TMC 204
Statistical Data Analysis with R
Unit 5
Statistical Functions in R Implementation

Presented By : Aditya Joshi
Asst. Professor
Department of Computer Application
Graphic Era Deemed to be University

21-04-2020

Descriptive statistics for a single group

- **Measure of central tendency: mean, median, mode**

Roughly speaking, the central tendency measures the “average” or the “middle” of your data. The most commonly used measures include:

the **mean**: the average value. It’s sensitive to outliers.

the **median**: the middle value. It’s a robust alternative to mean.

and the **mode**: the most frequent value

In R,

The function mean() and median() can be used to compute the mean and the median, respectively;

The function mfv() [in the modeest R package] can be used to compute the mode of a variable.

The R code below computes the mean, median and the mode of the variable Sepal.Length [in my_data data set]:

```
# Compute the mean value
> mean(my_data$Sepal.Length)
[1] 5.843333


# Compute the median value
> median(my_data$Sepal.Length)
[1] 5.8

# Compute the mode
# install.packages("modeest")
> require(modeest)
Loading required package: modeest
Warning message:
package ‘modeest’ was built under R version 3.6.3
> mfv(my_data$Sepal.Length)
[1] 5
```

- **Measure of variability**

Measures of variability gives how “spread out” the data are.

✓ **Range: minimum & maximum**

Range corresponds to biggest value minus the smallest value. It gives you the full spread of the data.

```
# Compute the minimum value
> min(my_data$Sepal.Length)
[1] 4.3

# Compute the maximum value
> max(my_data$Sepal.Length)
[1] 7.9

# Range
> range(my_data$Sepal.Length)
[1] 4.3 7.9
```

✓ Interquartile range

Recall that, quartiles divide the data into 4 parts. Note that, the interquartile range (IQR) - corresponding to the difference between the first and third quartiles - is sometimes used as a robust alternative to the standard deviation.

R function:

```
quantile(x, probs = seq(0, 1, 0.25))
```

x: numeric vector whose sample quantiles are wanted.
probs: numeric vector of probabilities with values in [0,1].

```
> quantile(my_data$Sepal.Length)
  0%  25%  50%  75% 100% 
 4.3  5.1  5.8  6.4  7.9 
```

By default, the function returns the minimum, the maximum and three quartiles (the 0.25, 0.50 and 0.75 quartiles).

To compute deciles (0.1, 0.2, 0.3, …., 0.9), use this:

```
> quantile(my_data$Sepal.Length, seq(0, 1, 0.1))
  0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100% 
4.30 4.80 5.00 5.27 5.60 5.80 6.10 6.30 6.52 6.90 7.90 
```

To compute the interquartile range, type this:

```
> IQR(my_data$Sepal.Length)
[1] 1.3
```

✓ Variance and standard deviation

The variance represents the average squared deviation from the mean. The standard deviation is the square root of the variance. It measures the average deviation of the values, in the data, from the mean value.

```
# Compute the variance
> var(my_data$Sepal.Length)
[1] 0.6856935

# Compute the standard deviation =
# square root of the variance
> sd(my_data$Sepal.Length)
[1] 0.8280661
```

✓ Median absolute deviation

The median absolute deviation (MAD) measures the deviation of the values, in the data, from the median value.

```
# Compute the median
> median(my_data$Sepal.Length)
[1] 5.8

# Compute the median absolute deviation
> mad(my_data$Sepal.Length)
[1] 1.03782
```

Which measure to use?

Range. It's not often used because it's very sensitive to outliers.

Interquartile range. It's pretty robust to outliers. It's used a lot in combination with the median.

Variance. It's completely uninterpretable because it doesn't use the same units as the data. It's almost never used except as a mathematical tool

Standard deviation. This is the square root of the variance. It's expressed in the same units as the data. The standard deviation is often used in the situation where the mean is the measure of central tendency.

Median absolute deviation. It's a robust way to estimate the standard deviation, for data with outliers. It's not used very often.

In summary, the IQR and the standard deviation are the two most common measures used to report the variability of the data.

Computing an overall summary of a variable and an entire data frame

- **summary() function**

The function **summary**() can be used to display several statistic summaries of either one variable or an entire data frame.

