Data Analysis Using R

Unit –III

Descriptive Statistics in R

Descriptive Statistics - Introduction

- Descriptive statistics is a branch of statistics aiming at summarizing, describing and presenting a series of values or a dataset.
- Descriptive statistics is often the first step and an important part in any statistical analysis.
- It allows to check the quality of the data and it helps to “understand” the data by having a clear overview of it.
- If well presented, descriptive statistics is already a good starting point for further analyses.

Descriptive Statistics - Introduction

- There exists many measures to summarize a dataset. They are divided into two types:

1. location measures and
2. dispersion measures

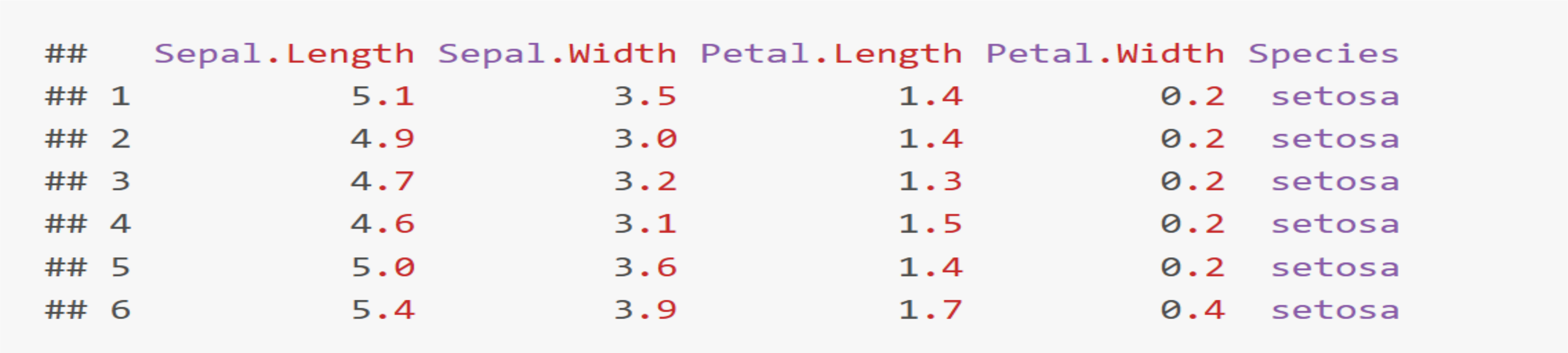
- Location measures give an understanding about the central tendency of the data
- Dispersion measures give an understanding about the spread of the data – measure of variability.

Data - Iris Dataset

- Dataset is imported by default in R, you only need to load it by running iris

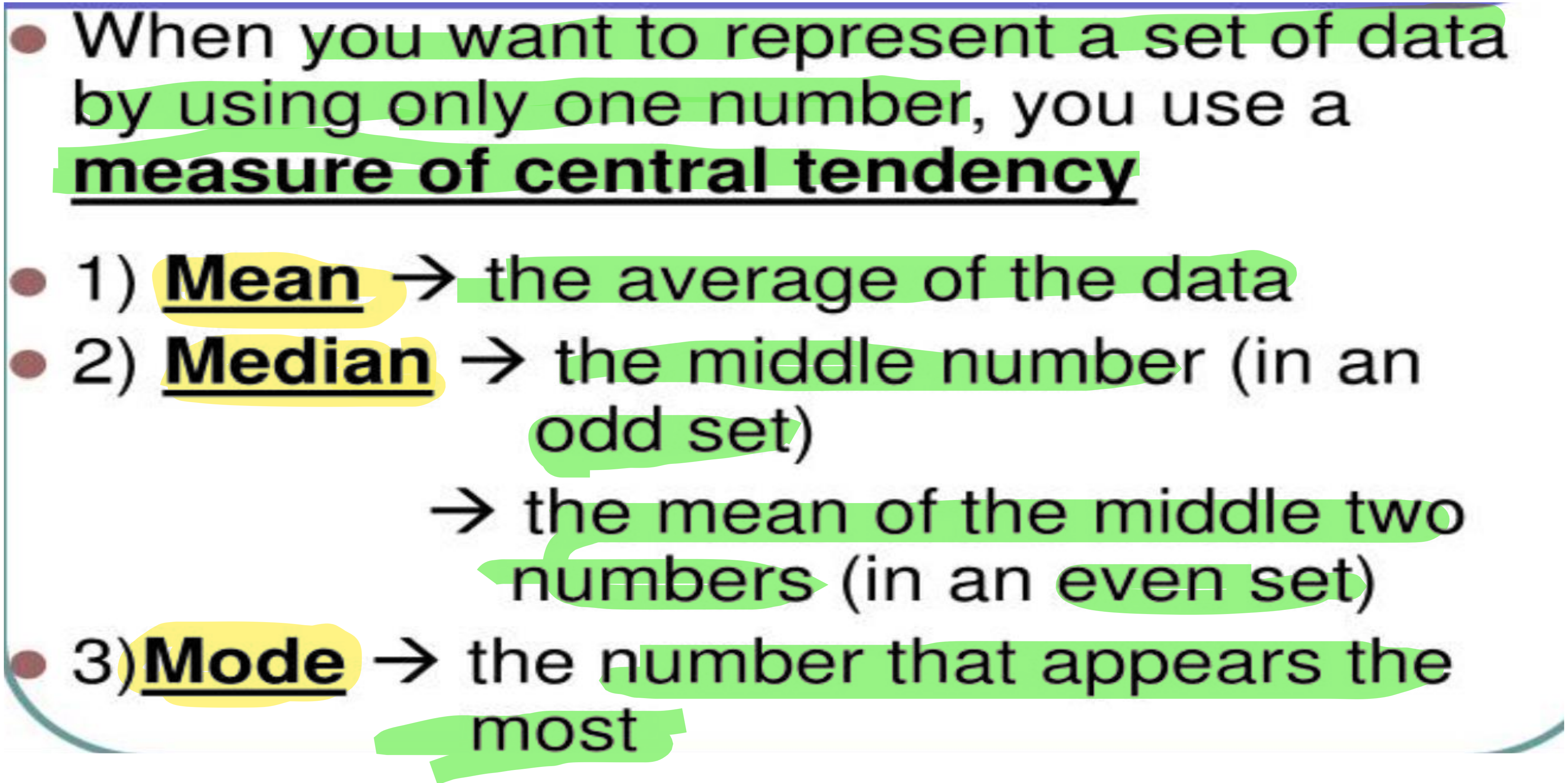
```
dat <- iris # load the iris dataset and renamed it dat
```

```
head(dat) # first 6 observations
```



```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Measures of Central Tendency



- When you want to represent a set of data by using only one number, you use a **measure of central tendency**
- 1) **Mean** → the average of the data
- 2) **Median** → the middle number (in an odd set)

  → the mean of the middle two numbers (in an even set)
- 3)**Mode** → the number that appears the most

Mean → Measure of Central Tendency

The mean is the average value of a set of data points. In R, the mean() function can be used to calculate the mean.

