**Numerical Summary of the Data**

**Accessing the Data**

There are several ways to extract data from a vector. Here is a summary using both slicing and extraction by a logical vector. Suppose x is the data vector say >x=1:15

how many elements? length(x)

ith element x[4] (i = 4)

all but ith element x[-3] (i = 3)

First k elements x[1:7] (k = 7)

last k elements x[(length(x)-5):length(x)] (k = 5)

specific elements. x[c(1,3,5)] (First, 3rd and 5th)

all greater than some value x[x>3] (the value is 3)

bigger than or less than some values x[ x< -2 | x > 2]

which indices are largest which(x == max(x))

|  |  |  |
| --- | --- | --- |
| **Entry** | **Package** | **Description** |
| min() | base | Calculates the minimum value of the numeric vector |
| max() | base | Calculates the maximum value of the numeric vector |
| range() | base | Calculates the range of the numeric vector |
| pmin() | base | Calculates the parallel minima of two or more numeric vectors or matrices |
| pmax() | base | Calculates the parallel maxima of two or more numeric vectors or matrices |
| quantile() | stats | Calculates various sample quantiles of a numeric vector |
| IQR() | stats | Calculates the inter-quartile range of a numeric vector |
| fivenum() | stats | Calculates Tukey's five-number summary |
| median() | stats | Calculates the median of a numeric vector |
| mean() | base | Calculates the arithmetic mean of the numeric vector |
| weighted.mean() | stats | Calculates the weighted mean of a numeric vector |
| mad() | stats | Calculates the median absolute difference of a numeric vector |
| sd() | stats | Calculates the standard deviation of a numeric vector |
| var() | stats | Calculates the variance of a numeric vector |
| rank() | base | Calculates the sample ranks of the values of a vector |
| smean.sd() | Hmisc | Calculates the mean and standard deviation of a numeric vector |
| wtd.mean() | Hmisc | Calculates the weighted mean of a numeric vector |
| wtd.var() | Hmisc | Calculates the weighted variance of a numeric vector |
| wtd.quantile | Hmisc | Calculates the weighted quantiles of a numeric vector |
| wtd.ecdf | Hmisc | Calculates the weighted ECDF of a numeric vector |
| wtd.rank | Hmisc | Calculates the weighted ranks of a numeric vector, using mid-ranks  for ties |
| describe | Hmisc | Provides a concise statistical description of a vector, matrix, or  data frame |
| *cor()* | stats | Calculates the correlation between two numeric vectors or matrices |
| cov() | stats | Calculates the covariance between two numeric vectors or matrices |
| var() | stats | Calculates the variance between two numeric vectors or matrices |
| cov2cor() | stats | Scales a covariance matrix into the corresponding correlation matrix |
| density() | stats | Calculates the kernel density estimates of a numeric vector |
| ecdf() | stats | Calculates the empirical cumulative distribution function (ECDF)  of a numeric vector |

**Examples**

Suppose, CEO yearly compensations are sampled and the following are found (in millions).

12, 0.4, 5, 2, 50, 8, 3, 1, 4, 0.25

> sals = scan() # read in with scan

1: 12 .4 5 2 50 8 3 1 4 0.25

11:

Read 10 items

> mean(sals) # the average

[1] 8.565

> var(sals) # the variance

[1] 225.5145

> sd(sals) # the standard deviation

[1] 15.01714

> median(sals) # the median

[1] 3.5

> fivenum(sals) # min,lower hinge, Median, upper hinge, max

[1] 0.25 1.00 3.50 8.00 50.00

> summary(sals)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.250 1.250 3.500 8.565 7.250 50.000

> mean(sals,trim=1/10) # This computes the 10%trimmed mean

[1] 4.425

> IQR(sals)

[1] 6

**Example**

> data=c(10, 17, 18, 25, 28, 28)

> summary(data)

Min. 1st Qu. Median Mean 3rd Qu. Max.

10.00 17.25 21.50 21.00 27.25 28.00

> quantile(data,.25)

25%

17.25

> quantile(data,c(.25,.75)) # two values of p at once

25% 75%

17.25 27.25

**Example:**

> x<-c(2,4,5,6,4,5,6,7,8,9,12,23)

> y<-c(5,1,3,5,6,7,8,9,3,21,13,21)

> pmin(x,y)

[1] 2 1 3 5 4 5 6 7 3 9 12 21

> pmax(x,y)

[1] 5 4 5 6 6 7 8 9 8 21 13 23

> range(x)

[1] 2 23

> smean.sd(x)

Error: could not find function "smean.sd"

> library(Hmisc)

> x

[1] 2 4 5 6 4 5 6 7 8 9 12 23

> smean.sd(x)

Mean SD

7.583333 5.517877

**Boxplot**

A boxplot is a way of summarizing a set of data measured on an interval scale. It is often used in exploratory data analysis. It is a type of graph which is used to show the shape of the distribution, its central value, and variability. The picture produced consist five number summaries.

The median for each dataset is indicated by center line, and the first and third quartiles are the edges of the box. The extreme values (within 1.5 times the inter-quartile range from the upper or lower quartile) are the ends of the lines extending from the IQR. Points at a greater distance from the median than 1.5 times the IQR are plotted individually as asterisks. These points represent potential outliers.

Example: Test score of 15 students in a class is given below

>x=c(24,58,61,67,71,73,76,79,82,83,85,87,88,88,92,93,94,97)

>boxplot(x, main="Boxplot of test scores", col=2)

> arrows(1,24,1.2,30)

> text(1.4,31,"This is an Outlier")



*>temperature="http://biostatistics.it/Didattica/Dati/SilwoodWeather.txt"*

>weather=read.table(temperature,header=T)

>attach(weather)

> names(weather)

[1] "upper" "lower" "rain" "month" "yr"

Before we can plot the data we need to declare month to be a factor. At the moment, R just thinks it is a number.

>month<-factor(month)

Now we can plot using a categorical explanatory variable (month) and, because the first variable is a factor, we get a boxplot rather than a scatterplot:

>plot(month,upper)

Please note that it is also possible to draw side by side box plot just using

>boxplot(upper~month) # In this case no need to change to factor



**Displaying Both Histogram and Boxplot**

The function ***simple.hist.and.boxplot*** will plot both a histogram and a boxplot to show the relationship between the two graphs for the same dataset.

Note that we need to have ***UsingR*** package downloaded before we implement this command.

> library(UsingR)

> x<-rnorm(100)

> simple.hist.and.boxplot(x)

****

**Recall**

* To list all available packages Use the command ***library().***
* To list all available datasets Use the command **data()**
* To list all data sets in a given package Use **data(package='package name')**

for example **data(package="UsingR")**

* To read in a dataset Use **data("Dataset")**