

Statistical Exploration of data

NWUDO CHIKAEZE FIDELIS JUNIOR

2023-02-01

Problem I: The following defects data collected from the last month's inspection reports for a particular type of tank. Construct a Pareto diagram and discuss the results.

```
library(qcc)

## Package 'qcc' version 2.7

## Type 'citation("qcc")' for citing this R package in publications.

library(ggplot2)
library(tidyverse)

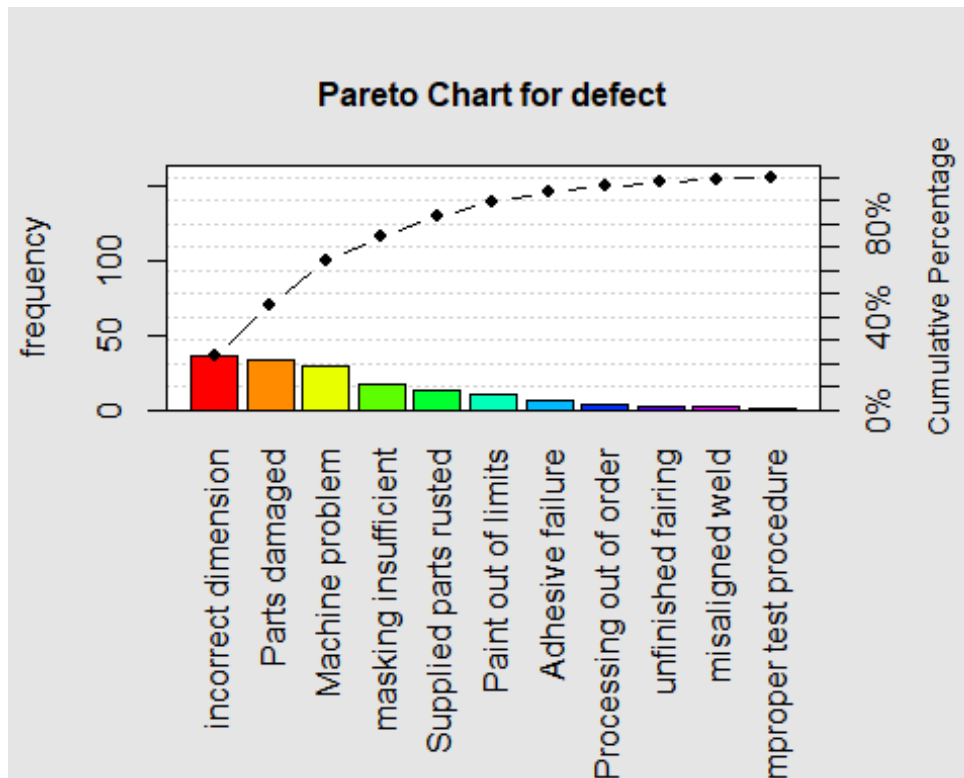
## — Attaching core tidyverse packages ————— tidyverse
2.0.0 —
## ✓ dplyr      1.1.0      ✓ readr      2.1.4
## ✓ forcats    1.0.0      ✓ stringr    1.5.0
## ✓ lubridate  1.9.2      ✓ tibble     3.1.8
## ✓ purrr      1.0.1      ✓ tidyr      1.3.0

## — Conflicts —————
tidyverse_conflicts() —
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()     masks stats::lag()
## i Use the ]8;;http://conflicted.r-lib.org/conflicted-package]8;; to force
all conflicts to become errors

defect <- c(34, 29, 13, 17, 2, 4, 3, 36, 6, 10, 1)

names(defect) <- c('Parts damaged', 'Machine problem', 'Supplied parts
rusted', 'masking insufficient', 'misaligned weld', 'Processing out of
order', 'unfinished fairing', 'incorrect dimension', 'Adhesive failure',
'Paint out of limits', 'improper test procedure')

pareto.chart(defect, ylab = "frequency", cumperc = seq(0, 100, by = 10),
col=rainbow(length(defect)))
```



```
##
## Pareto chart analysis for defect
##
```

	Frequency	Cum.Freq.	Percentage	Cum.Percent.
incorrect dimension	36.0000000	36.0000000	23.2258065	23.2258065
Parts damaged	34.0000000	70.0000000	21.9354839	45.1612903
Machine problem	29.0000000	99.0000000	18.7096774	63.8709677
masking insufficient	17.0000000	116.0000000	10.9677419	74.8387097
Supplied parts rusted	13.0000000	129.0000000	8.3870968	83.2258065
Paint out of limits	10.0000000	139.0000000	6.4516129	89.6774194
Adhesive failure	6.0000000	145.0000000	3.8709677	93.5483871
Processing out of order	4.0000000	149.0000000	2.5806452	96.1290323
unfinished fairing	3.0000000	152.0000000	1.9354839	98.0645161
misaligned weld	2.0000000	154.0000000	1.2903226	99.3548387
improper test procedure	1.0000000	155.0000000	0.6451613	100.0000000

The bars on the chart above depicts which solutions are more significant. The length of the bars represent the frequency and are arranged with the longest bars on the left and the shortest on the right. The linked dots are the cumulative sums. The cumulative sum for the last defect should always reach 100% on the right scale and we can see from the chart above that the cumulative sum for “improper test procedure” was 100%.

- The frequency of incorrect dimension is 36 | The cumulative frequency is 36.
- The frequency of parts damaged is 34 | The cumulative frequency is 70.
- The frequency of machine problem is 29 | The cumulative frequency is 99.
- The frequency of masking insufficient is 17 | The cumulative frequency is 116.
- The frequency of supplied parts rusted is 13 | The cumulative frequency is 129.

- The frequency of paints out of limit is 10 | The cumulative frequency is 139.
- The frequency of adhesive failure is 6 | The cumulative frequency is 145.
- The frequency of Processing out of order is 4 | The cumulative frequency is 149.
- The frequency of unfinished fairing is 3 | The cumulative frequency is 152.
- The frequency of misaligned weld is 2 | The cumulative frequency is 154.
- The frequency of improper test procedure is 1 | The cumulative frequency is 155.

PROBLEM II: Fifty soft drink bottles of a specific brand are collected from one day production and measured its net weight, which are given below: The specification limits for this brand are (16+0.5oz)

15.8 16.3 16.2 16.1 16.6 16.3 15.9 15.9 16.2 16.4 16.1 16.2 16.5 16.4 16.3 16.3 16.2 15.9
16.4 16.2 16.1 16.1 16.4 16.5 16.0 16.1 15.8 16.7 16.6 16.4 16.1 16.3 16.5 16.1 16.5 16.2
16.1 16.2 16.1 16.3 16.3 16.2 16.4 16.3 16.5 16.6 16.3 16.4 16.1 16.5

```
B <- c(15.8, 16.3, 16.1, 16.3, 16.1, 16.1, 16.1, 16.2, 16.3, 16.6, 16.3,
15.9, 16.2, 16.2, 16.1, 15.8, 16.3, 16.1, 16.2, 16.3,
16.2, 15.9, 16.5, 15.9, 16.4, 16.7, 16.5, 16.2, 16.4, 16.4, 16.1,
16.2, 16.4, 16.4, 16.5, 16.6, 16.1, 16.1, 16.3, 16.1, 16.6, 16.4, 16.3,
16.2, 16.0, 16.4, 16.5, 16.3, 16.5, 16.5)
```

1) Estimate the mean and standard deviation

MEAN

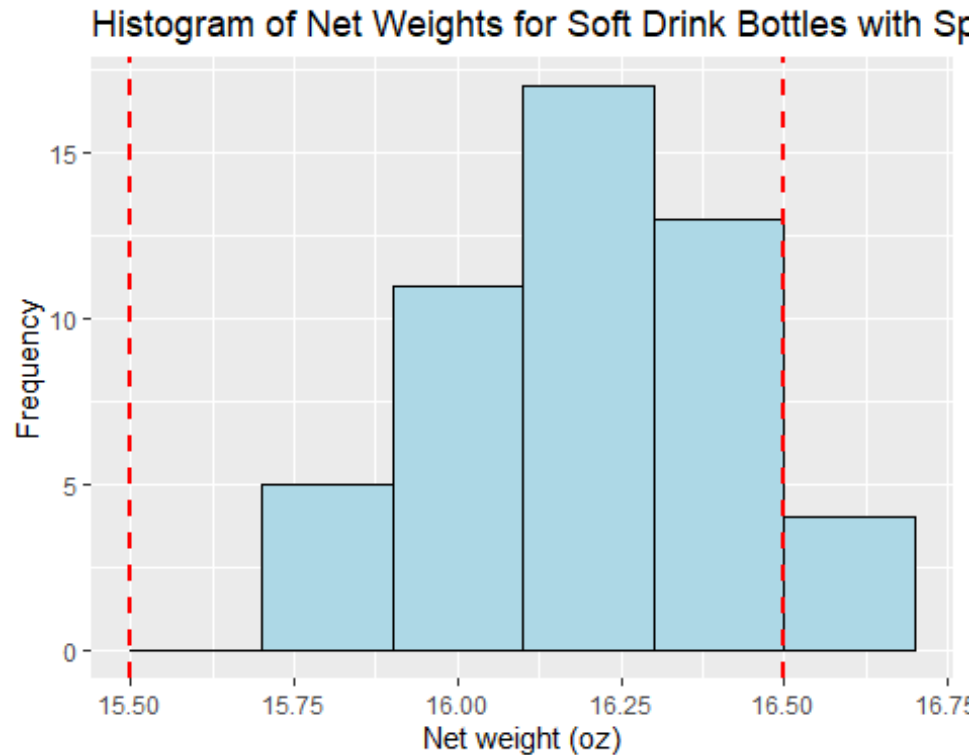
```
Mean <- sum(B) / length(B)
Mean
## [1] 16.258
```

STANDARD DEVIATION

```
Var <- var(B)
SD <- sqrt(Var)
SD
## [1] 0.2119781
```

2) Draw a histogram with superimposing the specification limits. Interpret the histogram focussing on how to improve the process

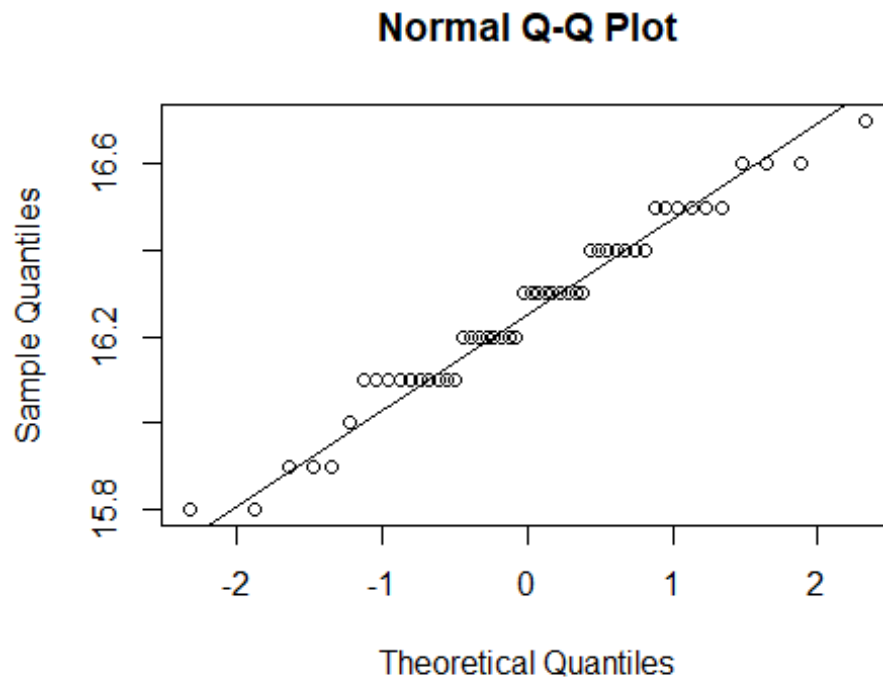
```
ggplot(data.frame(B), aes(x = B)) +
  geom_histogram(binwidth=0.2, fill = "lightblue", color = 'black') +
  geom_vline(xintercept = c(15.5, 16.5), linetype="dashed", linewidth=1,
color="red") +
  labs(x="Net weight (oz)", y="Frequency", title="Histogram of Net Weights
for Soft Drink Bottles with Specification Limits")
```



- The histogram with the superimposed specification limits shows us that some of the bottles in the sample have a net weight that falls outside of the specification limits of (16 ± 0.5) oz. Any item that falls outside of this range would be considered as defective. There are a few bottles with a net weight above 16.5 oz. This suggests that the manufacturing process may not be consistent and may need to be improved to reduce the number of bottles outside of the specification limits.

3) Draw normal probability plot to justify your answer in (2)

```
qqnorm(B)  
qqline(B)
```



4) Assuming the normality, estimate the sigma level of the process?

```
Z_score <- min((16.5 - Mean)/SD, (Mean - 15.5)/SD)
Sigma_level <- Z_score + 1.5
Sigma_level

## [1] 2.641628
```

- With a sigma level of 2.641628, we can say that the process is not meeting customers expectations. A sigma level of 6 is generally considered a very good level of performance.