```
mean(dat$Sepal.Length)
```

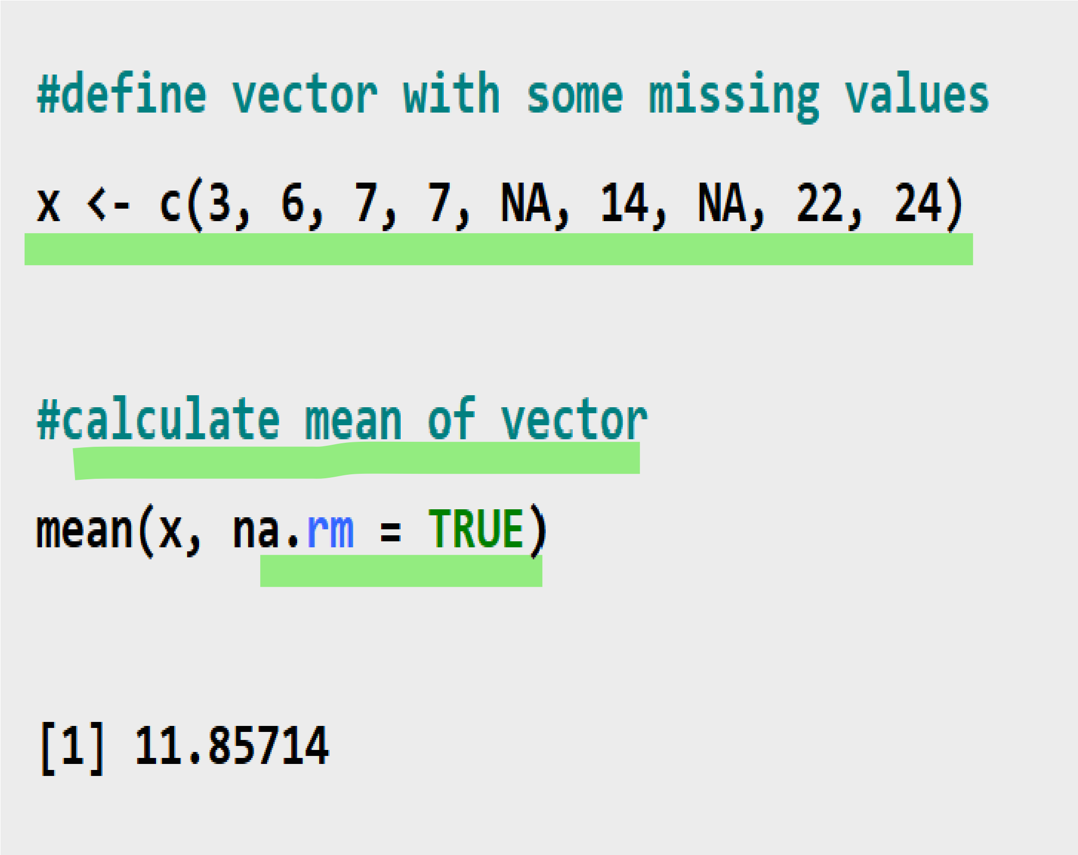
Tips:

- If there is at least one missing value in your dataset, use mean(dat$Sepal.Length,na.rm=TRUE) to compute the mean with the NA excluded

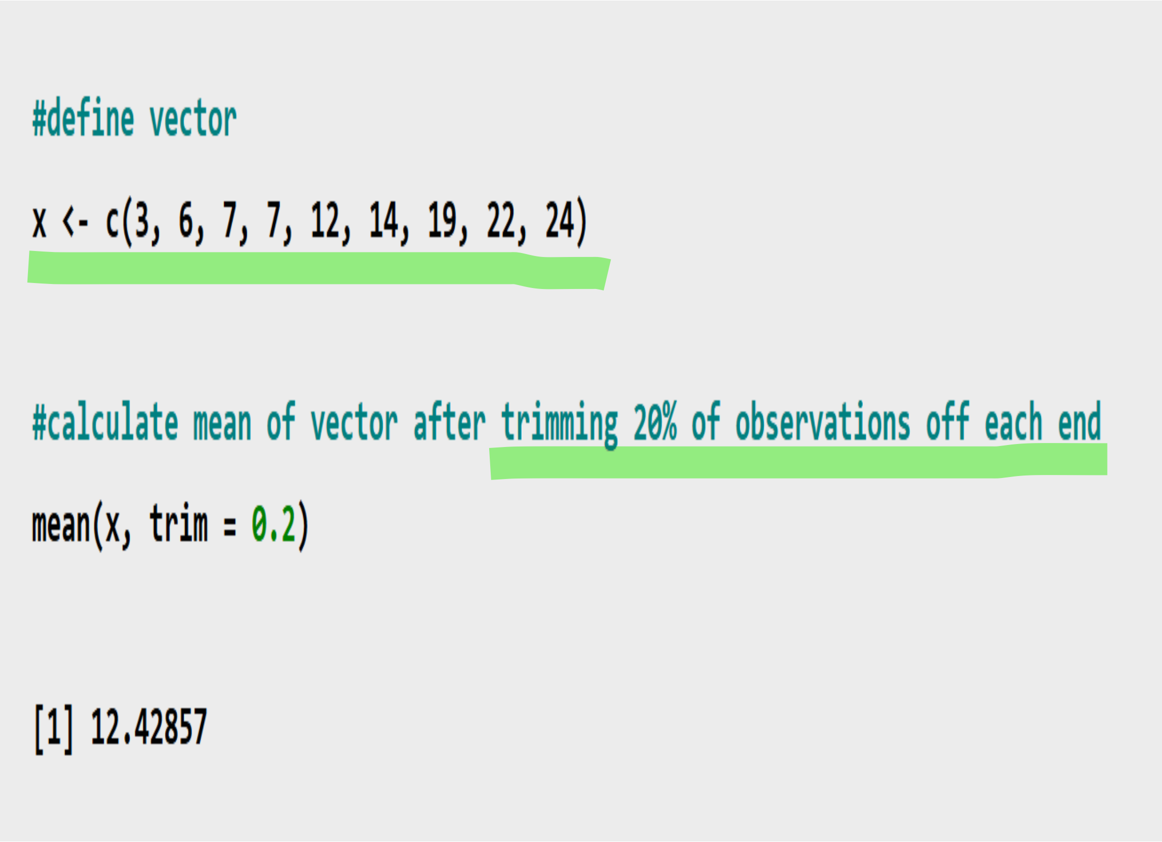
- For a truncated mean, use mean(dat$Sepal.Length,trim=0.10)

  trim varies from 0 to 0.5

Mean → Measure of Central Tendency

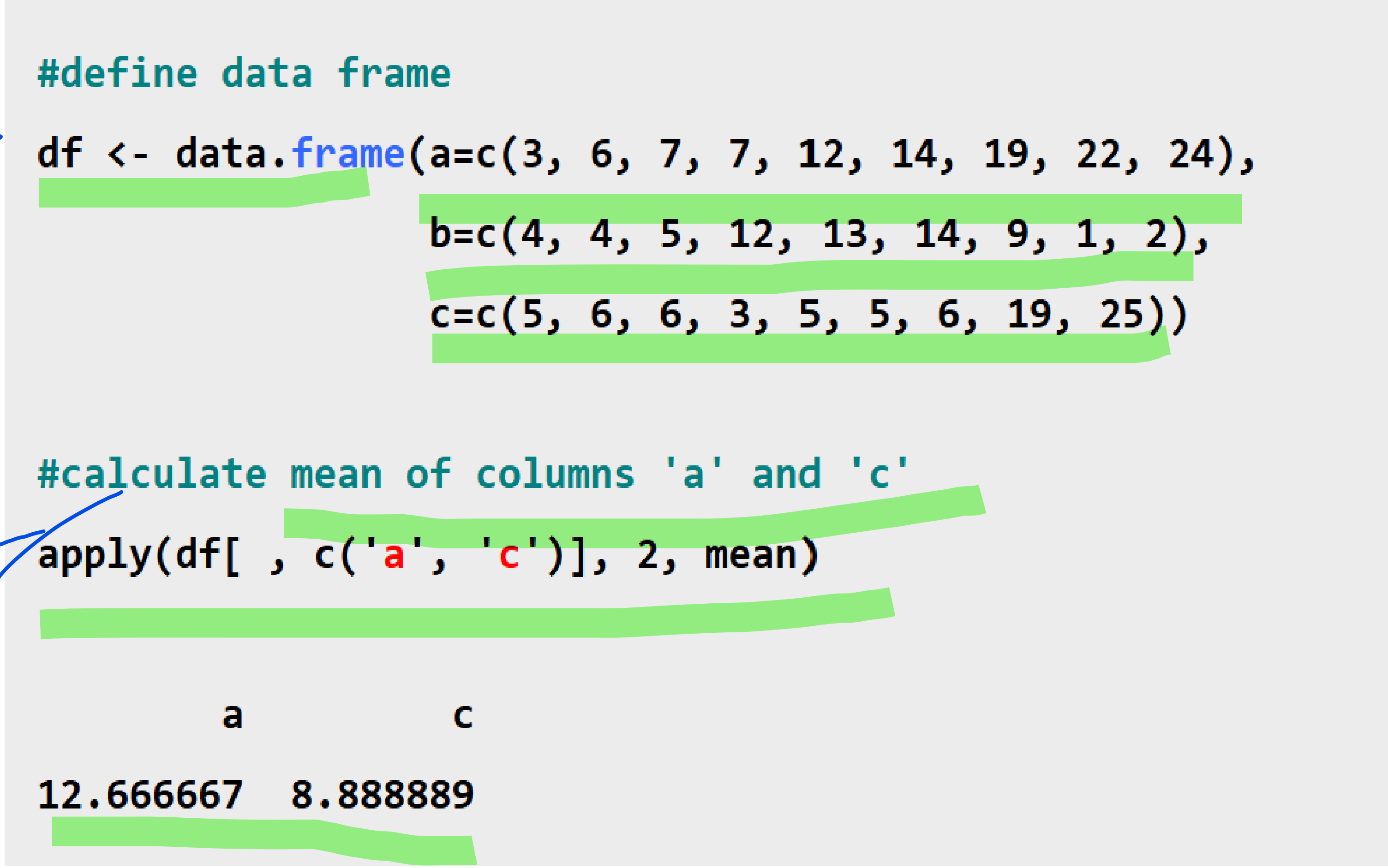


