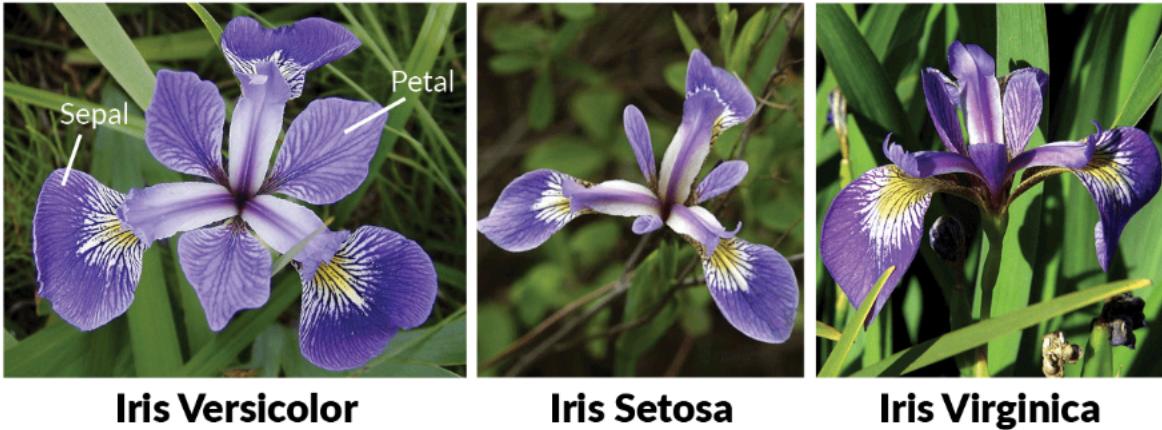


Iris flowers analytics

The Iris flower data set is a multivariate data set introduced by the British statistician and biologist Ronald Fisher in his 1936 paper *The use of multiple measurements in taxonomic problems*. It is sometimes called Anderson's Iris data set because Edgar Anderson collected the data to quantify the morphologic variation of Iris flowers of three related species. The data set consists of 50 samples from each of three species of Iris (*Iris Setosa*, *Iris virginica*, and *Iris versicolor*). Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters.

This dataset became a typical test case for many statistical classification techniques in machine learning such as support vector machines



In this paper we will study this dataset: get its statistics, the relationship between its variables, do some tests, create a predictive module and we will apply other useful functions.

Dataset columns:

- Id
- Sepal.Length: Length of the sepal (in cm).
- Sepal.Width: Width of the sepal (in cm).
- Petal.Length: Length of the petal (in cm).
- Petal.Width: Width of the petal (in cm).
- Species: Species name.

First of all, we need to get the dataset which is already included with in r packages.

```
# attach the iris dataset to the environment
data(iris)
dataset <- iris
#type of dataset
class(dataset)
attach(dataset)
```

Checking the class of this object using class () function and attach it to use the columns name directly.

```
> class(dataset)
[1] "data.frame"
> attach(dataset)
```

Data analysis

General information about our dataset (number of columns, columns' names, number of rows, attributes)

```
> #number of columns
> ncol(dataset)
[1] 5
> #columns' names
> colnames(dataset)
[1] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width" "Species"
> #number of rows
> nrow(dataset)
[1] 150
> #attributes
> attributes(dataset)
$names
[1] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width" "Species"

$class
[1] "data.frame"

$row.names
 [1]  1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31
[32] 32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62
[63] 63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93
[94] 94  95  96  97  98  99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124
[125] 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150
```

Now let's check the Spread of Data:

Mean and median

```
> #mean  
> mean(Sepal.Length)  
[1] 5.843333  
> mean(Sepal.Width)  
[1] 3.057333  
> mean(Petal.Length)  
[1] 3.758  
> mean(Petal.Width)  
[1] 1.199333  
> #median  
> median(Sepal.Length)  
[1] 5.8  
> median(Sepal.Width)  
[1] 3  
> median(Petal.Length)  
[1] 4.35  
> median(Petal.Width)  
[1] 1.3
```

#variance and standard deviation

```
> #Variance  
> var(Sepal.Length)  
[1] 0.6856935  
> var(Sepal.Width)  
[1] 0.1899794  
> var(Petal.Length)  
[1] 3.116278  
> var(Petal.Width)  
[1] 0.5810063  
> #standard deviation  
> sd(Sepal.Length)  
[1] 0.8280661  
> sd(Sepal.Width)  
[1] 0.4358663  
> sd(Petal.Length)  
[1] 1.765298  
> sd(Petal.Width)  
[1] 0.7622377
```