5) Estimate the percentage of soft drink bottle out side the lower specification and upper specification.

```
range_min <- 15.5
range_max <- 16.5
sample_size <- 50

# Find the percentage of data points that fall outside of the specified range
outside_range <- pnorm(range_min, Mean, SD) + (1 - pnorm(range_max, Mean,
SD))
percent_rejected <- outside_range * 100

print(paste0("Percentage of rejection: ", percent_rejected, "%"))

## [1] "Percentage of rejection: 12.6978967713574%"
```

- 6) Since the soft drink bottles fell outside the upper specification are quite large, it is decided to lower the process mean setting by 0.2 units. If you are the production manager, how many bottles of drink you need to produce to get 10000 accepted bottles, with new process mean setting, but same standard deviation?

```
new_mean = 16.258 - 0.2 # the process mean reduced by 0.2 units

# Find the percentage of data points that fall outside of the specified range
outside_range <- pnorm(range_min, new_mean, SD) + (1 - pnorm(range_max,
new_mean, SD))
percent_rejected <- outside_range * 100
percent_accepted <- 100 - percent_rejected

print(paste0("Percentage of acceptance: ", percent_accepted, "%"))

## [1] "Percentage of acceptance: 97.7231017779453%"

print(paste0(round(10000/(percent_accepted/100)), " bottles need to be
produced to get 10000 accepted bottles "))

## [1] "10233 bottles need to be produced to get 10000 accepted bottles "
```

PROBLEM III: The manager of the local hospital came to know about the SIX SIGMA methodology and overmuch interested to implement it in his hospital. You are hired to do a sample project as data scientist how to improve customer service quality. Since customer satisfaction is an important parameter for assessment, identify one or two quality characteristics related to the customer satisfaction. Explain how you can measure the quality characteristics you have identified.

As a data scientist, there are two critical quality characteristics related to customer satisfaction that we can focus on to improve customer service quality. The first one is response time, which measures the time taken by the hospital staff to respond to customer queries, requests or complaints. The second is communication, which refers to the quality of communication between the hospital staff and customers. Effective communication is essential to ensure that the customers' needs and expectations are understood and met.

To measure these quality characteristics, we can use customer feedback surveys that ask specific questions related to response time and communication quality. Based on the responses, we can calculate response time and communication scores for each customer and then derive the average scores for all customers. These scores can then be used to identify areas for improvement and take corrective actions to enhance customer service quality.

For instance, if the response time score is lower than expected, the hospital management can invest in staff training and revise processes to reduce response time. Similarly, if the communication score is low, the hospital staff can be trained to improve their communication skills, and communication processes can be revisited and improved.

In conclusion, by focusing on critical quality characteristics related to customer satisfaction, collecting and analyzing data through customer feedback surveys, we can identify areas for improvement and take corrective actions to enhance customer service quality.

PROBLEM IV :Perform an exploratory data analysis of the "IRIS Data". Give your interpretations / comments on each analysis you performed.

```
iris
##      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
## 7           4.6         3.4         1.4         0.3    setosa
## 8           5.0         3.4         1.5         0.2    setosa
## 9           4.4         2.9         1.4         0.2    setosa
## 10          4.9         3.1         1.5         0.1    setosa
## 11          5.4         3.7         1.5         0.2    setosa
## 12          4.8         3.4         1.6         0.2    setosa
## 13          4.8         3.0         1.4         0.1    setosa
## 14          4.3         3.0         1.1         0.1    setosa
## 15          5.8         4.0         1.2         0.2    setosa
## 16          5.7         4.4         1.5         0.4    setosa
## 17          5.4         3.9         1.3         0.4    setosa
## 18          5.1         3.5         1.4         0.3    setosa
## 19          5.7         3.8         1.7         0.3    setosa
## 20          5.1         3.8         1.5         0.3    setosa
## 21          5.4         3.4         1.7         0.2    setosa
## 22          5.1         3.7         1.5         0.4    setosa
## 23          4.6         3.6         1.0         0.2    setosa
## 24          5.1         3.3         1.7         0.5    setosa
## 25          4.8         3.4         1.9         0.2    setosa
## 26          5.0         3.0         1.6         0.2    setosa
## 27          5.0         3.4         1.6         0.4    setosa
## 28          5.2         3.5         1.5         0.2    setosa
## 29          5.2         3.4         1.4         0.2    setosa
## 30          4.7         3.2         1.6         0.2    setosa
## 31          4.8         3.1         1.6         0.2    setosa
## 32          5.4         3.4         1.5         0.4    setosa
## 33          5.2         4.1         1.5         0.1    setosa
## 34          5.5         4.2         1.4         0.2    setosa
## 35          4.9         3.1         1.5         0.2    setosa
## 36          5.0         3.2         1.2         0.2    setosa
## 37          5.5         3.5         1.3         0.2    setosa
## 38          4.9         3.6         1.4         0.1    setosa
## 39          4.4         3.0         1.3         0.2    setosa
## 40          5.1         3.4         1.5         0.2    setosa
```

## 41	5.0	3.5	1.3	0.3	setosa
## 42	4.5	2.3	1.3	0.3	setosa
## 43	4.4	3.2	1.3	0.2	setosa
## 44	5.0	3.5	1.6	0.6	setosa
## 45	5.1	3.8	1.9	0.4	setosa
## 46	4.8	3.0	1.4	0.3	setosa
## 47	5.1	3.8	1.6	0.2	setosa
## 48	4.6	3.2	1.4	0.2	setosa
## 49	5.3	3.7	1.5	0.2	setosa
## 50	5.0	3.3	1.4	0.2	setosa
## 51	7.0	3.2	4.7	1.4	versicolor
## 52	6.4	3.2	4.5	1.5	versicolor
## 53	6.9	3.1	4.9	1.5	versicolor
## 54	5.5	2.3	4.0	1.3	versicolor
## 55	6.5	2.8	4.6	1.5	versicolor
## 56	5.7	2.8	4.5	1.3	versicolor
## 57	6.3	3.3	4.7	1.6	versicolor
## 58	4.9	2.4	3.3	1.0	versicolor
## 59	6.6	2.9	4.6	1.3	versicolor
## 60	5.2	2.7	3.9	1.4	versicolor
## 61	5.0	2.0	3.5	1.0	versicolor
## 62	5.9	3.0	4.2	1.5	versicolor
## 63	6.0	2.2	4.0	1.0	versicolor
## 64	6.1	2.9	4.7	1.4	versicolor
## 65	5.6	2.9	3.6	1.3	versicolor
## 66	6.7	3.1	4.4	1.4	versicolor
## 67	5.6	3.0	4.5	1.5	versicolor
## 68	5.8	2.7	4.1	1.0	versicolor
## 69	6.2	2.2	4.5	1.5	versicolor
## 70	5.6	2.5	3.9	1.1	versicolor
## 71	5.9	3.2	4.8	1.8	versicolor
## 72	6.1	2.8	4.0	1.3	versicolor
## 73	6.3	2.5	4.9	1.5	versicolor
## 74	6.1	2.8	4.7	1.2	versicolor
## 75	6.4	2.9	4.3	1.3	versicolor
## 76	6.6	3.0	4.4	1.4	versicolor
## 77	6.8	2.8	4.8	1.4	versicolor
## 78	6.7	3.0	5.0	1.7	versicolor
## 79	6.0	2.9	4.5	1.5	versicolor
## 80	5.7	2.6	3.5	1.0	versicolor
## 81	5.5	2.4	3.8	1.1	versicolor
## 82	5.5	2.4	3.7	1.0	versicolor
## 83	5.8	2.7	3.9	1.2	versicolor
## 84	6.0	2.7	5.1	1.6	versicolor
## 85	5.4	3.0	4.5	1.5	versicolor
## 86	6.0	3.4	4.5	1.6	versicolor
## 87	6.7	3.1	4.7	1.5	versicolor
## 88	6.3	2.3	4.4	1.3	versicolor
## 89	5.6	3.0	4.1	1.3	versicolor
## 90	5.5	2.5	4.0	1.3	versicolor

## 91	5.5	2.6	4.4	1.2	versicolor
## 92	6.1	3.0	4.6	1.4	versicolor
## 93	5.8	2.6	4.0	1.2	versicolor
## 94	5.0	2.3	3.3	1.0	versicolor
## 95	5.6	2.7	4.2	1.3	versicolor
## 96	5.7	3.0	4.2	1.2	versicolor
## 97	5.7	2.9	4.2	1.3	versicolor
## 98	6.2	2.9	4.3	1.3	versicolor
## 99	5.1	2.5	3.0	1.1	versicolor
## 100	5.7	2.8	4.1	1.3	versicolor
## 101	6.3	3.3	6.0	2.5	virginica
## 102	5.8	2.7	5.1	1.9	virginica
## 103	7.1	3.0	5.9	2.1	virginica
## 104	6.3	2.9	5.6	1.8	virginica
## 105	6.5	3.0	5.8	2.2	virginica
## 106	7.6	3.0	6.6	2.1	virginica
## 107	4.9	2.5	4.5	1.7	virginica
## 108	7.3	2.9	6.3	1.8	virginica
## 109	6.7	2.5	5.8	1.8	virginica
## 110	7.2	3.6	6.1	2.5	virginica
## 111	6.5	3.2	5.1	2.0	virginica
## 112	6.4	2.7	5.3	1.9	virginica
## 113	6.8	3.0	5.5	2.1	virginica
## 114	5.7	2.5	5.0	2.0	virginica
## 115	5.8	2.8	5.1	2.4	virginica
## 116	6.4	3.2	5.3	2.3	virginica
## 117	6.5	3.0	5.5	1.8	virginica
## 118	7.7	3.8	6.7	2.2	virginica
## 119	7.7	2.6	6.9	2.3	virginica
## 120	6.0	2.2	5.0	1.5	virginica
## 121	6.9	3.2	5.7	2.3	virginica
## 122	5.6	2.8	4.9	2.0	virginica
## 123	7.7	2.8	6.7	2.0	virginica
## 124	6.3	2.7	4.9	1.8	virginica
## 125	6.7	3.3	5.7	2.1	virginica
## 126	7.2	3.2	6.0	1.8	virginica
## 127	6.2	2.8	4.8	1.8	virginica
## 128	6.1	3.0	4.9	1.8	virginica
## 129	6.4	2.8	5.6	2.1	virginica
## 130	7.2	3.0	5.8	1.6	virginica
## 131	7.4	2.8	6.1	1.9	virginica
## 132	7.9	3.8	6.4	2.0	virginica
## 133	6.4	2.8	5.6	2.2	virginica
## 134	6.3	2.8	5.1	1.5	virginica
## 135	6.1	2.6	5.6	1.4	virginica
## 136	7.7	3.0	6.1	2.3	virginica
## 137	6.3	3.4	5.6	2.4	virginica
## 138	6.4	3.1	5.5	1.8	virginica
## 139	6.0	3.0	4.8	1.8	virginica
## 140	6.9	3.1	5.4	2.1	virginica

```
## 141      6.7      3.1      5.6      2.4 virginica
## 142      6.9      3.1      5.1      2.3 virginica
## 143      5.8      2.7      5.1      1.9 virginica
## 144      6.8      3.2      5.9      2.3 virginica
## 145      6.7      3.3      5.7      2.5 virginica
## 146      6.7      3.0      5.2      2.3 virginica
## 147      6.3      2.5      5.0      1.9 virginica
## 148      6.5      3.0      5.2      2.0 virginica
## 149      6.2      3.4      5.4      2.3 virginica
## 150      5.9      3.0      5.1      1.8 virginica
```

```
# checking for missing values
missing_vals <- sum(is.na(iris))
print(paste0("sum of missing values:", missing_vals))

## [1] "sum of missing values:0"
```

```
# summary statistics
summary(iris)
```

```
##      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
##
##
##
```

The purpose of the `summary()` function is to provide a quick and easy way to obtain a summary of the data in an object. This can be useful for exploring and understanding the characteristics of the data, identifying potential outliers or anomalies, and checking the validity and quality of the data. The `summary()` function can also be useful for comparing the data in different objects or groups, or for identifying trends or patterns in the data.

```
# standard deviation for the first 4 columns of the iris dataset (which are
the variables) grouped by the Species column
aggregate(iris[, 1:4], by=list(Species=iris$Species), sd)
```

```
##      Species Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1      setosa    0.3524897    0.3790644    0.1736640    0.1053856
## 2 versicolor    0.5161711    0.3137983    0.4699110    0.1977527
## 3  virginica    0.6358796    0.3224966    0.5518947    0.2746501
```

- Table showing the Standard deviation value of each variable (Sepal.Length, Sepal.Width, Petal.Length, and Petal.Width) for each of the three species (setosa, versicolor, and virginica).