```
#define vector with some missing values

x <- c(3, 6, 7, 7, NA, 14, NA, 22, 24)


#calculate mean of vector

mean(x, na.rm = TRUE)


[1] 11.85714
```

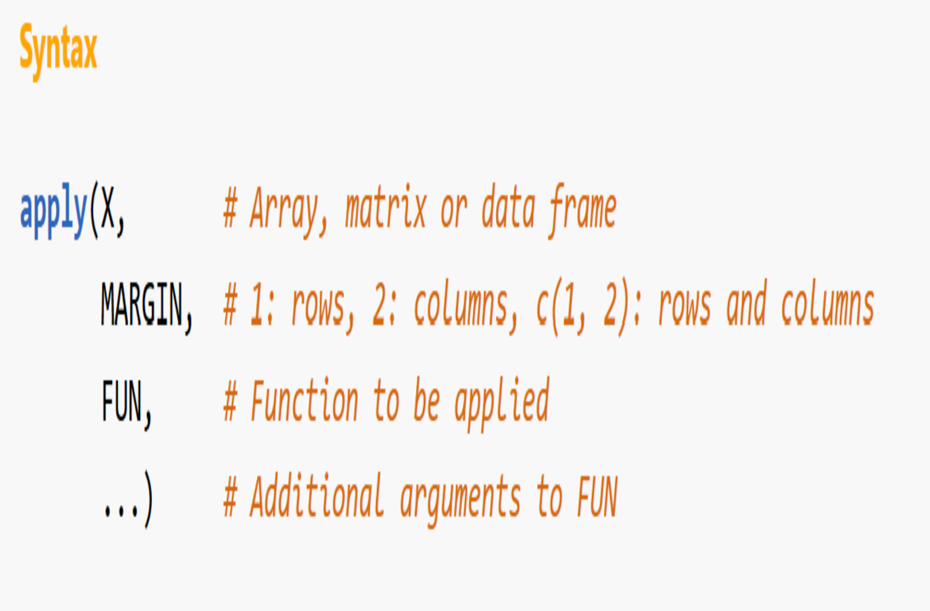


```
#define vector

x <- c(3, 6, 7, 7, 12, 14, 19, 22, 24)


#calculate mean of vector after trimming 20% of observations off each end

mean(x, trim = 0.2)


[1] 12.42857
```

Mean → Measure of Central Tendency



```
#define data frame

df <- data.frame(a=c(3, 6, 7, 7, 12, 14, 19, 22, 24),
                 b=c(4, 4, 5, 12, 13, 14, 9, 1, 2),
                 c=c(5, 6, 6, 3, 5, 5, 6, 19, 25))


#calculate mean of columns 'a' and 'c'

apply(df[ , c('a', 'c')], 2, mean)


        a         c

12.666667  8.888889
```



Syntax

```
apply(X,       # Array, matrix or data frame
      MARGIN,  # 1: rows, 2: columns, c(1, 2): rows and columns
      FUN,     # Function to be applied
      ...)     # Additional arguments to FUN
```

Median → Measure of Central Tendency

The median is the middle value in a set of data points when they are arranged in order. In R, the median() function can be used to calculate the mean.

```
median(dat$Sepal.Length)
```

Tips:

If there is at least one missing value in your dataset, use median(dat$Sepal.Length,na.rm=TRUE) to compute the mean with the NA excluded

Mode → Measure of Central Tendency

In R, unlike mean and median, there's no built-in function to calculate mode. We need to create a user defined function to calculate mode. For example,

```
# vector of marks
marks <- c(97, 78, 57,78, 97, 66, 87, 64, 87, 78)
# define mode() function
mode = function() {
 # calculate mode of marks
return(names(sort(-table(marks)))[1])
}
# call mode()
mode()
```

```
# define mode() function
mode = function(marks) {
  # calculate mode of marks
  return(names(sort(-table(marks)))[1])
}

# call mode() with a marks vector
# mode(marks)
```

Measures of variability

- **Variability** (also known as **Statistical Dispersion**) is another feature of descriptive statistics.
- Measures of central tendency and variability together comprise of descriptive statistics.
- Variability shows the spread of a data set around a point.

Example: Suppose, there exist 2 data sets with the same mean value:

A = 4, 4, 5, 6, 6

Mean(A) = 5

B = 1, 1, 5, 9, 9

Mean(B) = 5

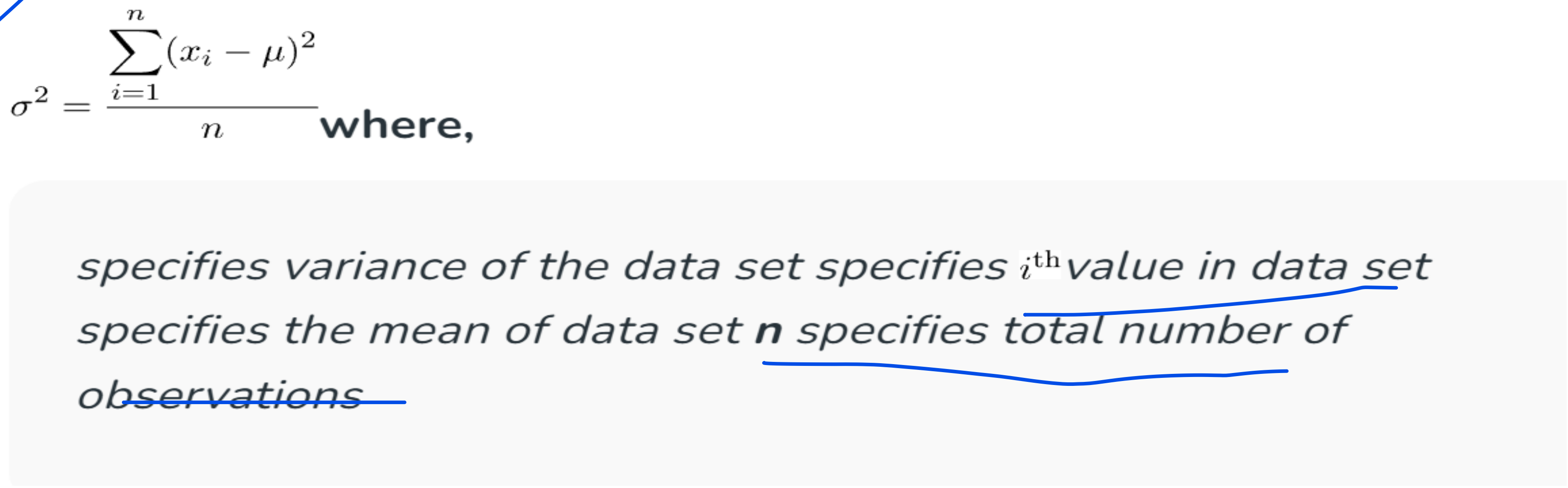
Measures of variability

- So, to differentiate among the two data sets, R offers various measures of variability.

- **Variance** ✓
- **Standard Deviation** ✓
- **Range** ✓
- **Interquartile Range** ✓

Variance -> Measures of variability

- Variance is a measure that shows how far each value is from a particular point, preferably the mean value.
- Mathematically, it is defined as the average of squared differences from the mean value.



$$\sigma^2 = \frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n}$$ **where,**

*specifies variance of the data set specifies i^{th} value in data set specifies the mean of data set **n** specifies total number of observations*

Variance -> Measures of variability

- In the R language, there is a standard built-in function to calculate the variance of a data set.
- Syntax: var(x)
- Where x is the data vector
- **Example**