#Checking the summary of our dataset:

```
> summary(dataset)  
 Sepal.Length     Sepal.Width      Petal.Length      Petal.Width       Species  
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50  
 1st Qu.:5.100  1st Qu.:2.800  1st Qu.:1.600  1st Qu.:0.300  versicolor:50  
 Median :5.800  Median :3.000  Median :4.350  Median :1.300  virginica :50  
 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
```

asymmetry and flattening

```
> #asymmetry
> skewness(Sepal.Length)
[1] 0.3117531
> skewness(Sepal.Width)
[1] 0.3157671
> skewness(Petal.Length)
[1] -0.2721277
> skewness(Petal.Width)
[1] -0.1019342
> #kurtosis
> kurtosis(Sepal.Length)
[1] 2.426432
> kurtosis(Sepal.Width)
[1] 3.180976
> kurtosis(Petal.Length)
[1] 1.604464
> kurtosis(Petal.Width)
[1] 1.663933
> |
```

So, we can note that data distribution is right-skewed for Sepal length and width and the data distribution is left-skewed for Petal length and width.

For Kurtosis we can note that only Sepal.width larger than 3 so it's leptokurtic distribution. For the other variables we can note that the flattening value is less than 3 then it's platykurtic distribution.

#frequent subinterval table:

```
> #frequent subinterval table
> table(cut(Sepal.Length,3))

(4.3,5.5] (5.5,6.7] (6.7,7.9]
      59       71       20
> table(cut(Sepal.Width,3))

(2,2.8] (2.8,3.6] (3.6,4.4]
      47       88       15
> table(cut(Petal.Length,3))

(0.994,2.97] (2.97,4.93] (4.93,6.91]
      50       54       46
> table(cut(Petal.Width,3))

(0.0976,0.9] (0.9,1.7] (1.7,2.5]
      50       54       46
```

```
#the proportion of the variables by Species
```