✓Summary of a single variable

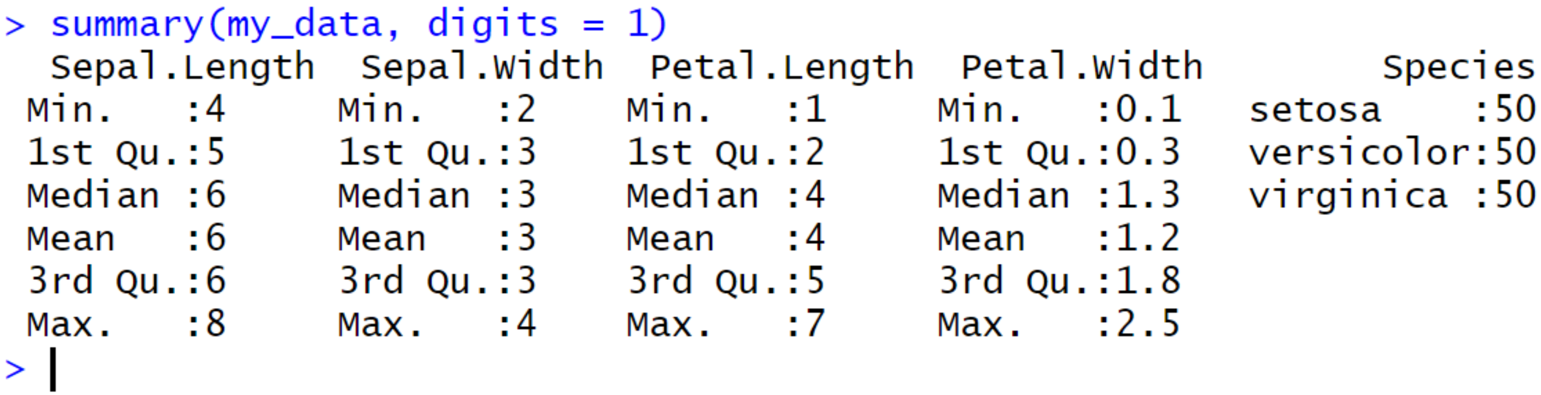
Five values are returned: the mean, median, 25th and 75th quartiles, min and max in one single line call:

```
> summary(my_data$Sepal.Length)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  4.300   5.100   5.800   5.843   6.400   7.900 
```

✓Summary of a data frame

In this case, the function **summary**() is automatically applied to each column. The format of the result depends on the type of the data contained in the column. For example:

- If the column is a numeric variable, mean, median, min, max and quartiles are returned.
- If the column is a factor variable, the number of observations in each group is returned.



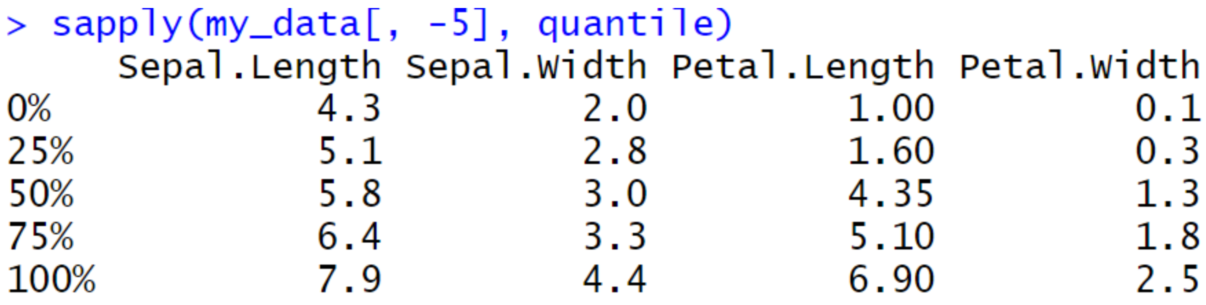
```
> summary(my_data, digits = 1)
  Sepal.Length   Sepal.Width    Petal.Length   Petal.Width          Species  
 Min.   :4      Min.   :2      Min.   :1      Min.   :0.1   setosa    :50  
 1st Qu.:5      1st Qu.:3      1st Qu.:2      1st Qu.:0.3   versicolor:50  
 Median :6      Median :3      Median :4      Median :1.3   virginica :50  
 Mean   :6      Mean   :3      Mean   :4      Mean   :1.2                  
 3rd Qu.:6      3rd Qu.:3      3rd Qu.:5      3rd Qu.:1.8                  
 Max.   :8      Max.   :4      Max.   :7      Max.   :2.5                  
> 
```

- **sapply() function**

It’s also possible to use the function **sapply**() to apply a particular function over a list or vector. For instance, we can use it, to compute for each column in a data frame, the mean, sd, var, min, quantile, …

```
# Compute the mean of each column
> sapply(my_data[, -5], mean)
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
   5.843333     3.057333     3.758000     1.199333
```

# Compute quartiles



```
> sapply(my_data[, -5], quantile)
     Sepal.Length Sepal.Width Petal.Length Petal.Width
0%            4.3         2.0         1.00         0.1
25%           5.1         2.8         1.60         0.3
50%           5.8         3.0         4.35         1.3
75%           6.4         3.3         5.10         1.8
100%          7.9         4.4         6.90         2.5
```