```
# Defining vector
x <- c(5, 5, 8, 12, 15, 16)

# Print variance of x
print(var(x))
```

Standard Deviation -> Measures of variability

- Standard deviation in statistics measures the spreadness of data values with respect to mean and mathematically, is calculated as square root of variance

- **Example**

```
# Defining vector
x <- c(5, 5, 8, 12, 15, 16)

# Print variance of x
print(sqrt(var(x)))
```

Range-> Measures of variability

- Range is the difference between the maximum and minimum value of a data set.
- In R language, **max()** and **min()** is used to find the same, unlike **range()** function that returns the minimum and maximum value of the data set.
- The **range()** **function** in R is used to return a vector with two elements:
  - ✓The first element represents the minimum value of the input vector.
  - ✓The second element is the maximum value of the input vector.
- The range() function takes the following parameter values:
  - First parameter that represents any numeric or character objects or vectors.
  - na.rm: This takes a Boolean value (TRUE or FALSE) indicating if the NaN (Not a Number) values should be omitted or not.

Range-> Measures of variability

- **Example**

```
# Defining vector
x <- c(5, 5, 8, 12, 15, 16)

# range() function output
print(range(x))              #5 16

# Using max() and min() function
# to calculate the range of data set
print(max(x)-min(x))         #11
```

Range-> Measures of variability

```
# create vector
data = c(12, 45, NA, NA, 67, 23, 45, 78, NA, 89)

# display
print(data)

# find range in vector
print(range(data, na.rm=TRUE))
```

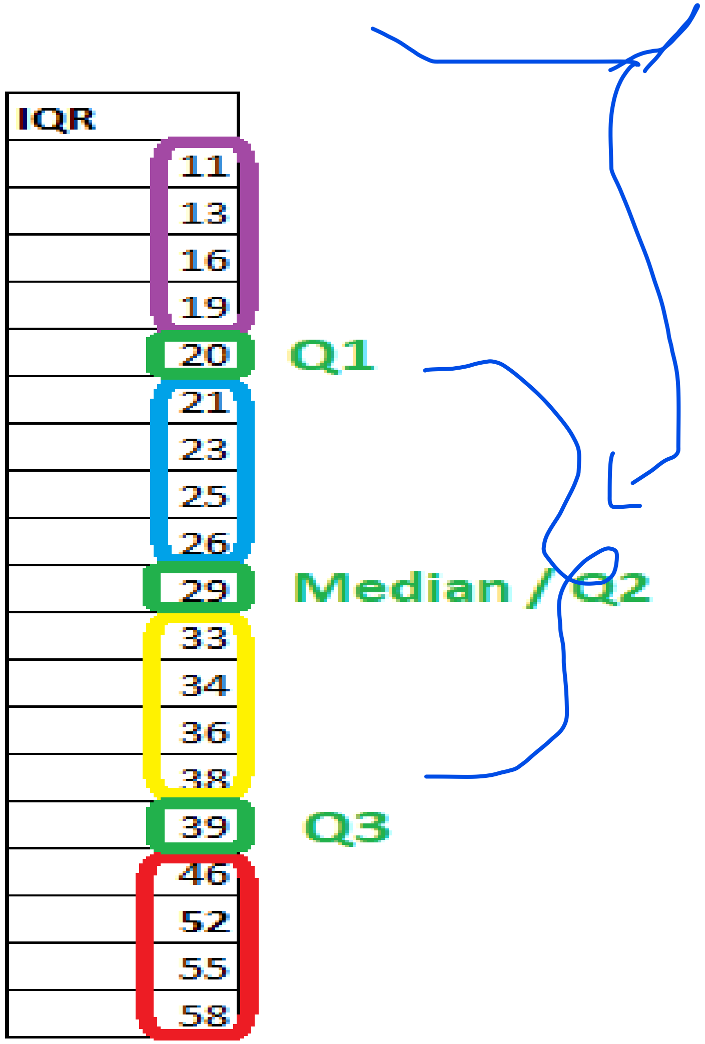
Range-> Measures of variability

- The range tells you the spread of your data from the lowest to the highest value in the distribution
- For example: Consider two datasets, dataset 1 has a range of 20 – 38 = 18 while dataset 2 has a range of 11 – 52 = 41. Dataset 2 has a broader range and, hence, more variability than dataset 1.
- Because only 2 numbers are used, the range is influenced by outliers and doesn’t give you any information about the distribution of values.

Interquartile Range-> Measures of variability

- The interquartile range is the middle half of the data.
- To visualize it, think about the median value that splits the dataset in half. Similarly, we can divide the data into quarters.
- Statisticians refer to these quarters as quartiles and denote them from low to high as Q1, Q2, and Q3.
- The lowest quartile (Q1) contains the quarter of the dataset with the smallest values.
- The upper quartile (Q3) contains the quarter of the dataset with the highest values.
- The interquartile range is the middle half of the data that is in between the upper and lower quartiles.
- In other words, the interquartile range includes the 50% of data points that fall between Q1 and Q3.

Interquartile Range-> Measures of variability



IQR
11
13
16
19
20
21
23
25
26
29
33
34
36
38
39
46
52
55
58

- Interquartile Range is based on splitting a data set into parts called as quartiles.
- There are 3 quartile values (Q1, Q2, Q3) that divide the whole data set into 4 equal parts.
- Q2 specifies the median of the whole data set. Mathematically, the interquartile range is depicted as:

  IQR = Q3 – Q1
- **where, Q3** specifies the median of n largest values **Q1** specifies the median of n smallest values

- **Here IQR= 39-20 = 19**

Interquartile Range-> Measures of variability