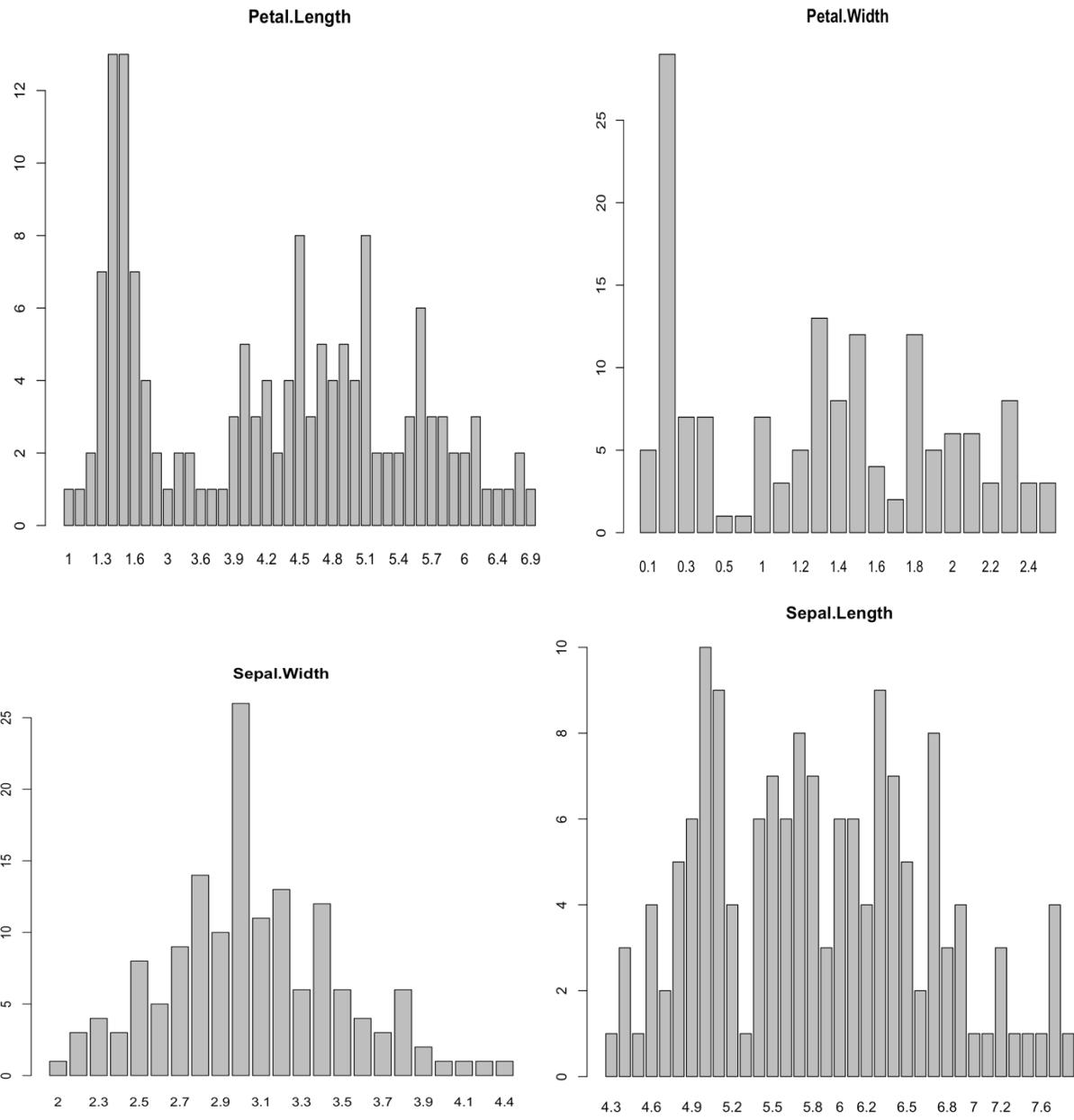
```
> #the proportion of the variables by Species
> prop.table(xtabs(Sepal.Length~Species))
Species
  setosa versicolor virginica
  0.2855676  0.3386195  0.3758129
> prop.table(xtabs(Sepal.Width~Species))
Species
  setosa versicolor virginica
  0.3737462  0.3020061  0.3242477
> prop.table(xtabs(Petal.Length~Species))
Species
  setosa versicolor virginica
  0.1296789  0.3778606  0.4924605
> prop.table(xtabs(Petal.Width~Species))
Species
  setosa versicolor virginica
  0.06837132 0.36853808 0.56309061
```

Graphs

```
#barplot
```

using table function to collect the data in groups:

```
> barplot(table(Sepal.Length),main="Sepal.Length")
> barplot(table(Sepal.Length),main="Sepal.Length")
> barplot(table(Sepal.Width),main="Sepal.Width")
> barplot(table(Petal.Length),main="Petal.Length")
> barplot(table(Petal.Width),main="Petal.Width")
```



#create histogram for each numeric variable and adding the density curve over the plot:

```
#create histogram of Sepal Length variable
hist(Sepal.Length,freq = F,xlab = "Sepal Length variable", main = "Histogram of Sepal length variable")
#adding the density curve over the plot
lines(density(Sepal.Length),col=2,lwd=3)

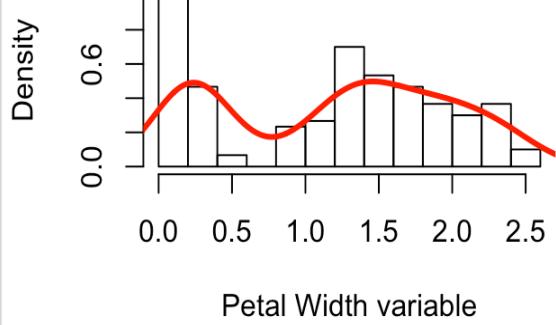
#create histogram of Sepal Width variable
hist(Sepal.Width,freq = F,xlab = "Sepal Width variable", main = "Histogram of Sepal Width variable")
#adding the density curve over the plot
lines(density(Sepal.Width),col=2,lwd=3)

#create histogram of Petal Length variable
hist(Petal.Length,freq = F,xlab = "Petal Length variable", main = "Histogram of Petal length variable")
#adding the density curve over the plot
lines(density(Petal.Length),col=2,lwd=3)

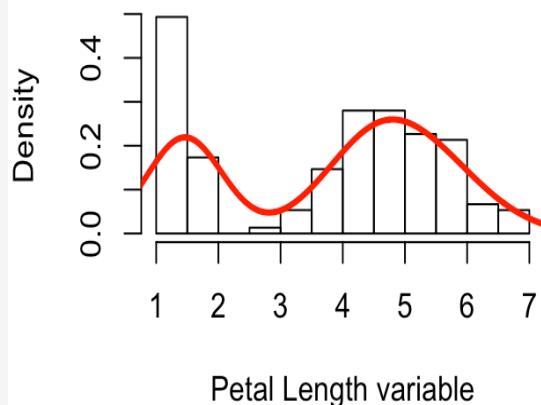
#create histogram of Petal Width variable
hist(Petal.Width,freq = F,xlab = "Petal Width variable", main = "Histogram of Petal Width variable")
#adding the density curve over the plot
lines(density(Petal.Width),col=2,lwd=3)

# Notice the shape of the data, most attributes exhibit a normal distribution.
```