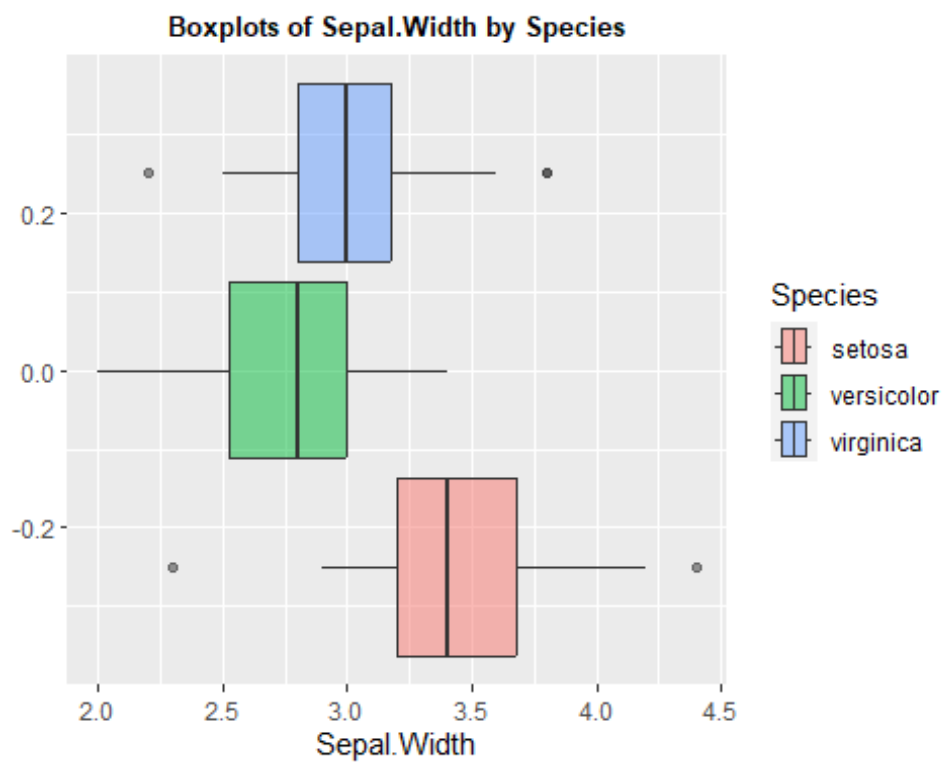
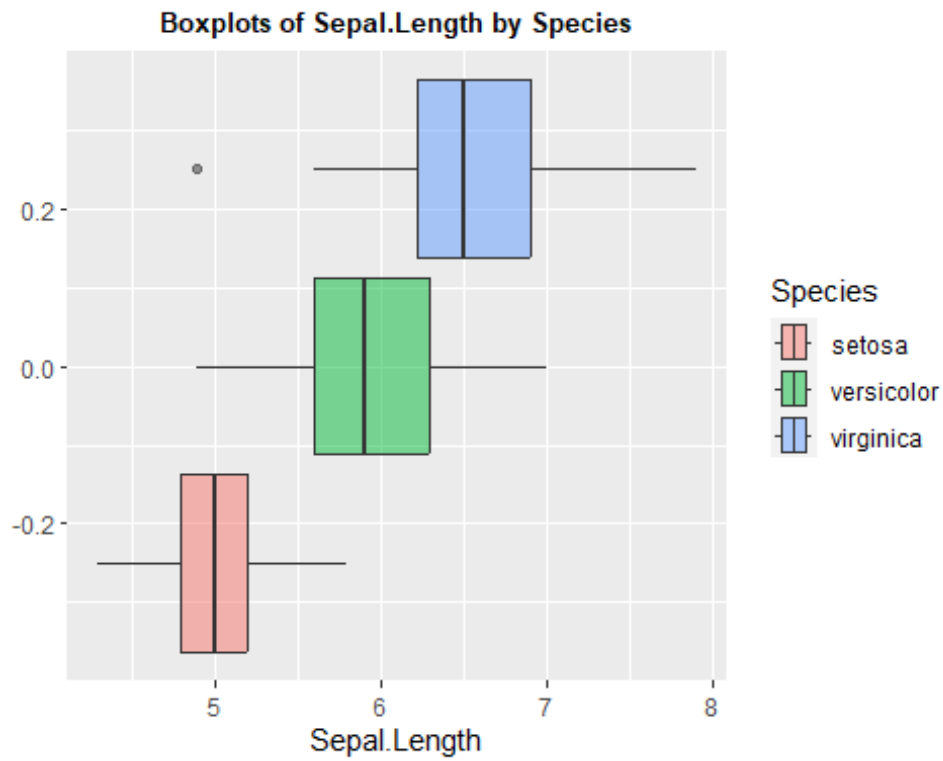
Get mean of individual variables for all species

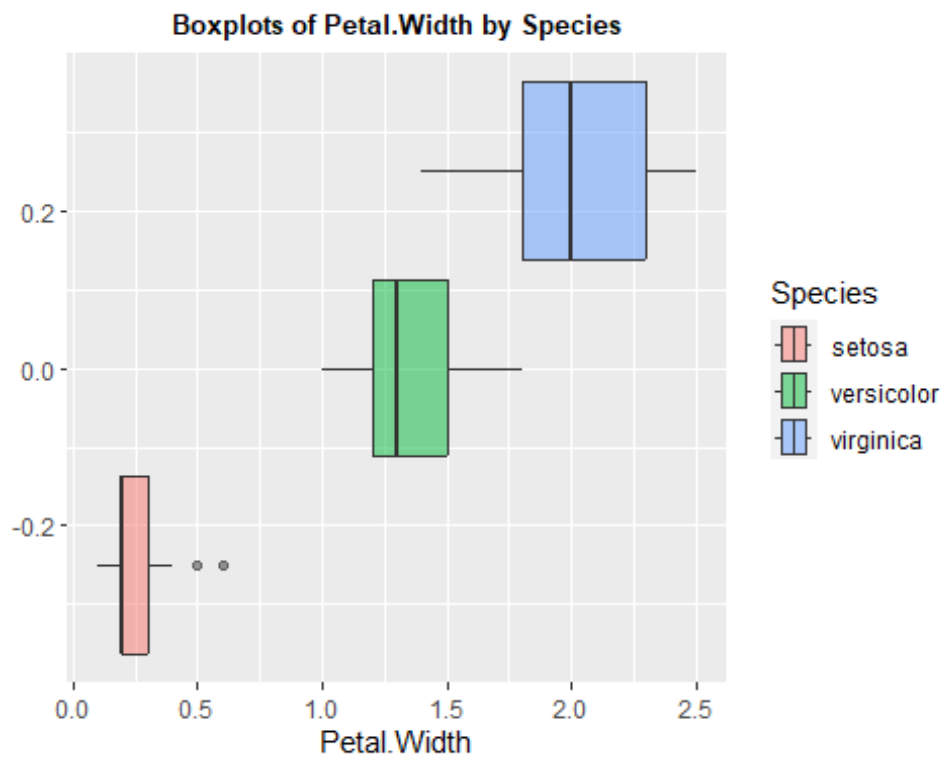
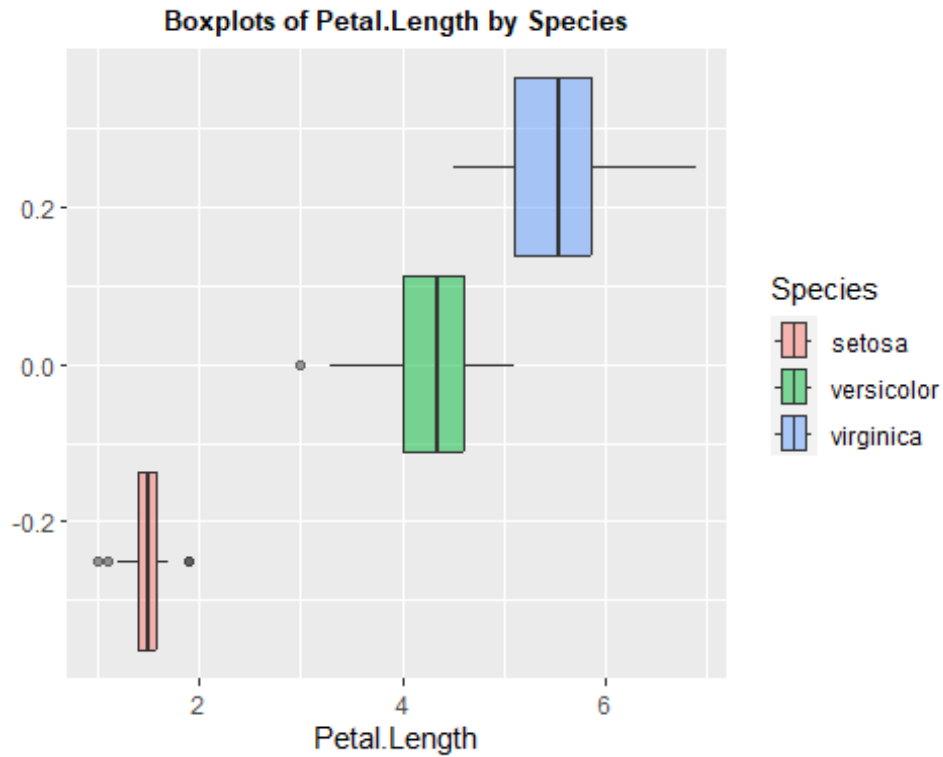
```
aggregate(. ~ Species, data = iris, mean)
```

```
##      Species Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1    setosa      5.006      3.428      1.462      0.246
## 2 versicolor      5.936      2.770      4.260      1.326
## 3  virginica      6.588      2.974      5.552      2.026
```

- Table showing the mean value of each variable (Sepal.Length, Sepal.Width, Petal.Length, and Petal.Width) for each of the three species (setosa, versicolor, and virginica).

```
for (i in 1:4){
  plot <- ggplot(iris, aes(x = iris[,i], fill = Species)) +
  geom_boxplot(alpha = 0.5) + ggtitle(paste0("Boxplots of ", names(iris)[i], "
  by Species")) + theme(plot.title = element_text(hjust = 0.5, face="bold",
  size = 10)) + xlab(names(iris[i]))
  print(plot)
}
```

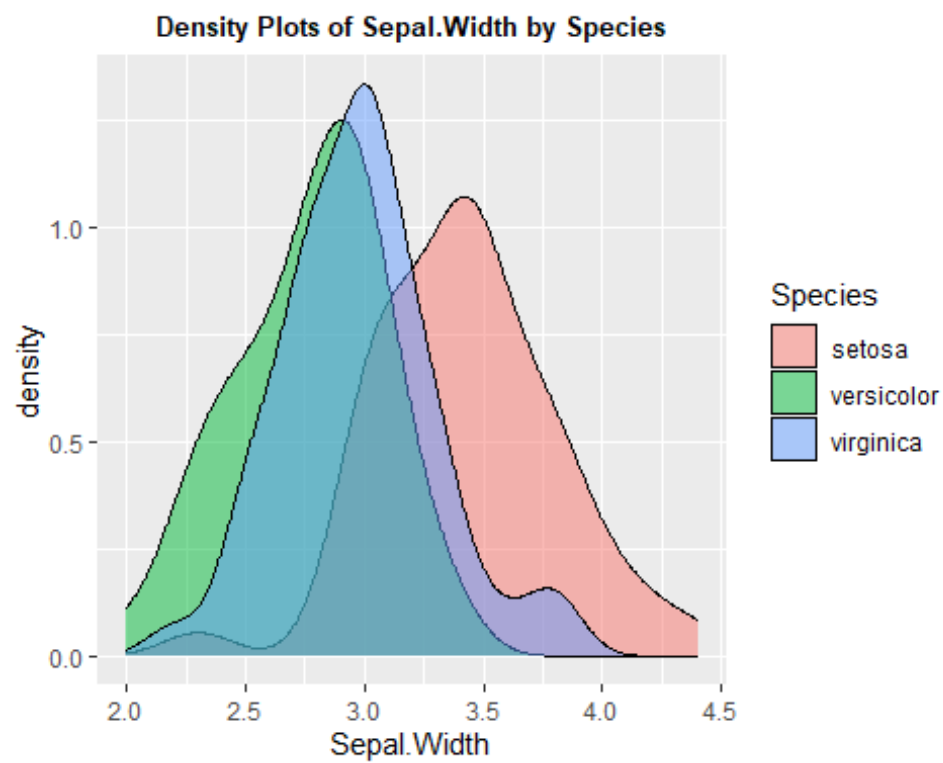
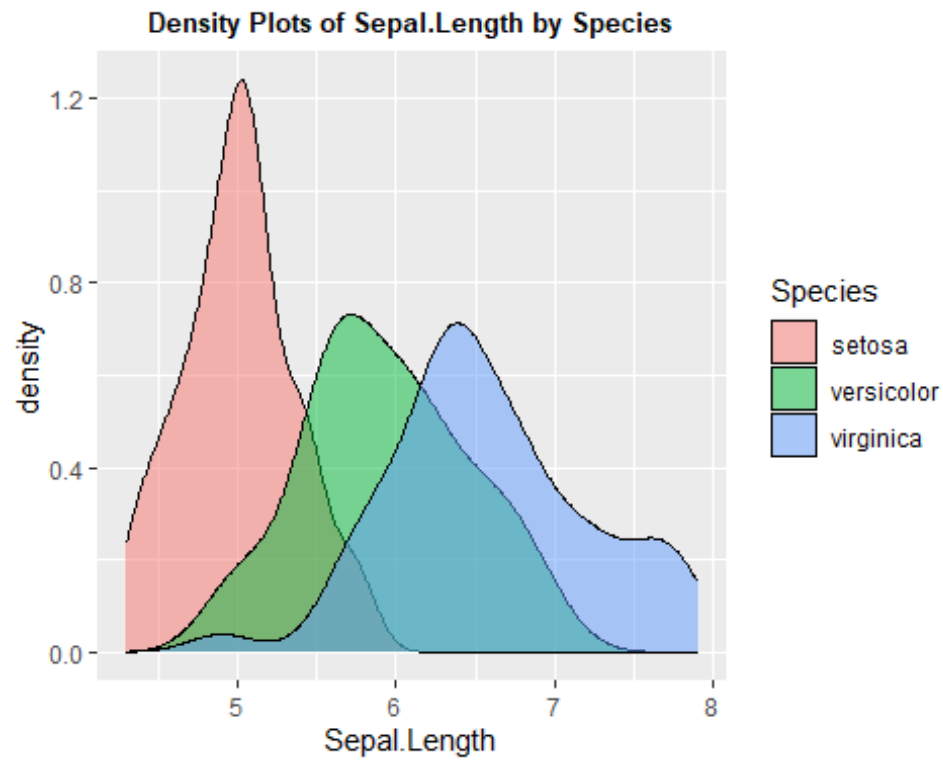