```
quantile(iris$Sepal.Length, 0.25)   #Q1- 5.1
quantile(iris$Sepal.Length, 0.75)   #Q3 – 6.4
IQR(iris$Sepal.Length)    #Q3-Q1 – 6.4-5.1=1.3
```

quantile is a legit function, not quartile

Skewness and Kurtosis

- In statistics, **skewness** and **kurtosis** are the measures that tell about the shape of the data distribution, or simply, both are numerical methods to analyze the shape of data set unlike, plotting graphs and histograms which are graphical methods.

- These are normality tests to check the irregularity and asymmetry of the distribution.

- To calculate skewness and kurtosis in R language, a **moments** package is required.

Skewness

- **Skewness** is a measure of the asymmetry of a distribution. This value can be positive or negative.

Formula:

$$\gamma_1 = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^3}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^{3/2}}$$

skew gamma = 1/n * Sum (Xi- X)^3

(1/n * Sum (Xi - X)^2) ^3/2

Where, x_i -> i th value in the data vector
$\bar{x}$ -> mean value of the data vector
n-> number of observations

- A negative skew indicates that the tail is on the left side of the distribution, which extends towards more negative values.
- A positive skew indicates that the tail is on the right side of the distribution, which extends towards more positive values.
- A value of zero indicates that there is no skewness in the distribution at all, meaning the distribution is perfectly symmetrical.

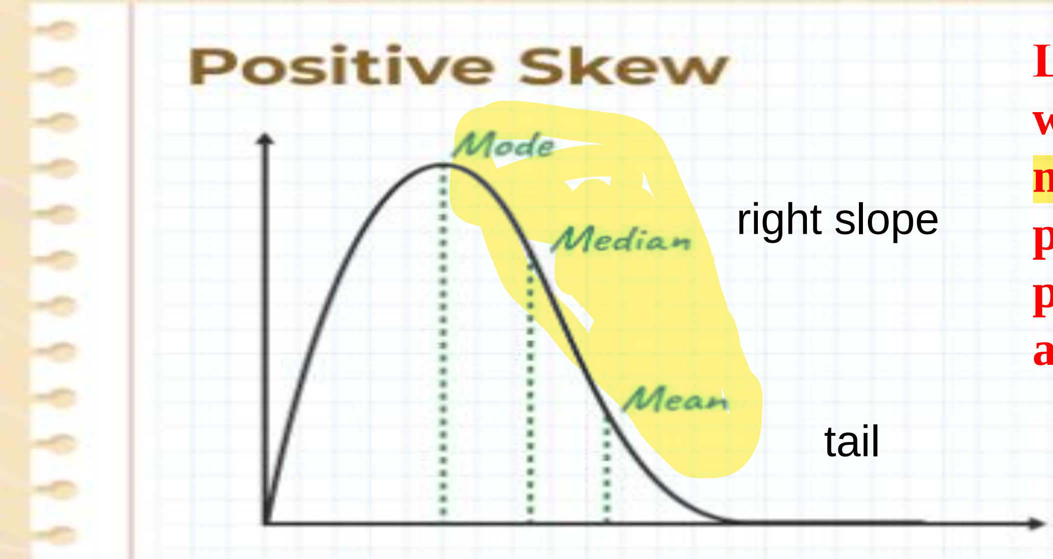
Skewness

Positive Skewness

Positive Skewness means the tail on the right side of the distribution is longer. The mean and median will be greater than the mode.

Condition for positive skewness = **Mean > Median >Mode**

The positive curve of skewness is shown in the image below,



Let's take an example of the income distribution where a few people earn very high incomes and the majority earn lower incomes. so, this is often positively skewed. Analyzing skewed data can provide valuable insights into the underlying causes and potential solutions or interventions.

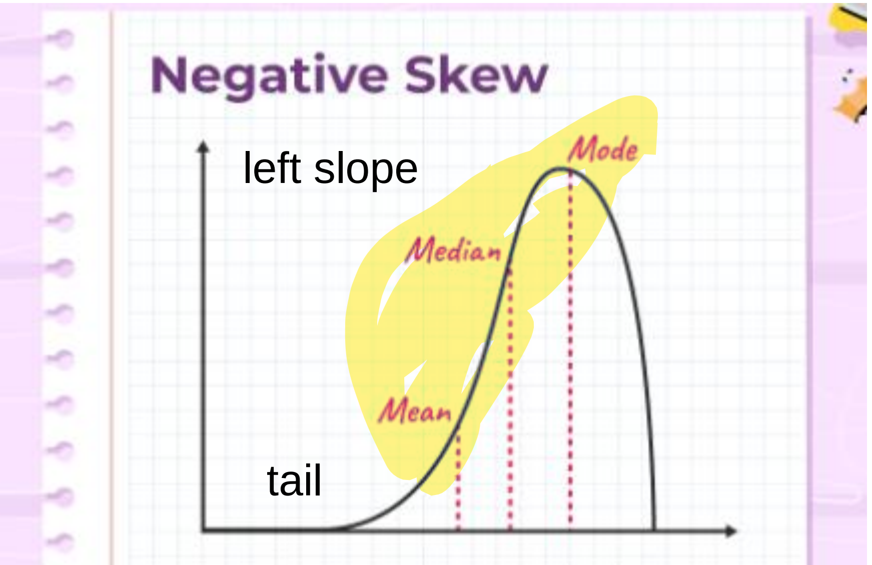
Skewness

Negative Skewness

Negative Skewness means when the tail of the left side of the distribution is longer than the tail on the right side. The mean and median will be less than the mode.

Condition for negative skewness is **Mode > Median > Mean**

The curve shows negative skewness in the image below,



Let's take an example of a match, during the match most of the players of a particular team scored runs above 50 and only a few of them scored below 10. In such a case, the data is generally represented with the help of a negatively skewed distribution. And this data is helpful to analyze the game's performance.

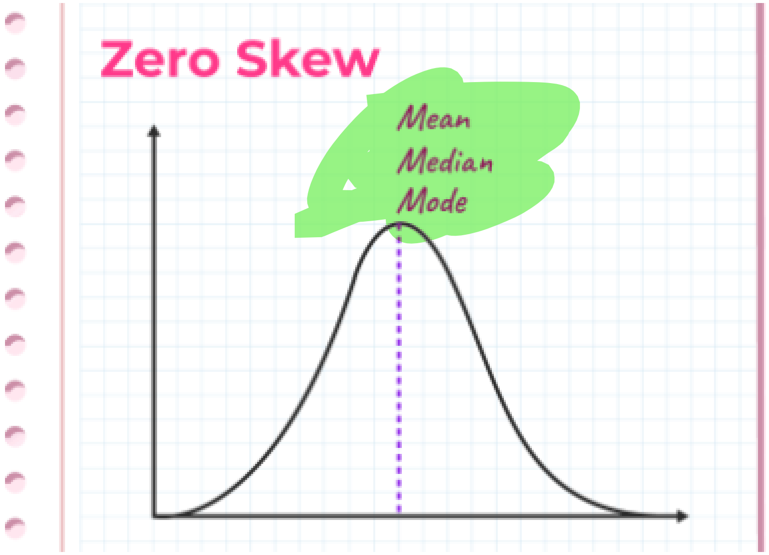
Skewness

Zero Skewness

It is also known as a “symmetric distribution”.It signifies that distribution of data is evenly distributed around the mean, with no long tails on either end of the distribution

Condition for zero skewness is **Mean = Mode = Median**

The curve for zero skews is shown in the image below,



Skewness

