# Measure of Central Tendency

## Mean

It is calculated by taking the sum of the values and dividing with the number of values in a data series.

The function **mean()** is used to calculate this in R.

**Syntax:** mean(x, trim=0, na.rm=FALSE)

Following is the description of the parameters used

* **x** is the input vector.
* **trim** is used to drop some observations from both end of the sorted vector.
* **na.rm** is used to remove the missing values from the input vector.

**Example**

# Create a vector

x <- c(12,7,3,4.2,18,2,54,-21,8,-5)

# Find Mean

result.mean <- mean(x)

print(result.mean)

### Applying Trim Option

When trim parameter is supplied, the values in the vector get sorted and then the required numbers of observations are dropped from calculating the mean.

When trim = 0.3, 3 values from each end will be dropped from the calculations to find mean.

In this case the sorted vector is (−21, −5, 2, 3, 4.2, 7, 8, 12, 18, 54) and the values removed from the vector for calculating mean are (−21,−5,2) from left and (12,18,54) from right.

**Example**

# Create a vector.

x <- c(12,7,3,4.2,18,2,54,-21,8,-5)

# Find Mean.

result.mean <- mean(x, trim = 0.3)

print(result.mean)

### Applying NA Option

If there are missing values, then the mean function returns NA.

To drop the missing values from the calculation use na.rm = TRUE. Which means remove the NA values.

**Example**

# Create a vector.

x <- c(12,7,3,4.2,18,2,54,-21,8,-5,NA)

# Find mean.

result.mean <- mean(x)

print(result.mean)

# Find mean dropping NA values.

result.mean <- mean(x,na.rm = TRUE)

print(result.mean)

## Mode

#There is no function in base R to find mode of set of numbers

x <- c(8,2,7,1,2,9,8,2,10)

y <- table(x)

names(y)[which(y==max(y))]

names(y)

**Another Way Just using a single line**

x <- c(8,2,7,1,2,9,8,2,10,9)

names(table(x))[table(x)==max(table(x))]

## Median

The middle most value in a data series is called the median. The **median()**function is used in R to calculate this value.

**Syntax=** median(x, na.rm = TRUE)

Following is the description of the parameters used −

* **x** is the input vector.
* **na.rm** is used to remove the missing values from the input vector.

**Example:**

# Create the vector.

x <- c(12,7,3,4.2,18,2,54,-21,8,-5)

# Find the median.

median.result <- median(x)

print(median.result)

## Quartiles

**Syntax:** quantile(data,duration)

where : duration = 0.25 for 25%, 0.5 for 50%, 0.75 for 75%

### For Interquartile Range:

Syntax: IQR(data) #IQR=Q3-Q1

**Example:**

x <- c(12,7,3,4.2,18,2,54,-21,8,-5)

x

quantile(x,0.25)

quantile(x,0.75)

IQR(x)

## Variance and Standard Deviation

# enter data

y=c(445, 530, 540, 510, 570, 530, 545, 545, 505, 535, 450, 500, 520, 460, 430, 520, 520, 430, 535, 535, 475, 545, 420, 495, 485, 570, 480, 495, 470, 490)

var(y) #Variance

sd(y) #Standard Deviation

# Compute Mean Median Mode from Frequency

midx<-seq(147.5,182.5,5)

frequency<-c(4,6,28,58,64,30,5,5)

fr.dist<-data.frame(midx,frequency)

fr.dist

c1<-cumsum(frequency) ## Cumulative Frequency

c1

n<-sum(frequency)

n

m1<-min(which(c1>=n/2)) ###Show the median class

m1

h<-5

cfr.dist<-data.frame(midx,frequency,c1)

cfr.dist ###Cumulative Frequency Distn

l<-midx[m1]-h/2# lower limit of median class

f<-frequency[m1]#Frequency of median class

c<-c1[m1-1] # cumulative frequency of pre median class

c

median<-l+(h/f)\*(n/2-c)

median

##########################################################

MODE

m<-which(frequency==max(frequency)) #frequency of modal class

m

fm<-frequency[m] #frequency of modal class

f1<-frequency[m-1]# Frequency of pre modal class

f2<-frequency[m+1] # Frequency of post modal class

h<-5

l<-midx[m]-h/2

mode<-l+(h)\*((fm-f1)/((fm-f1)+(fm-f2)))

mode

#######################################################################

Mean

mean<-sum(midx\*frequency)/sum(frequency)

mean

# Averages and Dispersion

getwd() # Shows the working directory (wd)

setwd(choose.dir()) # Select the working directory interactively

setwd("E:/FAST University/R Folders/ Codes/ week 2/data")

########

############Download file ‘students.csv’ from the internet.

download.file("http://dss.princeton.edu/training/students.xls",

"C:/test/students.xls",

method="auto",

quiet=FALSE,

mode = "wb",

cacheOK = TRUE)