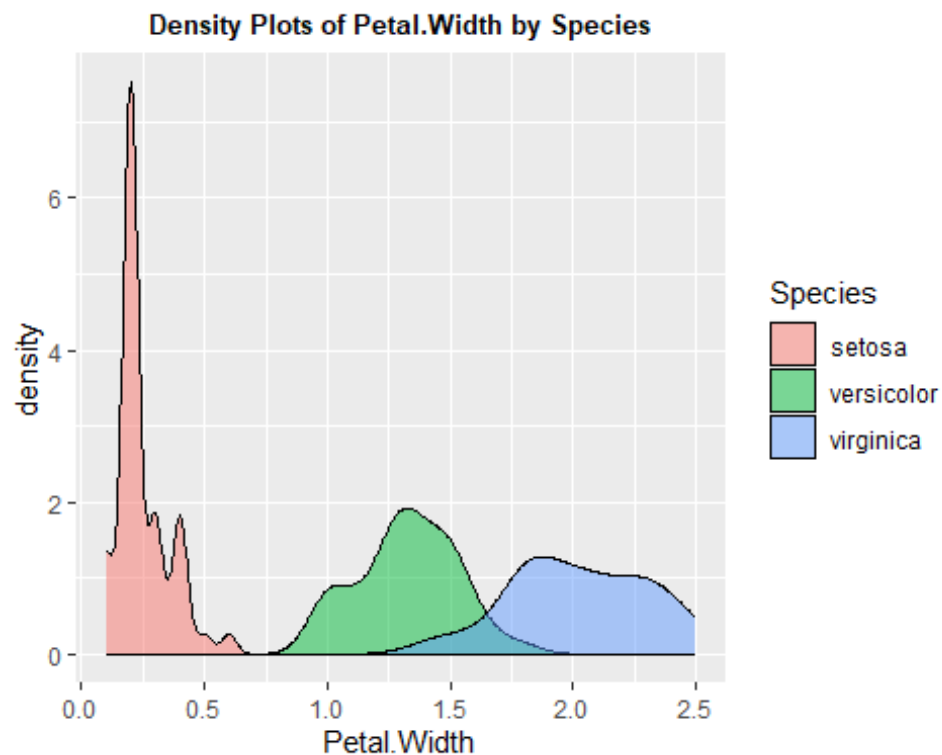
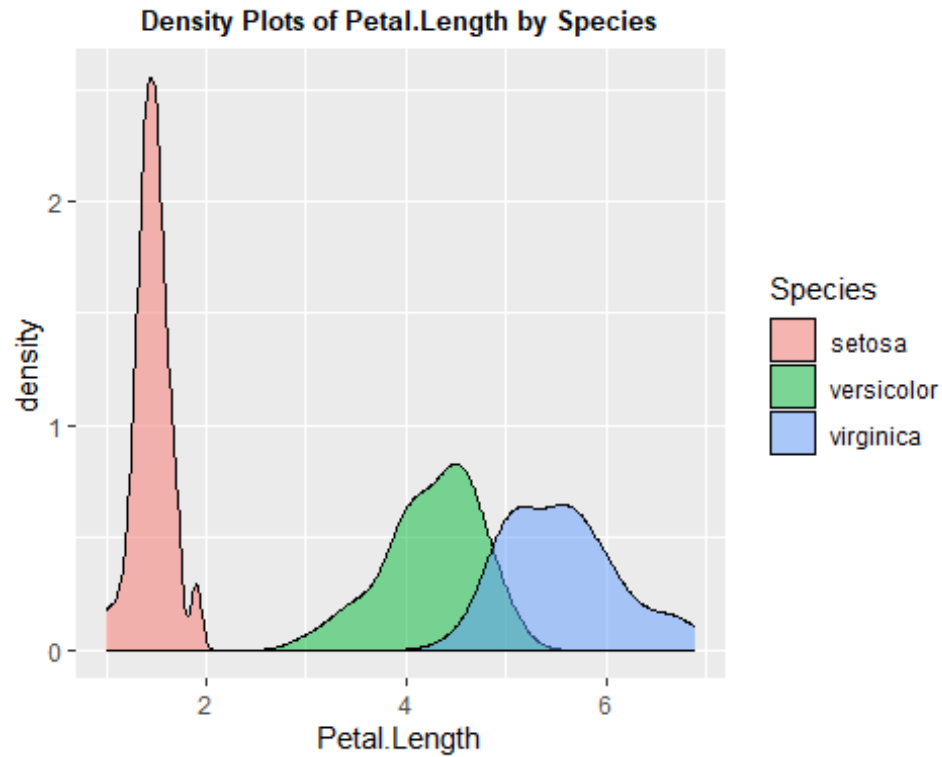


- For sepal length, the boxplot shows that the setosa species has the shortest sepal length, while the versicolor and virginica species have similar ranges with slightly longer median values.

- For sepal width, the setosa species has the widest range of values, while the versicolor and virginica species have similar ranges with slightly narrower median values. There are also some outliers in the setosa and virginica species.
- For petal length, the boxplot shows a clear distinction between the setosa species and the versicolor/virginica species, with the setosa species having much shorter petal lengths. The versicolor and virginica species have similar ranges, with the virginica species having slightly longer median values. There is an outlier in the setosa species.
- For petal width, the boxplot shows a similar pattern to petal length, with the setosa species having much smaller petal widths than the versicolor and virginica species. The virginica species has slightly longer median values than the versicolor species. There are outliers in the setosa species.

```
for (i in 1:4){
  plot <- ggplot(iris, aes(x = iris[,i], fill = Species)) +
    geom_density(alpha = 0.5) + ggtitle(paste0("Density Plots of ",
names(iris)[i], " by Species")) + theme(plot.title = element_text(hjust =
0.5, face="bold", size = 10)) + xlab(names(iris[i]))
  print(plot)
}
```





- The density curve of the iris data represents the distribution of all variables for three different species of iris flowers: setosa, versicolor, and virginica. Each species has its own distinct curve, indicating that they have different distributions of the variables. The setosa species has the shortest sepal length, with a peak around 5.0 cm, while

the versicolor species has a broader distribution of sepal length, with a peak around 6.0 cm. The virginica species has the longest sepal length, with a peak around 6.5 cm. The density curve provides a useful visualization of the distribution of the data and allows us to easily compare the sepal length characteristics across the three different species of iris.

- The sepal width for all three species appears to be approximately normally distributed, with the setosa species having the widest sepals and the versicolor and virginica species having similar and narrower distributions with peaks around 3.0-3.5 cm.
- The setosa species has the shortest petal length, while the virginica species has the longest, with a noticeable separation from the other two species in terms of petal length.
- The petal width distribution of the iris dataset has a unimodal shape, with the highest density occurring around 0.3 to 0.4 cm. The virginica species has the widest petal width, while the setosa species has the narrowest petal width

filtering a separate dataframe for just Specie Setosa

```
Setosa <- iris %>% filter(Species == "setosa")
```

Setosa

##	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
## 7	4.6	3.4	1.4	0.3	setosa
## 8	5.0	3.4	1.5	0.2	setosa
## 9	4.4	2.9	1.4	0.2	setosa
## 10	4.9	3.1	1.5	0.1	setosa
## 11	5.4	3.7	1.5	0.2	setosa
## 12	4.8	3.4	1.6	0.2	setosa
## 13	4.8	3.0	1.4	0.1	setosa
## 14	4.3	3.0	1.1	0.1	setosa
## 15	5.8	4.0	1.2	0.2	setosa
## 16	5.7	4.4	1.5	0.4	setosa
## 17	5.4	3.9	1.3	0.4	setosa
## 18	5.1	3.5	1.4	0.3	setosa
## 19	5.7	3.8	1.7	0.3	setosa
## 20	5.1	3.8	1.5	0.3	setosa
## 21	5.4	3.4	1.7	0.2	setosa
## 22	5.1	3.7	1.5	0.4	setosa
## 23	4.6	3.6	1.0	0.2	setosa
## 24	5.1	3.3	1.7	0.5	setosa
## 25	4.8	3.4	1.9	0.2	setosa
## 26	5.0	3.0	1.6	0.2	setosa
## 27	5.0	3.4	1.6	0.4	setosa

```
## 28      5.2      3.5      1.5      0.2 setosa
## 29      5.2      3.4      1.4      0.2 setosa
## 30      4.7      3.2      1.6      0.2 setosa
## 31      4.8      3.1      1.6      0.2 setosa
## 32      5.4      3.4      1.5      0.4 setosa
## 33      5.2      4.1      1.5      0.1 setosa
## 34      5.5      4.2      1.4      0.2 setosa
## 35      4.9      3.1      1.5      0.2 setosa
## 36      5.0      3.2      1.2      0.2 setosa
## 37      5.5      3.5      1.3      0.2 setosa
## 38      4.9      3.6      1.4      0.1 setosa
## 39      4.4      3.0      1.3      0.2 setosa
## 40      5.1      3.4      1.5      0.2 setosa
## 41      5.0      3.5      1.3      0.3 setosa
## 42      4.5      2.3      1.3      0.3 setosa
## 43      4.4      3.2      1.3      0.2 setosa
## 44      5.0      3.5      1.6      0.6 setosa
## 45      5.1      3.8      1.9      0.4 setosa
## 46      4.8      3.0      1.4      0.3 setosa
## 47      5.1      3.8      1.6      0.2 setosa
## 48      4.6      3.2      1.4      0.2 setosa
## 49      5.3      3.7      1.5      0.2 setosa
## 50      5.0      3.3      1.4      0.2 setosa
```

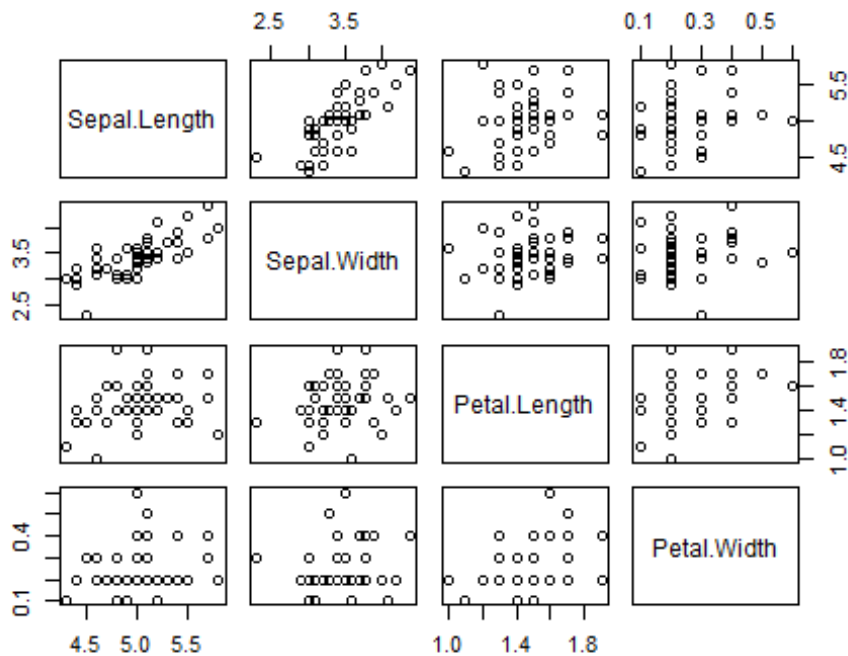
correlation matrix for specie

```
cor(Setosa[,1:4])
```

```
##          Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    1.0000000    0.7425467    0.2671758    0.2780984
## Sepal.Width      0.7425467    1.0000000    0.1777000    0.2327520
## Petal.Length     0.2671758    0.1777000    1.0000000    0.3316300
## Petal.Width      0.2780984    0.2327520    0.3316300    1.0000000
```

pairplot for setosa specie

```
pairs(Setosa[,1:4])
```



- The scatter plot for the setosa species shows that the sepal length and width are highly correlated, with a positive linear relationship. The petal length and width are also highly correlated, with a steeper linear relationship compared to the sepal measurements. Overall, the scatter plot for the setosa species indicates a clear separation between the two types of measurements (sepal and petal).

```
# Loop over columns 1 to 4 and create Q-Q plot and correlation test
for (i in 1:4) {
  colname <- names(Setosa)[i] # Get column name
  data <- Setosa %>% select(colname) # Select column data
  data <- data %>% arrange(.data[[colname]]) # Sort data in ascending order
  data <- data %>% mutate(probability = (row_number() - 0.5) / nrow(Setosa),
# Add probability column
                        standard_quantiles = qnorm(probability)) # Add
standard quantiles column

  # Plot Q-Q plot using ggplot2
  plot <- ggplot(data, aes(x = standard_quantiles, y = !!sym(colname))) +
    geom_point() + geom_abline(intercept = mean(Setosa[[colname]]), slope =
sd(Setosa[[colname]])) + labs(x = "Theoretical Quantiles", y = "Sample
Quantiles") +
    ggtitle(paste("Q-Q Plot for", colname, "in Setosa")) + theme(plot.title =
element_text(hjust = 0.5, face="bold", size = 10))
  print(plot)

  # Correlation test for normality
  rQ <- sum((data[[colname]] - mean(data[[colname]])) *
```