```
library(moments)
d<-c(25,28,26,30,40,50,40)
skewness(d)   # 0.6121401
```

So skewness for these data is positive, indicates what???

this indicates that the distribution is right-skewed.

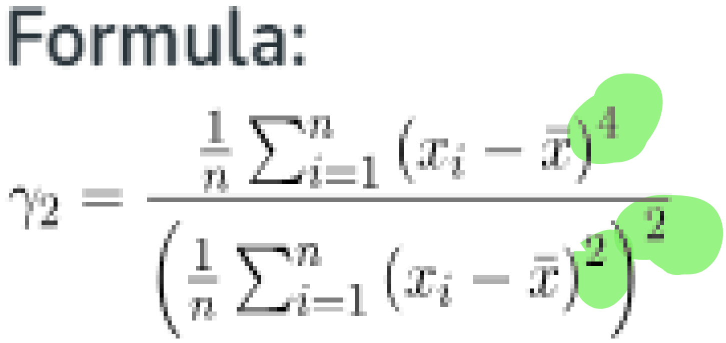
```
library(moments)
d<-c(2,4,6,6)
skewness(d)   #-0.4933822
mean(d)       #4.5
median(d)     #5
```

So skewness for these data is negative,indicates what???

this indicates that the distribution is left-skewed.

Kurtosis

- A statistical measure known as kurtosis measures the peakedness, flatness, and weight of the tails of data distributions.
- In a number of disciplines, including finance, economics, social sciences, and data analysis, an understanding of kurtosis is crucial.



Formula:

$$\gamma_2 = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^4}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^2}$$

Where, x_i -> i th value in the data vector
$\bar{x}$ -> mean value of the data vector
n-> number of observations

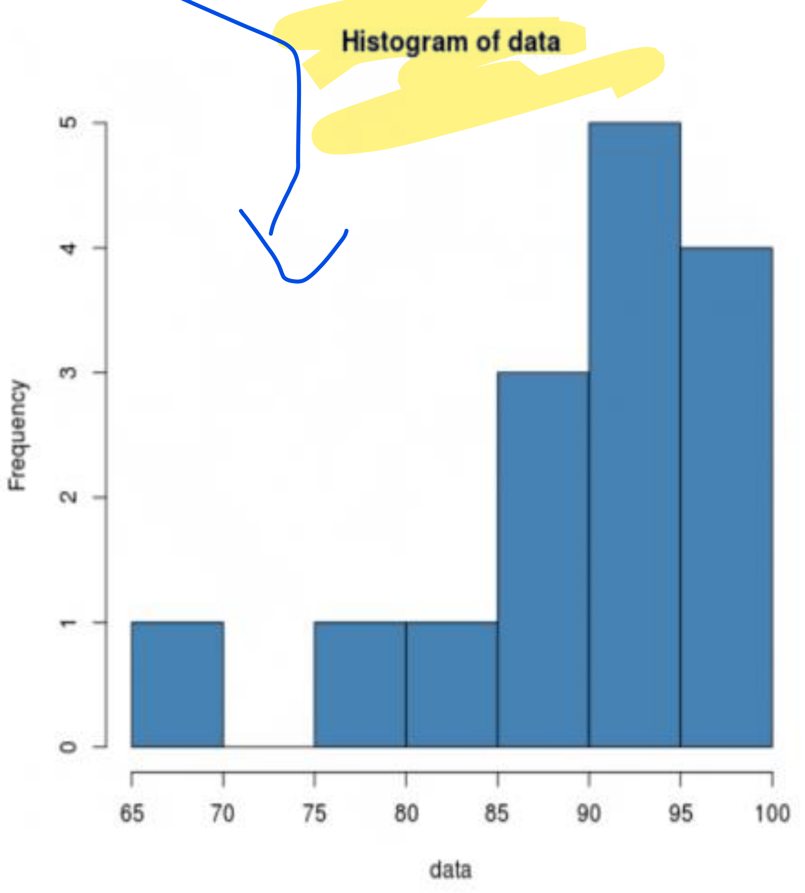
Kurtosis

- **Kurtosis** is a measure of whether or not a distribution is heavy-tailed or light-tailed relative to a normal distribution.
  - ✓ The kurtosis of a normal distribution is 3. cuz it is symmetric
  - ✓ If a given distribution has a kurtosis less than 3, it is said to be *playkurtic*, which means it tends to produce fewer and less extreme outliers than the normal distribution. light tailed
  - ✓ If a given distribution has a kurtosis greater than 3, it is said to be *leptokurtic*, which means it tends to produce more outliers than the normal distribution.

heavy tailed

Kurtosis

```
library(moments)
data = c(88, 95, 92, 97, 96, 97, 94, 86, 91, 95, 97, 88, 85, 76, 68)
kurtosis(data)    #4.177865
hist(data)
```



Since the kurtosis is greater than 3, this indicates that the distribution has more values in the tails compared to a normal distribution.

leptokurtic

Summary() Function

summary() Function:

- **The summary() function** in R is a versatile tool that provides a concise and informative overview of the key characteristics of a dataset, including numerical and categorical variables.
- It is particularly useful for performing initial exploratory data analysis (EDA) to quickly understand the distribution and basic properties of the data.
- The function generates a summary output for each variable in the dataset, presenting a variety of descriptive statistics based on the data type.

Summary() Function

For numerical variables, the summary() function produces the following information:

- **Minimum and Maximum:** The smallest and largest values in the dataset.

- **1st Quartile (Q1), Median (2nd Quartile), and 3rd Quartile (Q3):** These are the values that divide the data into four equal parts, providing insights into the central tendency and data spread.

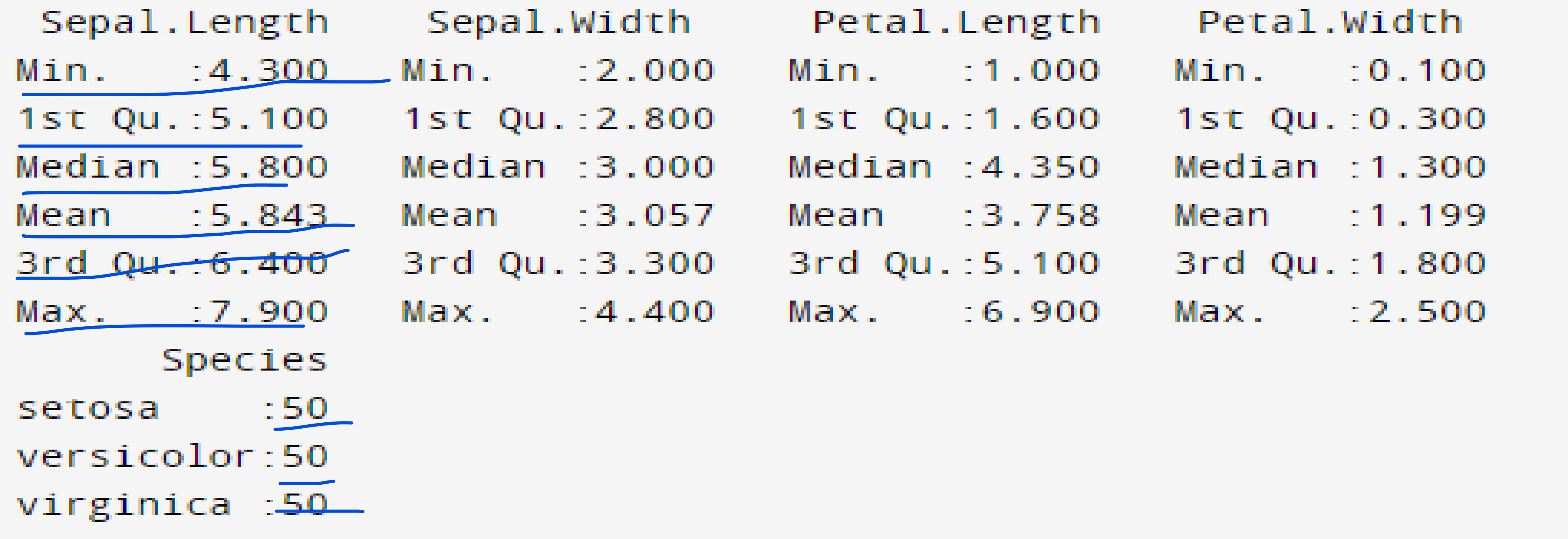
- **Mean:** The arithmetic average of the data points.

- **Standard Deviation:** A measure of the dispersion or spread of the data around the mean.

For categorical variables, the summary() function displays the frequency count of each unique value and the mode (most frequently occurring value).

Summary() Function

summary(iris,nar.rm=TRUE)