- **stat.desc() function**

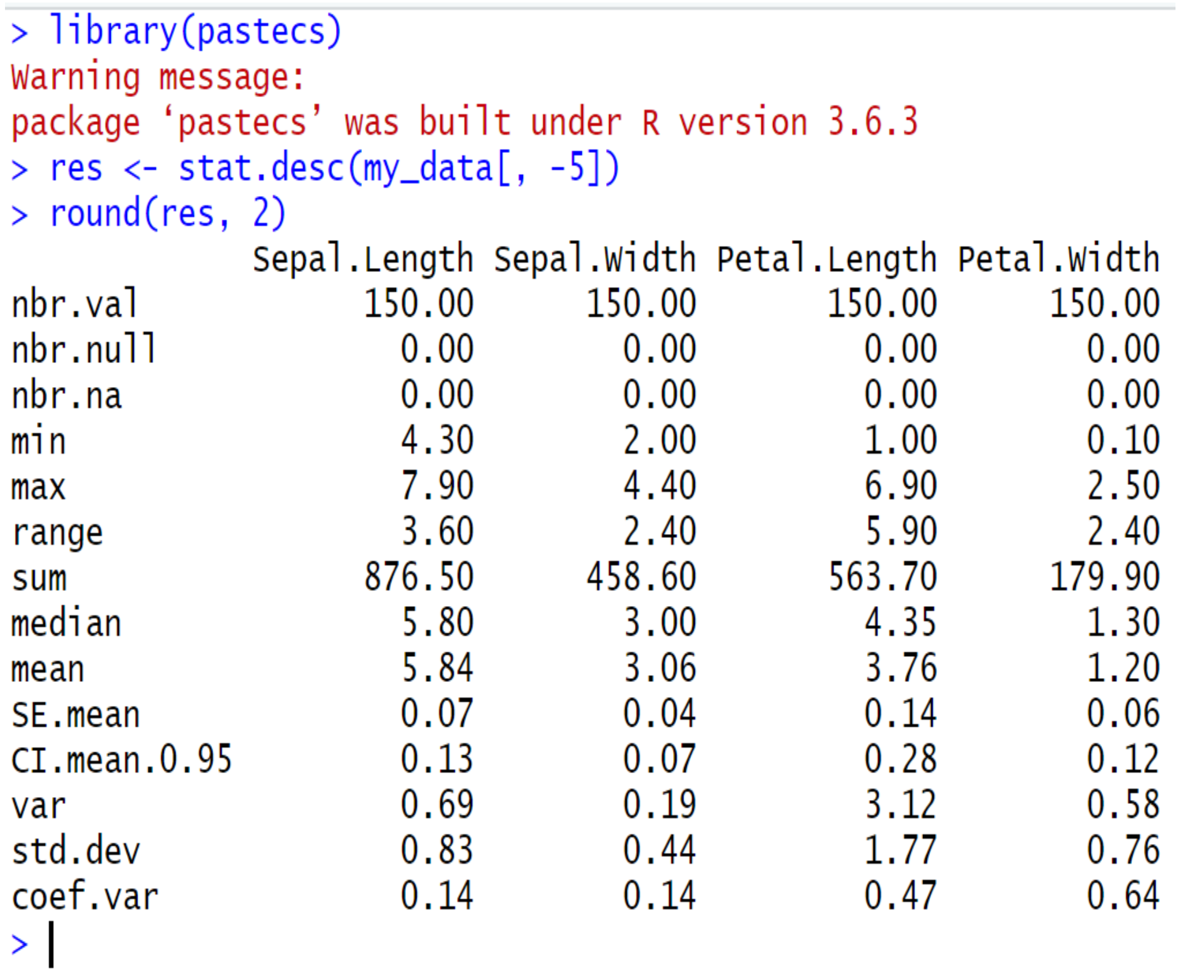
stat.desc() function

The function **stat.desc**() [in **pastecs** package], provides other useful statistics including:

- the median
- the mean
- the standard error on the mean (SE.mean)
- the confidence interval of the mean (CI.mean) at the p level (default is 0.95)
- the variance (var)
- the standard deviation (std.dev)
- and the variation coefficient (coef.var) defined as the standard deviation divided by the mean
- Install **pastecs** package

```
install.packages("pastecs")
```

Use the function **stat.desc**() to compute descriptive statistics



```
> library(pastecs)
Warning message:
package 'pastecs' was built under R version 3.6.3
> res <- stat.desc(my_data[, -5])
> round(res, 2)
             Sepal.Length Sepal.Width Petal.Length Petal.Width
nbr.val            150.00      150.00       150.00      150.00
nbr.null             0.00        0.00         0.00        0.00
nbr.na               0.00        0.00         0.00        0.00
min                  4.30        2.00         1.00        0.10
max                  7.90        4.40         6.90        2.50
range                3.60        2.40         5.90        2.40
sum                876.50      458.60       563.70      179.90
median               5.80        3.00         4.35        1.30
mean                 5.84        3.06         3.76        1.20
SE.mean              0.07        0.04         0.14        0.06
CI.mean.0.95         0.13        0.07         0.28        0.12
var                  0.69        0.19         3.12        0.58
std.dev              0.83        0.44         1.77        0.76
coef.var             0.14        0.14         0.47        0.64
> |
```

Case of missing values

when the data contains missing values, some R functions will return errors or NA even if just a single value is missing.

For example, the **mean()** function will return NA if even only one value is missing in a vector. This can be avoided using the argument **na.rm = TRUE**, which tells to the function to remove any NAs before calculations. An example using the **mean** function is as follow:

```
> mean(my_data$Sepal.Length, na.rm = TRUE)
[1] 5.843333
```