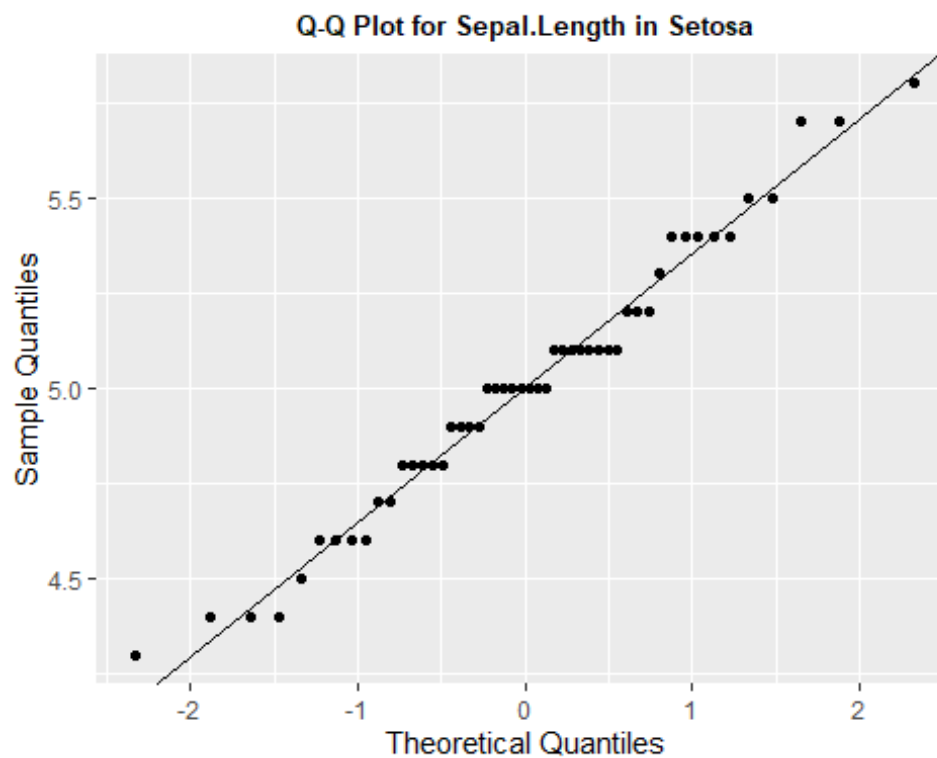
```

(data$standard_quantiles - mean(data$standard_quantiles))) /
  sqrt((sum((data[[colname]] - mean(data[[colname]]))^2)) *
(sum((data$standard_quantiles - mean(data$standard_quantiles))^2)))

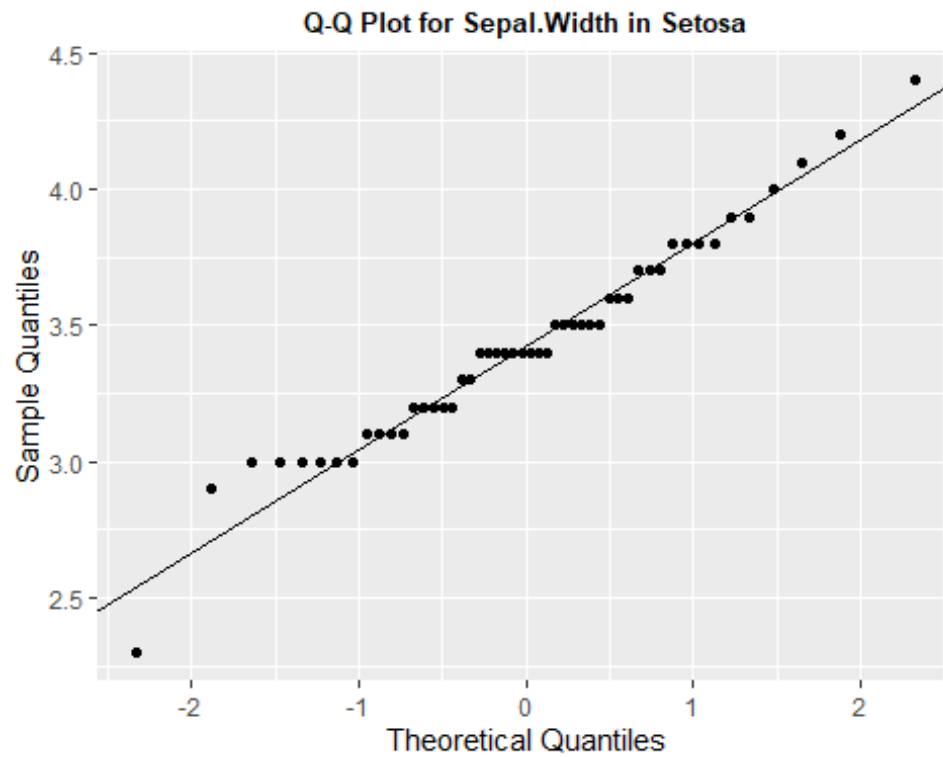
  print(paste("Correlation coefficient for", colname, "in Setosa is:",
round(rQ, 4)))
}

## Warning: Using an external vector in selections was deprecated in
tidyselect 1.1.0.
## i Please use `all_of()` or `any_of()` instead.
## # Was:
## data %>% select(colname)
##
## # Now:
## data %>% select(all_of(colname))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.

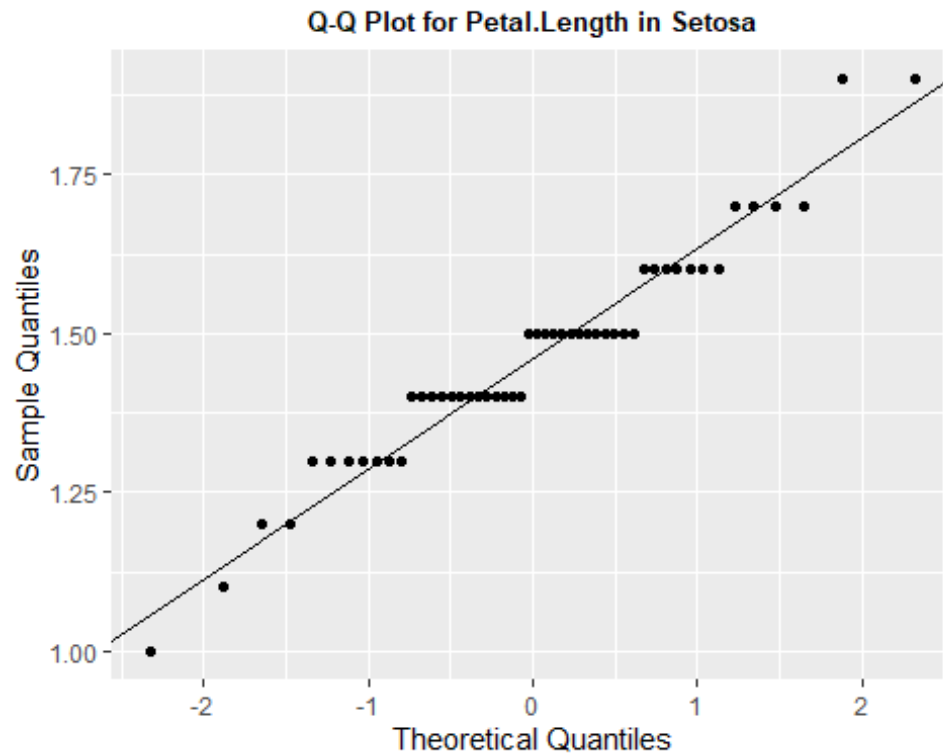
```



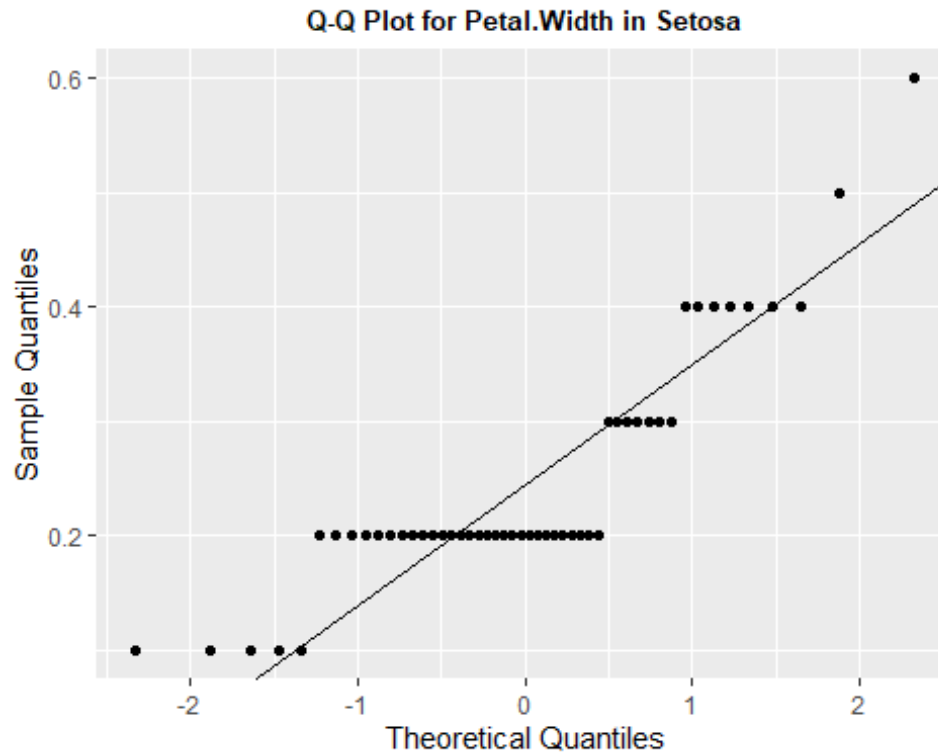
```
## [1] "Correlation coefficient for Sepal.Length in Setosa is: 0.9905"
```



```
## [1] "Correlation coefficient for Sepal.Width in Setosa is: 0.983"
```



```
## [1] "Correlation coefficient for Petal.Length in Setosa is: 0.9754"
```



```
## [1] "Correlation coefficient for Petal.Width in Setosa is: 0.8928"
```

- At a significance level of 0.05 and sample size $n=50$, if we refer to the critical points for the Q-Q plot correlation coefficient test for normality table, we realise that the entry is 0.9768. Since the correlation coefficient for Sepal.length and Sepal.Width in Setosa is greater than 0.9768, we do not reject the hypothesis of normality.
- Meanwhile, Correlation coefficient for Petal.Length and Petal.Width in Setosa is less than 0.9768. So we reject the hypothesis of normality.

filtering a separate dataframe for just Specie virginica

```
virginica <- iris %>% filter(Species == "virginica")
virginica
```