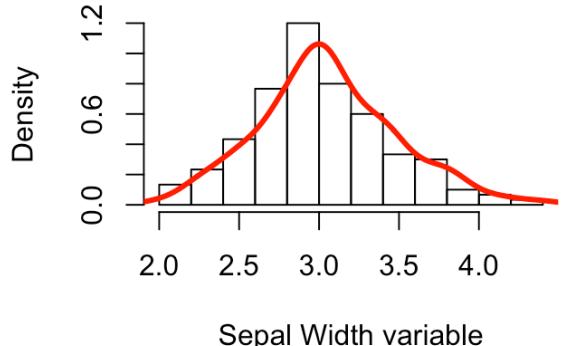
Histogram of Petal Width variable



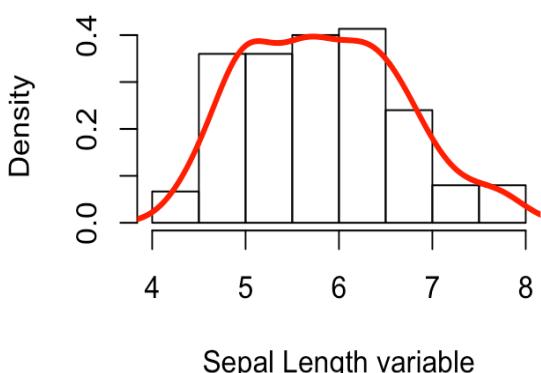
Histogram of Petal length variable



Histogram of Sepal Width variable



Histogram of Sepal length variable

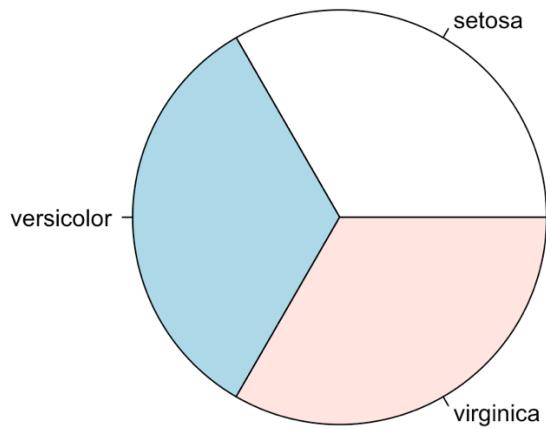


Notice the shape of the data, most attributes exhibit a normal distribution.

```
# pie graph for Species:
```

```
>  
> pie(table(Species), main = "Pie Chart of the Iris data set Species", radius = 1)  
>
```