```
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  
 Median :5.800   Median :3.000   Median :4.350   Median :1.300  
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500  
       Species  
 setosa    :50  
 versicolor:50  
 virginica :50  
```

Describe() Function

- The describe() function in R Programming Language is a useful tool for generating descriptive statistics of data.
- It provides a comprehensive summary of the variables in a data frame, including central tendency, variability, and distribution measures.
- This function is particularly valuable for preliminary data analysis, helping to understand the basic characteristics of the dataset.
- The describe() function is available in several R packages, with Hmisc and psych being the most popular.

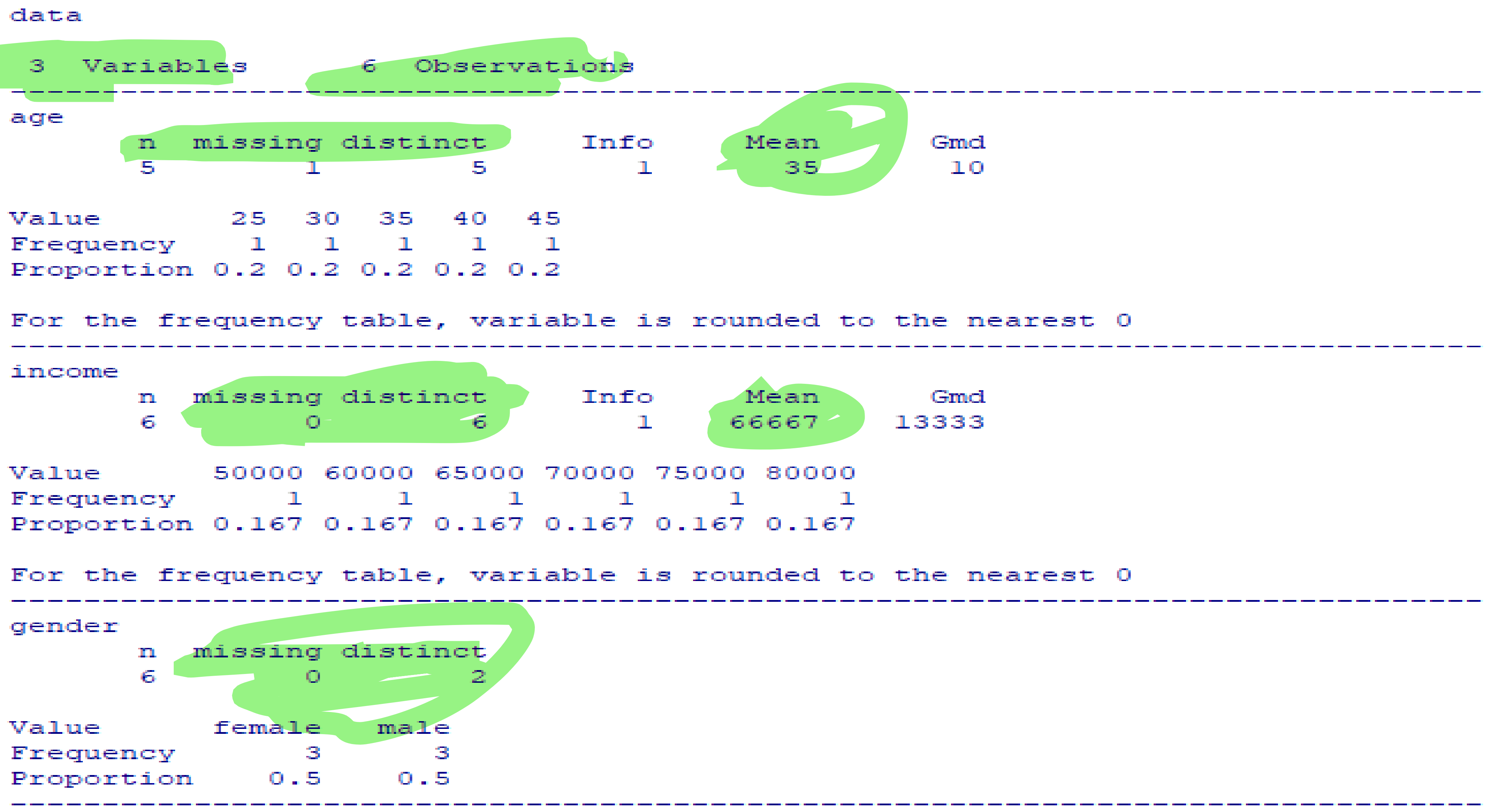
```
install.packages("Hmisc")
library(Hmisc)

install.packages("psych")
library(psych)
```

Describe() Function – Hmisc Package

```
library(Hmisc)
# Example data frame
data <- data.frame(
  age = c(25, 30, 35, 40, 45, NA),
  income = c(50000, 60000, 65000, 70000, 75000, 80000),
  gender = factor(c("male", "female", "female", "male", "male", "female"))
)
# Using describe() from Hmisc
describe(data)
```

The output includes the number of observations (n), missing values (missing), unique values (unique), mean, standard deviation (sd), and various percentiles for numeric variables. For factor variables, it shows the count and the unique categories.

Describe() Function



```
data

 3  Variables        6  Observations
---------------------------------------------------------------------------
age
       n  missing distinct     Info     Mean      Gmd
       5        1        5        1       35       10

Value         25   30   35   40   45
Frequency      1    1    1    1    1
Proportion   0.2  0.2  0.2  0.2  0.2

For the frequency table, variable is rounded to the nearest 0
---------------------------------------------------------------------------
income
       n  missing distinct     Info     Mean      Gmd
       6        0        6        1    66667    13333

Value       50000 60000 65000 70000 75000 80000
Frequency       1     1     1     1     1     1
Proportion  0.167 0.167 0.167 0.167 0.167 0.167

For the frequency table, variable is rounded to the nearest 0
---------------------------------------------------------------------------
gender
       n  missing distinct
       6        0        2

Value       female   male
Frequency        3      3
Proportion     0.5    0.5
---------------------------------------------------------------------------
```

Describe() Function - psych Package

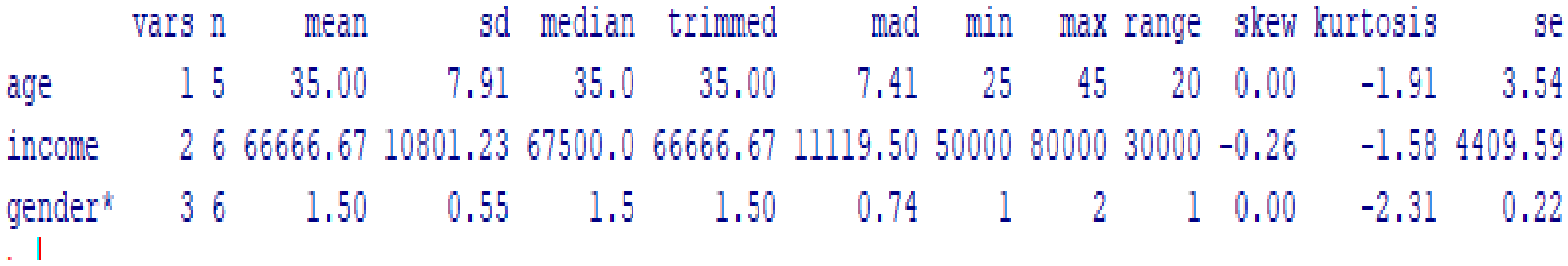
- The describe() function from the psych package also provides a summary of descriptive statistics, but with a focus on psychological data. It includes measures such as skewness and kurtosis.

- Output includes the following:
  - vars indicates the variable index.
  - n is the number of non-missing values.
  - mean is the average.
  - sd is the standard deviation.
  - median is the middle value.
  - trimmed is the mean after trimming 10% of the observations from each tail.
  - mad is the median absolute deviation.
  - min and max are the minimum and maximum values.
  - range is the difference between the maximum and minimum.
  - skew is the skewness of the distribution.
  - kurtosis is the measure of the “tailedness” of the distribution.
  - se is the standard error.

All

Describe() Function – psych Package