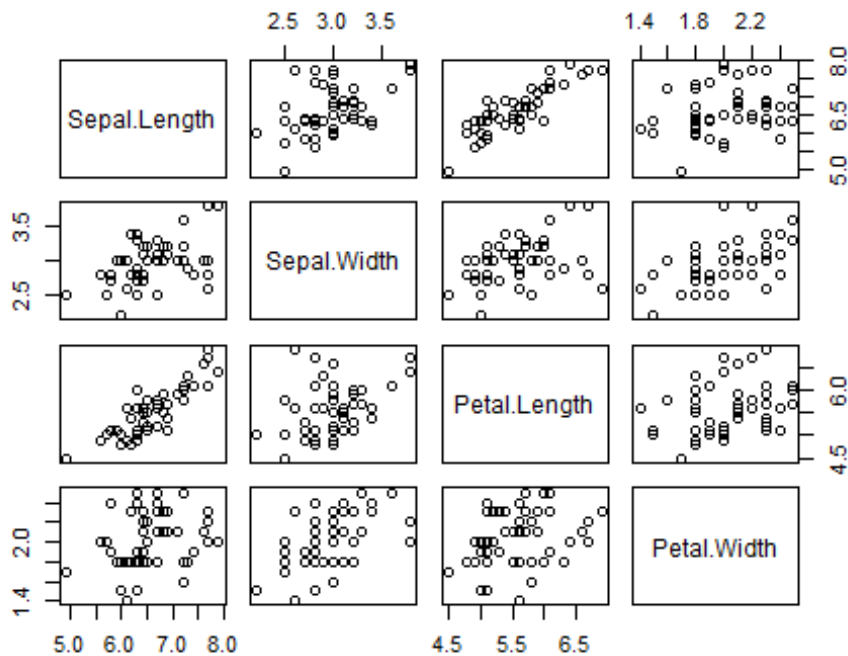
```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width  Species
## 1         6.3         3.3         6.0         2.5 virginica
## 2         5.8         2.7         5.1         1.9 virginica
## 3         7.1         3.0         5.9         2.1 virginica
## 4         6.3         2.9         5.6         1.8 virginica
## 5         6.5         3.0         5.8         2.2 virginica
## 6         7.6         3.0         6.6         2.1 virginica
## 7         4.9         2.5         4.5         1.7 virginica
## 8         7.3         2.9         6.3         1.8 virginica
## 9         6.7         2.5         5.8         1.8 virginica
## 10        7.2         3.6         6.1         2.5 virginica
## 11        6.5         3.2         5.1         2.0 virginica
## 12        6.4         2.7         5.3         1.9 virginica
## 13        6.8         3.0         5.5         2.1 virginica
```

```
## 14      5.7      2.5      5.0      2.0 virginica
## 15      5.8      2.8      5.1      2.4 virginica
## 16      6.4      3.2      5.3      2.3 virginica
## 17      6.5      3.0      5.5      1.8 virginica
## 18      7.7      3.8      6.7      2.2 virginica
## 19      7.7      2.6      6.9      2.3 virginica
## 20      6.0      2.2      5.0      1.5 virginica
## 21      6.9      3.2      5.7      2.3 virginica
## 22      5.6      2.8      4.9      2.0 virginica
## 23      7.7      2.8      6.7      2.0 virginica
## 24      6.3      2.7      4.9      1.8 virginica
## 25      6.7      3.3      5.7      2.1 virginica
## 26      7.2      3.2      6.0      1.8 virginica
## 27      6.2      2.8      4.8      1.8 virginica
## 28      6.1      3.0      4.9      1.8 virginica
## 29      6.4      2.8      5.6      2.1 virginica
## 30      7.2      3.0      5.8      1.6 virginica
## 31      7.4      2.8      6.1      1.9 virginica
## 32      7.9      3.8      6.4      2.0 virginica
## 33      6.4      2.8      5.6      2.2 virginica
## 34      6.3      2.8      5.1      1.5 virginica
## 35      6.1      2.6      5.6      1.4 virginica
## 36      7.7      3.0      6.1      2.3 virginica
## 37      6.3      3.4      5.6      2.4 virginica
## 38      6.4      3.1      5.5      1.8 virginica
## 39      6.0      3.0      4.8      1.8 virginica
## 40      6.9      3.1      5.4      2.1 virginica
## 41      6.7      3.1      5.6      2.4 virginica
## 42      6.9      3.1      5.1      2.3 virginica
## 43      5.8      2.7      5.1      1.9 virginica
## 44      6.8      3.2      5.9      2.3 virginica
## 45      6.7      3.3      5.7      2.5 virginica
## 46      6.7      3.0      5.2      2.3 virginica
## 47      6.3      2.5      5.0      1.9 virginica
## 48      6.5      3.0      5.2      2.0 virginica
## 49      6.2      3.4      5.4      2.3 virginica
## 50      5.9      3.0      5.1      1.8 virginica
```

```
cor(virginica[,1:4])
```

```
##          Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length      1.0000000      0.4572278      0.8642247      0.2811077
## Sepal.Width        0.4572278      1.0000000      0.4010446      0.5377280
## Petal.Length       0.8642247      0.4010446      1.0000000      0.3221082
## Petal.Width        0.2811077      0.5377280      0.3221082      1.0000000
```

```
pairs(virginica[,1:4])
```



- The correlation plot for the Virginica species shows a strong positive correlation between petal length and petal width, as well as between sepal length and petal length.

```
# Loop over columns 1 to 4 and create Q-Q plot and correlation test
for (i in 1:4) {
  colname <- names(virginica)[i] # Get column name
  data <- virginica %>% select(colname) # Select column data
  data <- data %>% arrange(.data[[colname]]) # Sort data in ascending order
  data <- data %>% mutate(probability = (row_number() - 0.5) /
nrow(virginica), # Add probability column
                          standard_quantiles = qnorm(probability)) # Add
standard quantiles column

  # Plot Q-Q plot using ggplot2
  plot <- ggplot(data, aes(x = standard_quantiles, y = !!sym(colname))) +
    geom_point() + geom_abline(intercept = mean(virginica[[colname]]), slope
= sd(virginica[[colname]])) + labs(x = "Theoretical Quantiles", y = "Sample
Quantiles") +
    ggtitle(paste("Q-Q Plot for", colname, "in virginica")) +
  theme(plot.title = element_text(hjust = 0.5, face="bold", size = 10))
  print(plot)

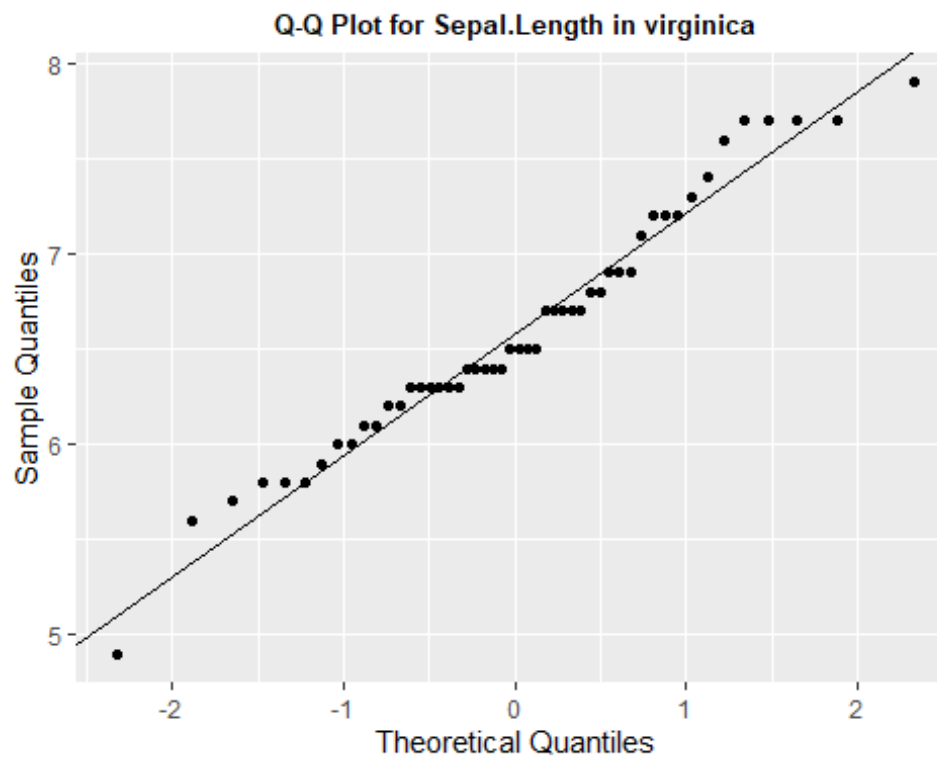
  # Correlation test for normality
  rQ <- sum((data[[colname]] - mean(data[[colname]])) *
(data$standard_quantiles - mean(data$standard_quantiles))) /
  sqrt((sum((data[[colname]] - mean(data[[colname]]))^2)) *

```

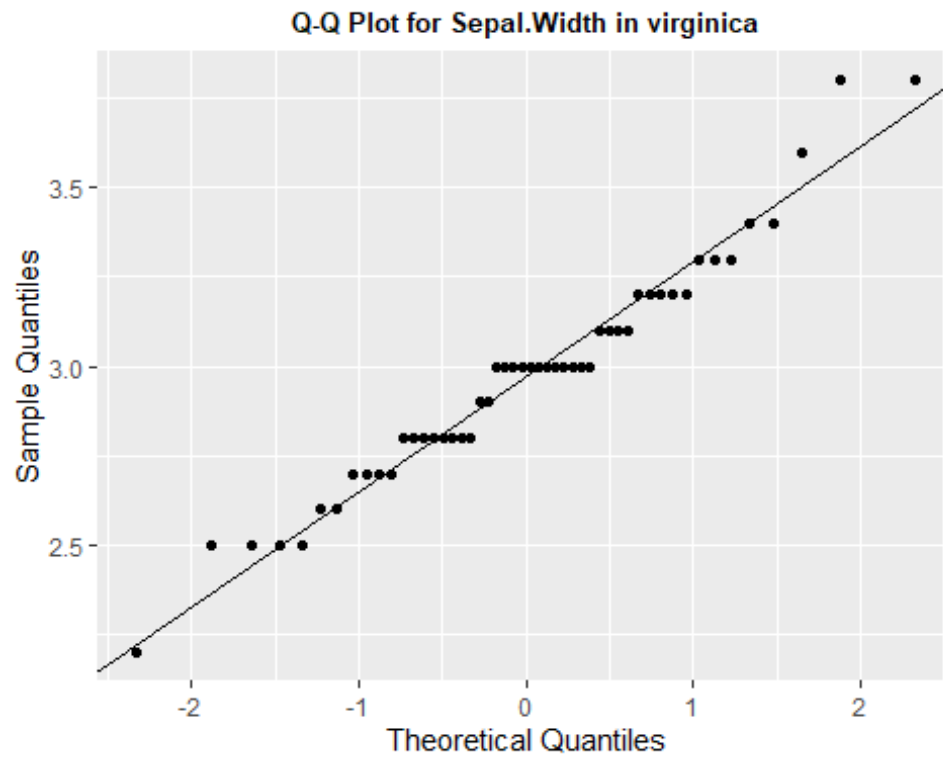


```
(sum((data$standard_quantiles - mean(data$standard_quantiles))^2)))

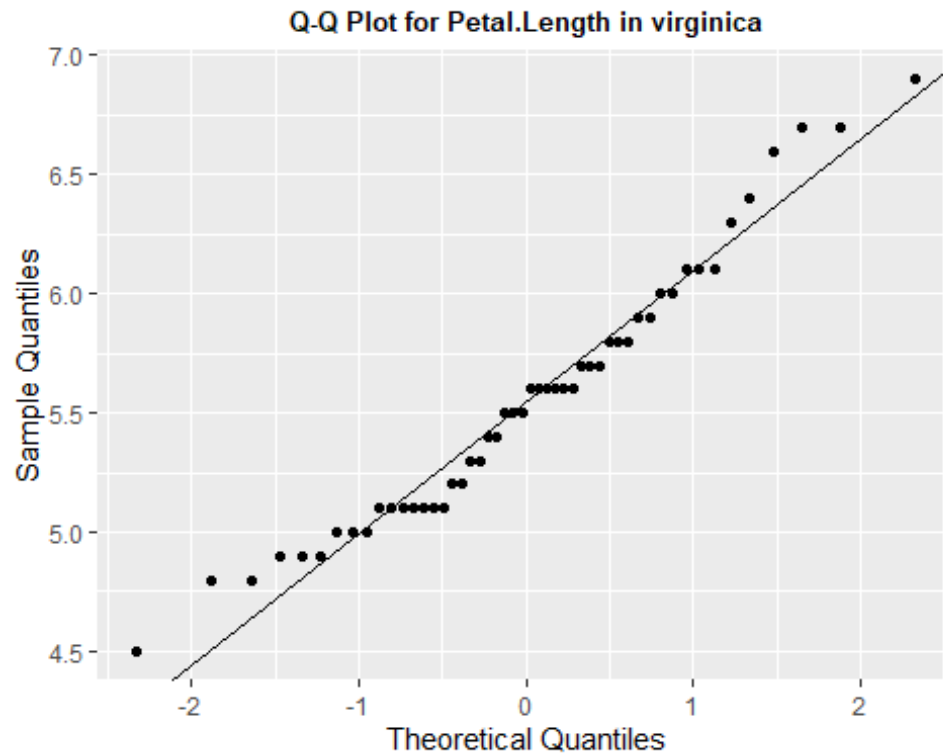
  print(paste("Correlation coefficient for", colname, "in virginica is:",
round(rQ, 4)))
}
```



