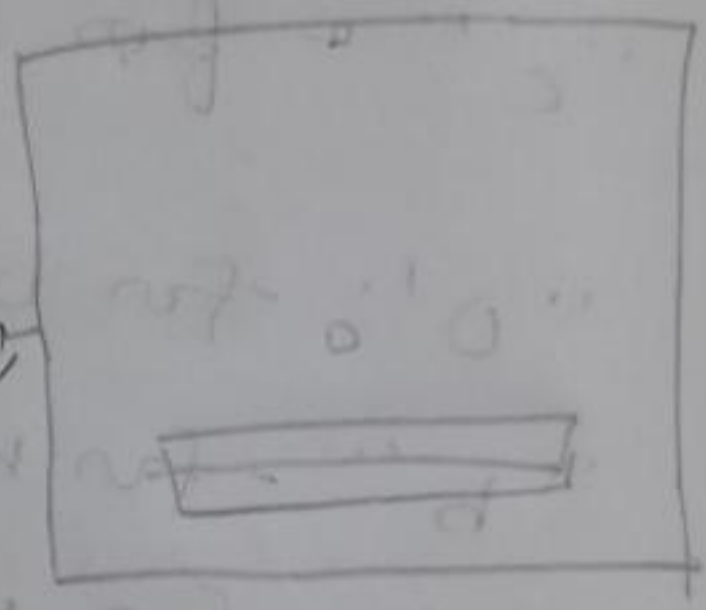
x = seq(0, 10, 0, 0.1)
y = sin(x)
plot(x, y)

Box plot :

Used to plot quantitative data

Eg :-

```
v = c(1, 1, 1, 2, 2, 1, 1, 3, 3, 3, 4, 4, 5,
  7, 4, 4, 6, 5, 7, 209, 20, 20, 25, 25,
                              45, 200)
boxplot(v)
```



> Box plot can be used to identify median, range, quartile deviation and various other statistical measures.

```
> str(airquality)
> boxplot(airquality$Ozone)
> boxplot(airquality$Ozone, main="
in parts per billion from 'BOJ 70 1520
hrs at Roosevelt island', xlab = "parts
per billion"; ylab = "Ozone", col="orang
  rotl=T, horizontal = T)
```

For drawing multiple boxplots

```
> ozone = airquality $ ozo
> temp = airquality $ temp
> wind = airquality $ wind
> boxplot( ozone, temp, wind )
```

For changing the width of bars

```
> boxplot (ozone, width = 1, borde = "red")
```