```
library(pysch)
# Example data frame
data <- data.frame(
  age = c(25, 30, 35, 40, 45, NA),
  income = c(50000, 60000, 65000, 70000, 75000, 80000),
  gender = factor(c("male", "female", "female", "male", "male", "female"))
)
# Using describe() from Hmisc
describe(data)
```



```
        vars n     mean       sd  median  trimmed      mad   min   max range  skew kurtosis      se
age        1 5    35.00     7.91    35.0    35.00     7.41    25    45    20  0.00    -1.91    3.54
income     2 6 66666.67 10801.23 67500.0 66666.67 11119.50 50000 80000 30000 -0.26    -1.58 4409.59
gender*    3 6     1.50     0.55     1.5     1.50     0.74     1     2     1  0.00    -2.31    0.22
```

Descriptive statistics by group

- We may want to calculate descriptive statistics for each column in a data frame in R, grouped by a particular column.

- One of the easiest ways to do so is by using the **describeBy()** function from the **psych** package in R, which can be used to perform this exact task.

- The **describeBy()** function uses the following syntax:

  describeBy(x, group=NULL, …)

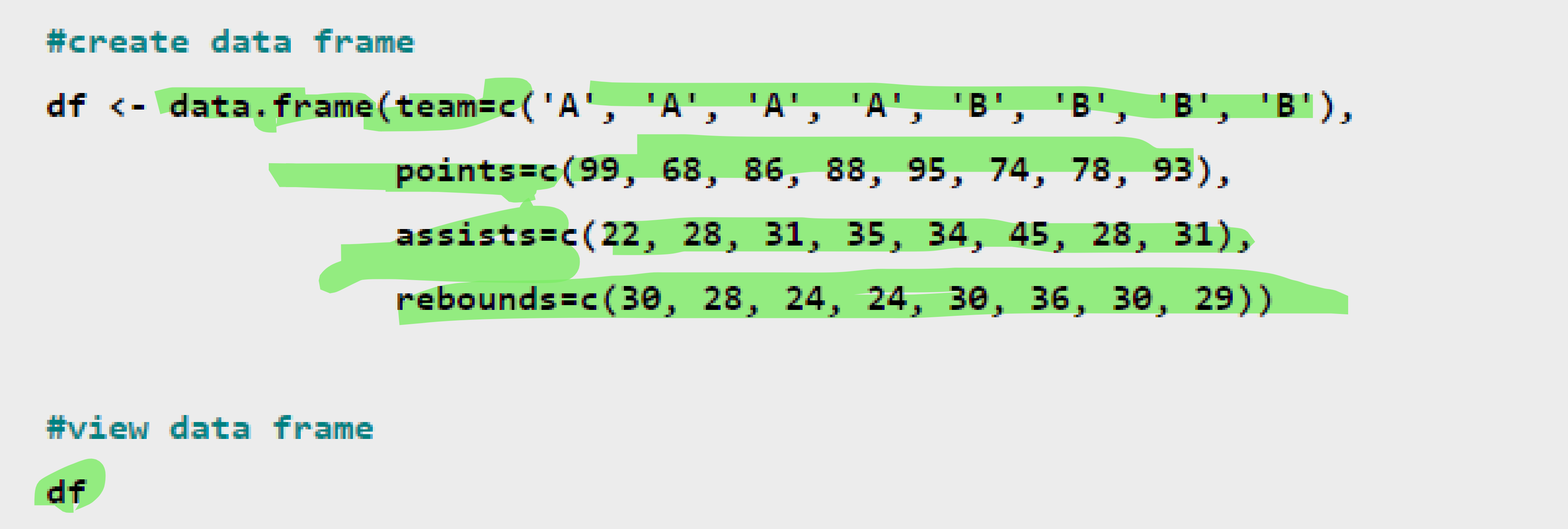
Where,

x: Name of data frame

group: A grouping variable or list of grouping variables

Descriptive statistics by group

- Suppose that we create the following data frame in R that contains information about various basketball players:

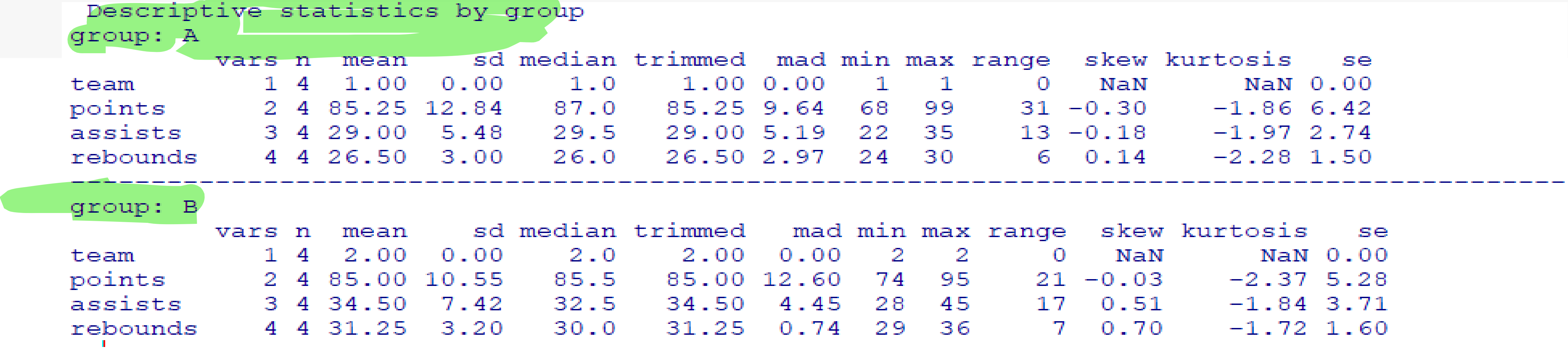


```
#create data frame
df <- data.frame(team=c('A', 'A', 'A', 'A', 'B', 'B', 'B', 'B'),
                 points=c(99, 68, 86, 88, 95, 74, 78, 93),
                 assists=c(22, 28, 31, 35, 34, 45, 28, 31),
                 rebounds=c(30, 28, 24, 24, 30, 36, 30, 29))

#view data frame
df
```

Descriptive statistics by group

- To calculate descriptive statistics for each of the numeric variables in the data frame, grouped by the values in the **team** column.

library(pysch)
describeBy(df, df$team)   #grouping one variable



```
 Descriptive statistics by group
group: A
         vars n  mean    sd median trimmed  mad min max range  skew kurtosis   se
team        1 4  1.00  0.00    1.0    1.00 0.00   1   1     0   NaN      NaN 0.00
points      2 4 85.25 12.84   87.0   85.25 9.64  68  99    31 -0.30    -1.86 6.42
assists     3 4 29.00  5.48   29.5   29.00 5.19  22  35    13 -0.18    -1.97 2.74
rebounds    4 4 26.50  3.00   26.0   26.50 2.97  24  30     6  0.14    -2.28 1.50
---------------------------------------------------------------------------------------------
group: B
         vars n  mean    sd median trimmed   mad min max range  skew kurtosis   se
team        1 4  2.00  0.00    2.0    2.00  0.00   2   2     0   NaN      NaN 0.00
points      2 4 85.00 10.55   85.5   85.00 12.60  74  95    21 -0.03    -2.37 5.28
assists     3 4 34.50  7.42   32.5   34.50  4.45  28  45    17  0.51    -1.84 3.71
rebounds    4 4 31.25  3.20   30.0   31.25  0.74  29  36     7  0.70    -1.72 1.60
```

describeBy(df, list(df$team,df$points))   #grouping two variables