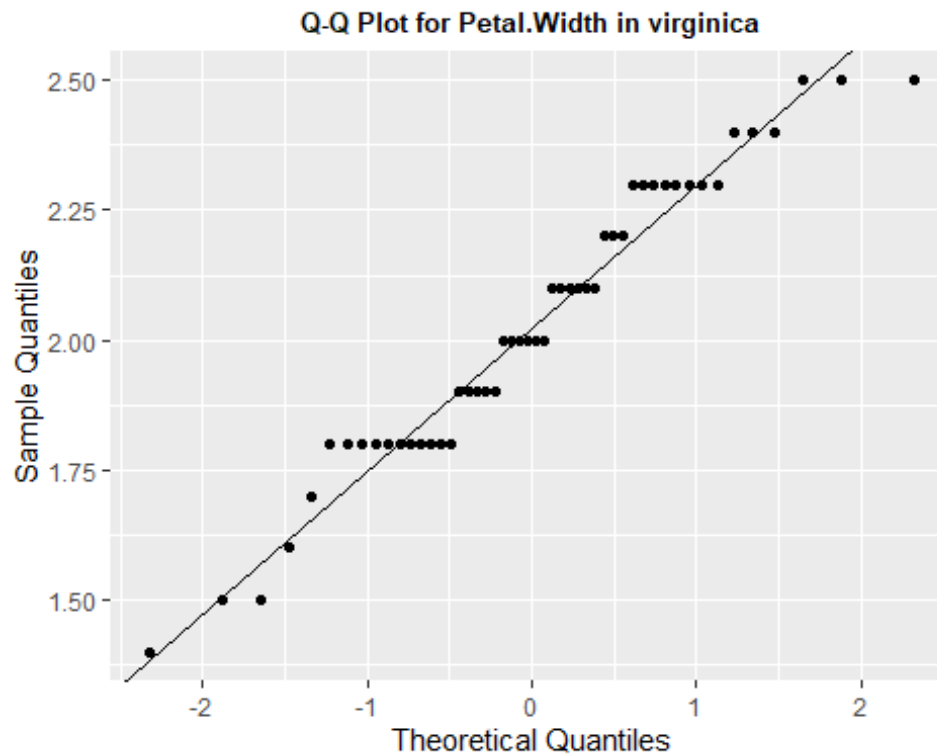
```
## [1] "Correlation coefficient for Sepal.Length in virginica is: 0.9856"
```



```
## [1] "Correlation coefficient for Sepal.Width in virginica is: 0.9828"
```



```
## [1] "Correlation coefficient for Petal.Length in virginica is: 0.9821"
```



```
## [1] "Correlation coefficient for Petal.Width in virginica is: 0.9819"
```

- At a significance level of 0.05 and sample size $n=50$, if we refer to the critical points for the Q-Q plot correlation coefficient test for normality table, we realise that the entry is 0.9768. Since the correlation coefficient for all variables in virginica are greater than 0.9768, we do not reject the hypothesis of normality.

```
# filtering a separate dataframe for just Specie versicolor
```

```
versicolor <- iris %>% filter(Species == "versicolor")
versicolor
```

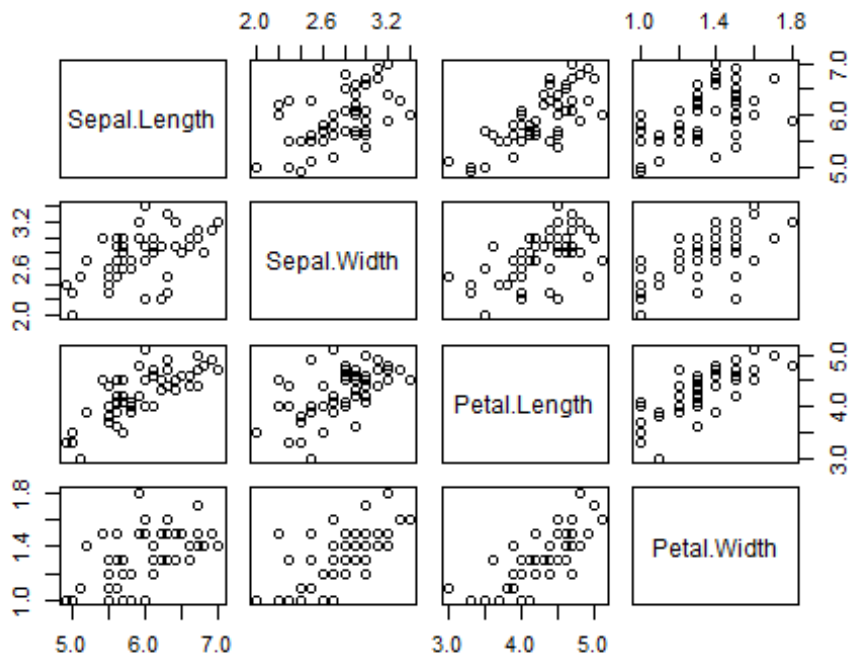
```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 1           7.0         3.2         4.7         1.4 versicolor
## 2           6.4         3.2         4.5         1.5 versicolor
## 3           6.9         3.1         4.9         1.5 versicolor
## 4           5.5         2.3         4.0         1.3 versicolor
## 5           6.5         2.8         4.6         1.5 versicolor
## 6           5.7         2.8         4.5         1.3 versicolor
## 7           6.3         3.3         4.7         1.6 versicolor
## 8           4.9         2.4         3.3         1.0 versicolor
## 9           6.6         2.9         4.6         1.3 versicolor
## 10          5.2         2.7         3.9         1.4 versicolor
## 11          5.0         2.0         3.5         1.0 versicolor
## 12          5.9         3.0         4.2         1.5 versicolor
## 13          6.0         2.2         4.0         1.0 versicolor
## 14          6.1         2.9         4.7         1.4 versicolor
## 15          5.6         2.9         3.6         1.3 versicolor
## 16          6.7         3.1         4.4         1.4 versicolor
```

```
## 17      5.6      3.0      4.5      1.5 versicolor
## 18      5.8      2.7      4.1      1.0 versicolor
## 19      6.2      2.2      4.5      1.5 versicolor
## 20      5.6      2.5      3.9      1.1 versicolor
## 21      5.9      3.2      4.8      1.8 versicolor
## 22      6.1      2.8      4.0      1.3 versicolor
## 23      6.3      2.5      4.9      1.5 versicolor
## 24      6.1      2.8      4.7      1.2 versicolor
## 25      6.4      2.9      4.3      1.3 versicolor
## 26      6.6      3.0      4.4      1.4 versicolor
## 27      6.8      2.8      4.8      1.4 versicolor
## 28      6.7      3.0      5.0      1.7 versicolor
## 29      6.0      2.9      4.5      1.5 versicolor
## 30      5.7      2.6      3.5      1.0 versicolor
## 31      5.5      2.4      3.8      1.1 versicolor
## 32      5.5      2.4      3.7      1.0 versicolor
## 33      5.8      2.7      3.9      1.2 versicolor
## 34      6.0      2.7      5.1      1.6 versicolor
## 35      5.4      3.0      4.5      1.5 versicolor
## 36      6.0      3.4      4.5      1.6 versicolor
## 37      6.7      3.1      4.7      1.5 versicolor
## 38      6.3      2.3      4.4      1.3 versicolor
## 39      5.6      3.0      4.1      1.3 versicolor
## 40      5.5      2.5      4.0      1.3 versicolor
## 41      5.5      2.6      4.4      1.2 versicolor
## 42      6.1      3.0      4.6      1.4 versicolor
## 43      5.8      2.6      4.0      1.2 versicolor
## 44      5.0      2.3      3.3      1.0 versicolor
## 45      5.6      2.7      4.2      1.3 versicolor
## 46      5.7      3.0      4.2      1.2 versicolor
## 47      5.7      2.9      4.2      1.3 versicolor
## 48      6.2      2.9      4.3      1.3 versicolor
## 49      5.1      2.5      3.0      1.1 versicolor
## 50      5.7      2.8      4.1      1.3 versicolor
```

```
cor(versicolor[,1:4])
```

```
##           Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    1.0000000    0.5259107    0.7540490    0.5464611
## Sepal.Width      0.5259107    1.0000000    0.5605221    0.6639987
## Petal.Length     0.7540490    0.5605221    1.0000000    0.7866681
## Petal.Width      0.5464611    0.6639987    0.7866681    1.0000000
```

```
pairs(versicolor[,1:4])
```



- Sepal Length vs. Sepal Width shows a moderately positive linear relationship between the two variables, with some variation in the data points. Petal Length vs. Petal Width shows a strong positive linear relationship between the two variables, with very little variation in the data points. The scatter plot for Sepal Length vs. Petal Length shows a moderately positive linear relationship between the two variables, with some variation in the data points. Plot for Sepal Width vs. Petal Width shows a weak positive linear relationship between the two variables, with some variation in the data points.

```
# Loop over columns 1 to 4 and create Q-Q plot and correlation test
for (i in 1:4) {
  colname <- names(versicolor)[i] # Get column name
  data <- versicolor %>% select(colname) # Select column data
  data <- data %>% arrange(.data[[colname]]) # Sort data in ascending order
  data <- data %>% mutate(probability = (row_number() - 0.5) /
nrow(versicolor), # Add probability column
                        standard_quantiles = qnorm(probability)) # Add
standard quantiles column

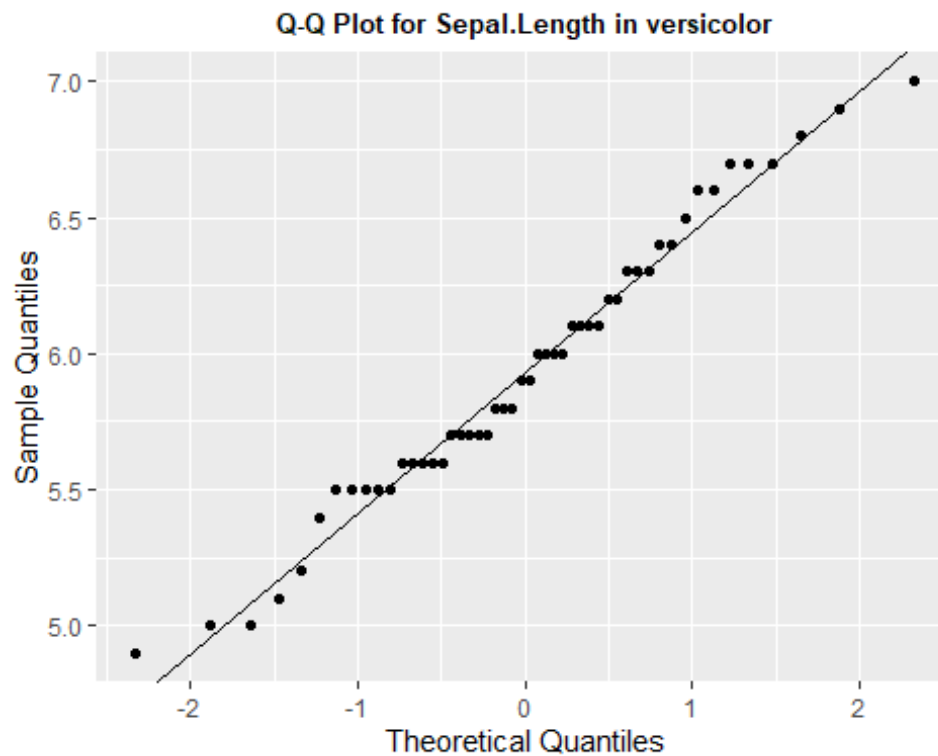
  # Plot Q-Q plot using ggplot2
  plot <- ggplot(data, aes(x = standard_quantiles, y = !!sym(colname))) +
    geom_point() + geom_abline(intercept = mean(versicolor[[colname]]), slope
= sd(versicolor[[colname]])) + labs(x = "Theoretical Quantiles", y = "Sample
Quantiles") +
    ggtitle(paste("Q-Q Plot for", colname, "in versicolor")) +
  theme(plot.title = element_text(hjust = 0.5, face="bold", size = 10))
  print(plot)
```

```

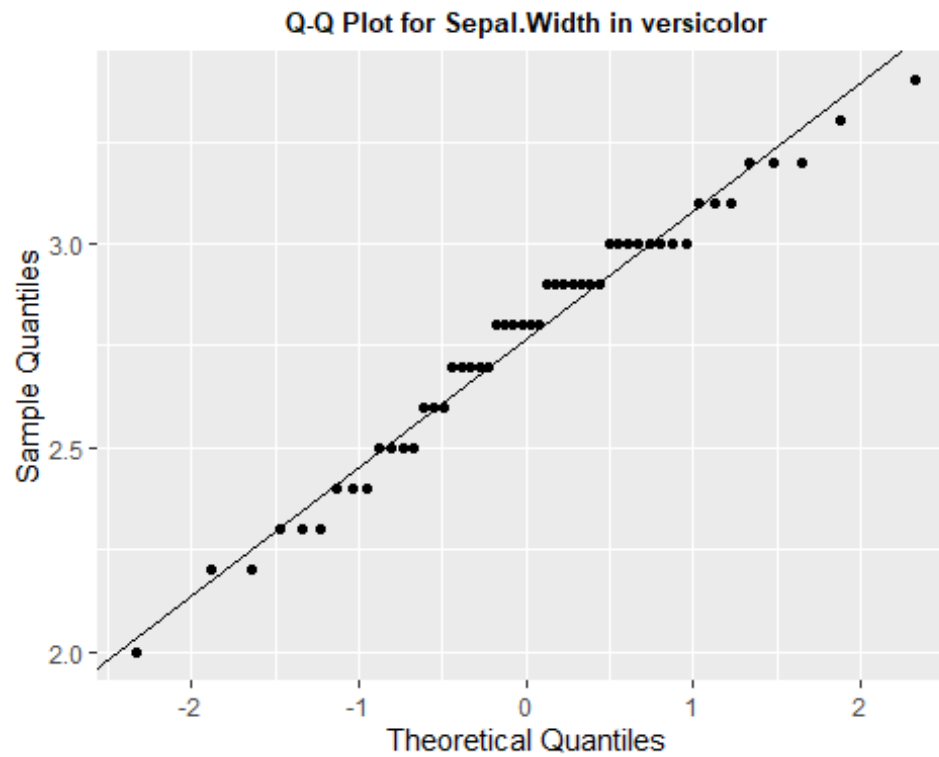
# Correlation test for normality
rQ <- sum((data[[colname]] - mean(data[[colname]])) *
(data$standard_quantiles - mean(data$standard_quantiles))) /
  sqrt((sum((data[[colname]] - mean(data[[colname]]))^2) *
(sum((data$standard_quantiles - mean(data$standard_quantiles))^2)))

print(paste("Correlation coefficient for", colname, "in versicolor is:",
round(rQ, 4)))
}

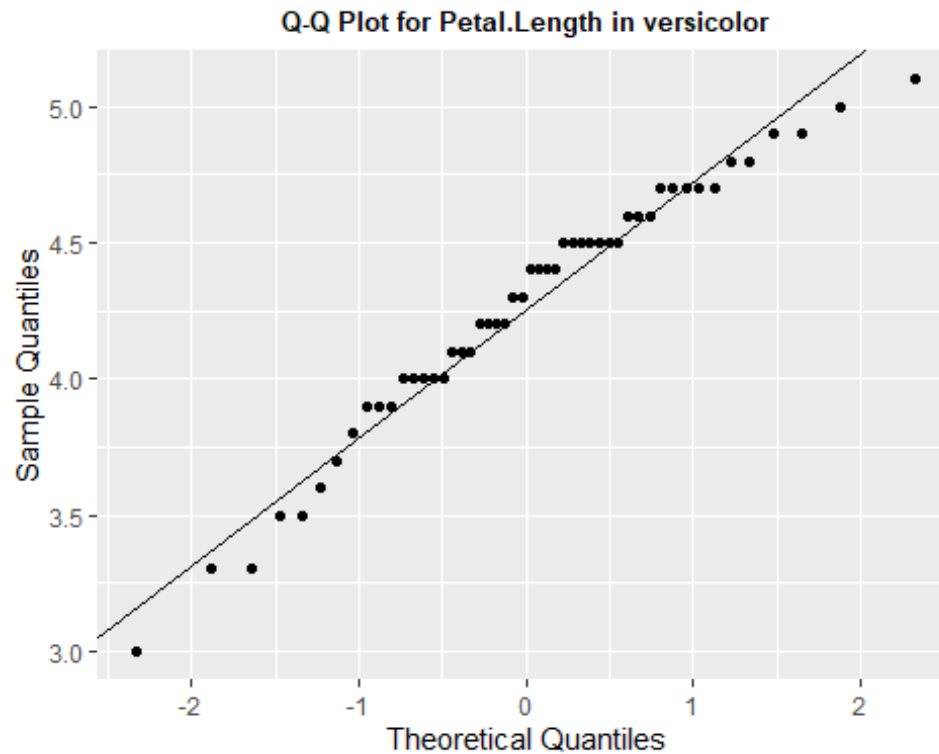
```