###############

###############

"http://cran.r-project.org/web/views" # Full list of packages by subject area

##############

##############Import Excel Data####

mydata=read.csv("c:/test/students.csv", header=TRUE)

mydata=read.csv("E:/FAST University/R Folder/Codes/week 2/students.csv", header=TRUE)

mydata

######### You may Edit the mydata###

edit(mydata)

############## Or You may Import by Uisng This######

mydata=read.csv(file.choose(), header = TRUE)

mydata

names(mydata) ######## To check the variables names

mysummary(mydata)

############## Direct From Internet###############

mydata <- read.csv("http://dss.princeton.edu/training/students.csv", header=TRUE)

#############

########################### Import SPSS FIle##

install.packages("foreign") # Need to install package –-foreign–- first (you do this only once)

library(foreign) # Load package –-foreign--

mydata.spss <- read.spss("http://dss.princeton.edu/training/mydata.sav",

to.data.frame = TRUE,

use.value.labels=TRUE,

use.missings = to.data.frame)

##################################

##################################

####################descriptive statistics######################

install.packages("psych")

library(psych)

install.packages("DescTools")

libarary("DescTools")

library(psych)

describe(mtcars$wt)

summary(mtcars$wt) # Provides basic descriptive statistics and frequencies.

edit(mtcars) # Open data editor

str(mtcars) # Provides the structure of the dataset

names(mtcars) # Lists variables in the dataset

head(mtcars) # First 6 rows of dataset

head(mtcars, n=10)# First 10 rows of dataset

head(mtcars, n= -10) # All rows but the last 10

tail(mtcars) # Last 6 rows

tail(mtcars, n=10) # Last 10 rows

tail(mtcars, n= -10) # All rows but the first 10

mtcars[1:10, ] # First 10 rows

mtcars[1:10,1:3] # First 10 rows of data of the first 3 variables

mean(mydata) # Mean of all numeric variables

mean(mydata$SAT)

with(mydata, mean(SAT))

median(mydata$SAT)

var(mydata$SAT) # Variance

sd(mydata$SAT) # Standard deviation

max(mydata$SAT) # Max value

min(mydata$SAT) # Min value

range(mydata$SAT) # Range

quantile(mtcars$wt) # Quantiles 25%

quantile((mtcars$wt), c(.3,.6,.9)) # Customized quantiles

fivenum(mtcars$wt) # Boxplot elements. From help: "Returns Tukey's five number summary (minimum,

boxplot(mtcars$wt)

# lower-hinge, median, upper-hinge, maximum) for the input data ~ boxplot"

length(mydata$SAT) # Num of observations when a variable is specify

length(mtcars) # Number of variables when a dataset is specify

which.max(mtcars$wt) # From help: "Determines the location, i.e., index of the (first) minimum or maximum of a

numeric vector"

which.min(mtcars$wt) # From help: "Determines the location, i.e., index of the (first) minimum or maximum of a

numeric vector”

# Mode by frequencies

table(mtcars$wt)

max(table(mtcars$wt))

names(sort(-table(mtcars$wt)))[1]

####### RUG PLOT########

head(mtcars)

hist(mtcars$hp)

hist(mtcars$wt, col="green")

with(mtcars, hist(wt))

rug(mtcars$wt)

# Histogram of wt with a nicer title.

# Applying Freedman/Diaconis rule p.120 ("Algorithm that chooses bin widths and locations

automatically, based on the sample size and the spread of the data"

http://www.mathworks.com/help/toolbox/stats/bqucg6n.html)

with(mtcars, hist(wt, breaks="FD", col="green"))

box()

hist(mtcars$mpg,breaks=20)

# Conditional histograms

par(mfrow=c(1, 2))

hist(mydata$SAT[mydata$Gender=="Female"], breaks="FD", main="Female", xlab="SAT",col="green")

hist(mydata$SAT[mydata$Gender=="Male"], breaks="FD", main="Male", xlab="SAT", col="green")

# Braces indicate a compound command allowing several commands with 'with' command

par(mfrow=c(1, 1))

with(mtcars, {

hist(wt, breaks="FD", freq=FALSE, col="green")

lines(density(wt), lwd=2)

lines(density(wt, adjust=0.5),lwd=1)

rug(wt)

})

###################################

#######################################

Histograms overlaid

hist(mydata$SAT, breaks="FD", col="green")

hist(mydata$SAT[mydata$Gender=="Male"], breaks="FD", col="gray", add=TRUE)

legend("topright", c("Female","Male"), fill=c("green","gray"))

# Check

satgender=table(mydata$SAT,mydata$Gender)

satgender

#########################

################Curve########################3

x=rnorm(100)

hist(x, freq=F)

curve(dnorm(x), add(T))

h=hist(x, plot=F)

ylim <- range(0. h$density, dnorm(0))

hist(x, freq=F, ylim=ylim)

curve(dnorm(x), add=T)