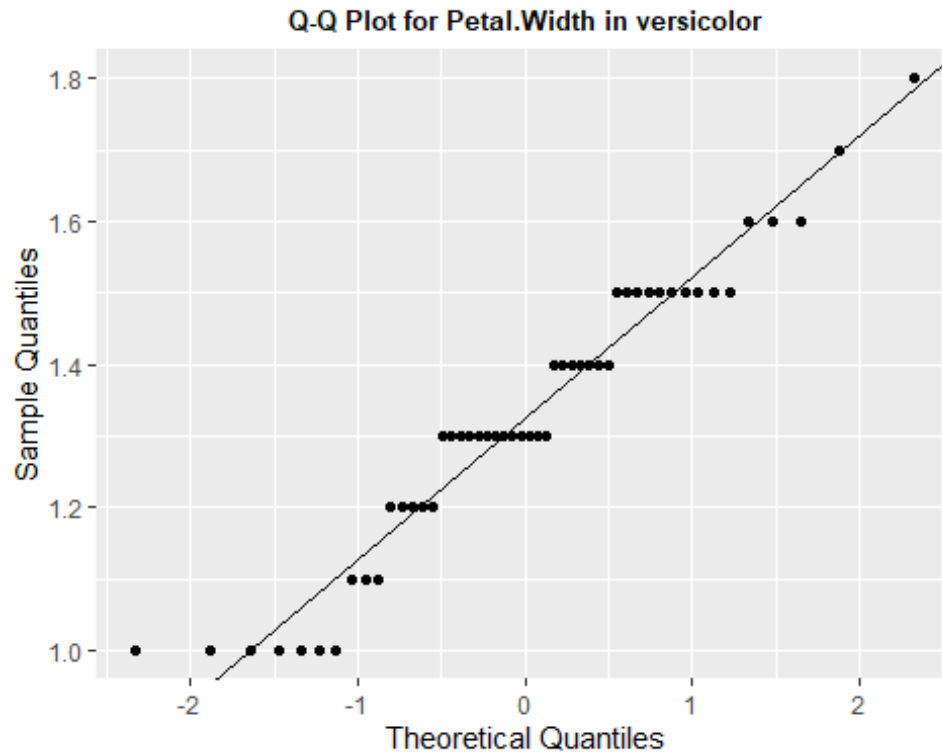
```
## [1] "Correlation coefficient for Sepal.Length in versicolor is: 0.9911"
```



```
## [1] "Correlation coefficient for Sepal.Width in versicolor is: 0.9879"
```



```
## [1] "Correlation coefficient for Petal.Length in versicolor is: 0.9837"
```



```
## [1] "Correlation coefficient for Petal.Width in versicolor is: 0.9755"
```

- At a significance level of 0.05 and sample size $n=50$, if we refer to the critical points for the Q-Q plot correlation coefficient test for normality table, we realise that the entry is 0.9768. Since the correlation coefficient for Sepal.length, Sepal.Width and Petal.Length in versicolor is greater than 0.9768, we do not reject the hypothesis of normality.
- Meanwhile, Correlation coefficient for Petal.Width in versicolor is less than 0.9768. So we reject the hypothesis of normality.

UNIVARIATE NORMALITY TEST

The Shapiro-Wilk test is a univariate test for normality, which means it can only be used to test the normality of one variable at a time. The iris dataset has four variables (sepal length, sepal width, petal length, and petal width), so we cannot use the Shapiro-Wilk test on the entire dataset. We can, however, use it on each variable individually to test for normality. If the p-value is less than the significance level (commonly set at 0.05), we reject the null hypothesis and conclude that the sample does not come from a normal distribution.

```
# shapiro-Wilk test for Setosa Specie
```

```
shapiro.test(Setosa$Sepal.Length)
```

```
##
## Shapiro-Wilk normality test
##
```



```
## data: Setosa$Sepal.Length
## W = 0.9777, p-value = 0.4595

shapiro.test(Setosa$Sepal.Width)

##
## Shapiro-Wilk normality test
##
## data: Setosa$Sepal.Width
## W = 0.97172, p-value = 0.2715

shapiro.test(Setosa$Petal.Length)

##
## Shapiro-Wilk normality test
##
## data: Setosa$Petal.Length
## W = 0.95498, p-value = 0.05481

shapiro.test(Setosa$Petal.Width)

##
## Shapiro-Wilk normality test
##
## data: Setosa$Petal.Width
## W = 0.79976, p-value = 8.659e-07
```

Using the Shapiro-Wilk test, the p_value for setosa Petal.Width is less than 0.05, so we reject the null hypothesis. There is less than a 5% chance that the deviation from normality was due to random sampling. Meanwhile the p values for the other variables (Sepal.Length, Sepal.Width and Petal.Length) are greater than 0.05 which means that the sample may come from a random distribution.

Shapiro-Wilk test for virginica specie

```
shapiro.test(virginica$Sepal.Length)

##
## Shapiro-Wilk normality test
##
## data: virginica$Sepal.Length
## W = 0.97118, p-value = 0.2583

shapiro.test(virginica$Sepal.Width)

##
## Shapiro-Wilk normality test
##
## data: virginica$Sepal.Width
## W = 0.96739, p-value = 0.1809

shapiro.test(virginica$Petal.Length)
```

```
##
## Shapiro-Wilk normality test
##
## data:  virginica$Petal.Length
## W = 0.96219, p-value = 0.1098

shapiro.test(virginica$Petal.Width)

##
## Shapiro-Wilk normality test
##
## data:  virginica$Petal.Width
## W = 0.95977, p-value = 0.08695
```

- Using the Shapiro-Wilk test, none of the p-values for the virginica specie for the different variables is less than 0.05, so we cannot reject the null hypothesis and conclude that the sample may come from a random distribution.

Shapiro-Wilk test for versicolor specie

```
shapiro.test(versicolor$Sepal.Length)

##
## Shapiro-Wilk normality test
##
## data:  versicolor$Sepal.Length
## W = 0.97784, p-value = 0.4647

shapiro.test(versicolor$Sepal.Width)

##
## Shapiro-Wilk normality test
##
## data:  versicolor$Sepal.Width
## W = 0.97413, p-value = 0.338

shapiro.test(versicolor$Petal.Length)

##
## Shapiro-Wilk normality test
##
## data:  versicolor$Petal.Length
## W = 0.966, p-value = 0.1585

shapiro.test(versicolor$Petal.Width)

##
## Shapiro-Wilk normality test
##
## data:  versicolor$Petal.Width
## W = 0.94763, p-value = 0.02728
```

Using the Shapiro-Wilk test, the p -value for versicolor Petal.Width is less than 0.05, so we reject the null hypothesis. There is less than a 5% chance that the deviation from normality was due to random sampling. Meanwhile the p values for the other variables (Sepal.Length, Sepal.Width and Petal.Length) are greater than 0.05.

The Anderson-Darling test is a statistical test used to assess whether a sample of data comes from a known probability distribution, such as the normal distribution. The null hypothesis of the test is that the sample comes from the specified distribution, and a low p -value indicates that the null hypothesis should be rejected in favor of an alternative hypothesis that the sample does not come from the specified distribution.

```
library(nortest)

# Anderson-Darling test for setosa specie

ad.test(Setosa$Sepal.Length)

##
## Anderson-Darling normality test
##
## data: Setosa$Sepal.Length
## A = 0.40799, p-value = 0.3352

ad.test(Setosa$Sepal.Width)

##
## Anderson-Darling normality test
##
## data: Setosa$Sepal.Width
## A = 0.49096, p-value = 0.2102

ad.test(Setosa$Petal.Length)

##
## Anderson-Darling normality test
##
## data: Setosa$Petal.Length
## A = 1.0073, p-value = 0.01079

ad.test(Setosa$Petal.Width)

##
## Anderson-Darling normality test
##
## data: Setosa$Petal.Width
## A = 4.7148, p-value = 7.437e-12
```

Using the Anderson-Darling Test, the p -value for setosa Petal.Width and Petal.Length is less than 0.05, so we reject the null hypothesis. There is less than a 5% chance that the deviation from normality was due to random sampling. Meanwhile the p values for the other

variables (Sepal.Length and Sepal.Width) are greater than 0.05 which means that the sample may come from a random distribution.

```
# Anderson-Darling test for virginica specie
```

```
ad.test(virginica$Sepal.Length)

##
##  Anderson-Darling normality test
##
## data:  virginica$Sepal.Length
## A = 0.55164, p-value = 0.1475

ad.test(virginica$Sepal.Width)

##
##  Anderson-Darling normality test
##
## data:  virginica$Sepal.Width
## A = 0.61821, p-value = 0.1018

ad.test(virginica$Petal.Length)

##
##  Anderson-Darling normality test
##
## data:  virginica$Petal.Length
## A = 0.60896, p-value = 0.1074

ad.test(virginica$Petal.Width)

##
##  Anderson-Darling normality test
##
## data:  virginica$Petal.Width
## A = 0.73879, p-value = 0.05076
```

- Using the Anderson-Darling test, none of the p-values for the virginica specie for the different variables is less than 0.05, so we cannot reject the null hypothesis and conclude that the sample may come from a random distribution

```
# Anderson-Darling test for versicolor specie
```

```
ad.test(versicolor$Sepal.Length)

##
##  Anderson-Darling normality test
##
## data:  versicolor$Sepal.Length
## A = 0.36084, p-value = 0.4333

ad.test(versicolor$Sepal.Width)
```

```
##
## Anderson-Darling normality test
##
## data: versicolor$Sepal.Width
## A = 0.55976, p-value = 0.1406
ad.test(versicolor$Petal.Length)

##
## Anderson-Darling normality test
##
## data: versicolor$Petal.Length
## A = 0.55506, p-value = 0.1446
ad.test(versicolor$Petal.Width)

##
## Anderson-Darling normality test
##
## data: versicolor$Petal.Width
## A = 0.95685, p-value = 0.01443
```

- Using the Anderson-Darling Test, the p_value for versicolor Petal.Width is less than 0.05, so we reject the null hypothesis. There is less than a 5% chance that the deviation from normality was due to random sampling. Meanwhile the p values for the other variables (Sepal.Length, Sepal.Width and Petal.Length) are greater than 0.05 which means that the sample may come from a random distribution.