Pie Chart of the Iris data set Species

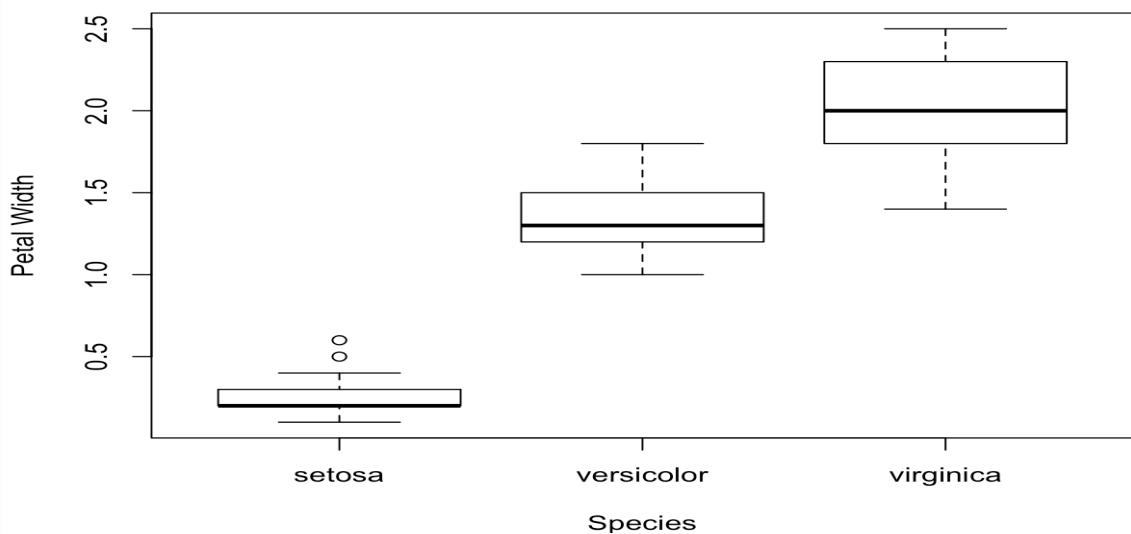


We can note that we have the same numbers of observation for each Species.

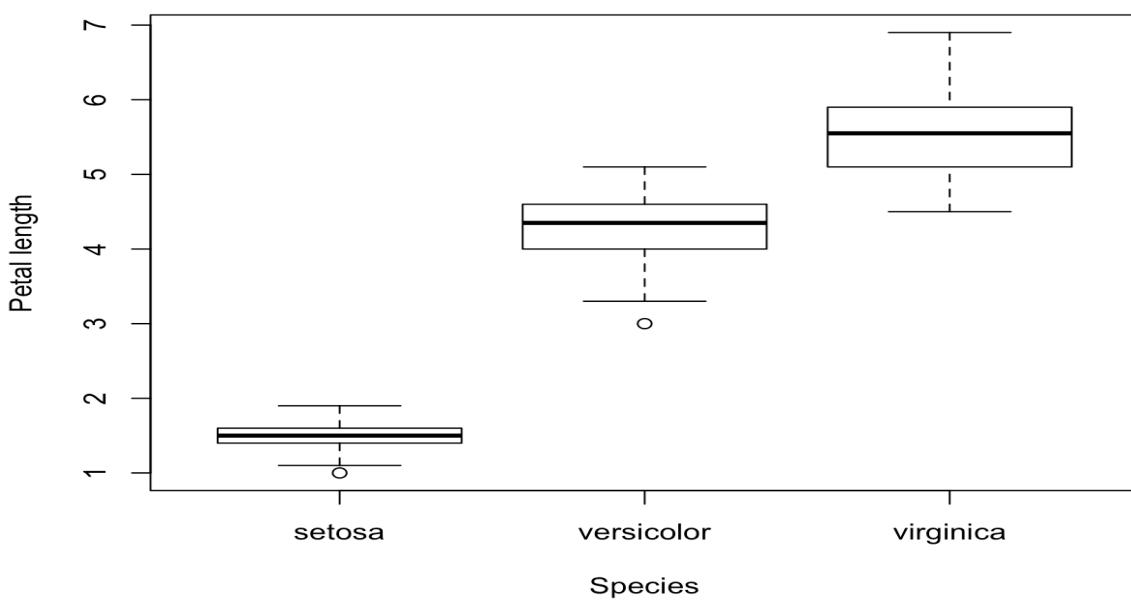
```
#Boxplot
```

```
> pie(table(Species), main = "Pie Chart of the Iris data set Species", radius = 1)  
> #Boxplot  
> #compare each variable with the Species variable  
> boxplot(Sepal.Length~Species, ylab = "Sepal Length", main = "distribution of Sepal Length for Species variables")  
> boxplot(Sepal.Width~Species, ylab = "Sepal Width", main = "distribution of Sepal Width for Species variables")  
> boxplot(Petal.Length~Species, ylab = "Petal length", main = "distribution of Petal length for Species variables")  
> boxplot(Petal.Width~Species, ylab = "Petal Width", main = "distribution of Petal Width for Species variables")  
~ |
```

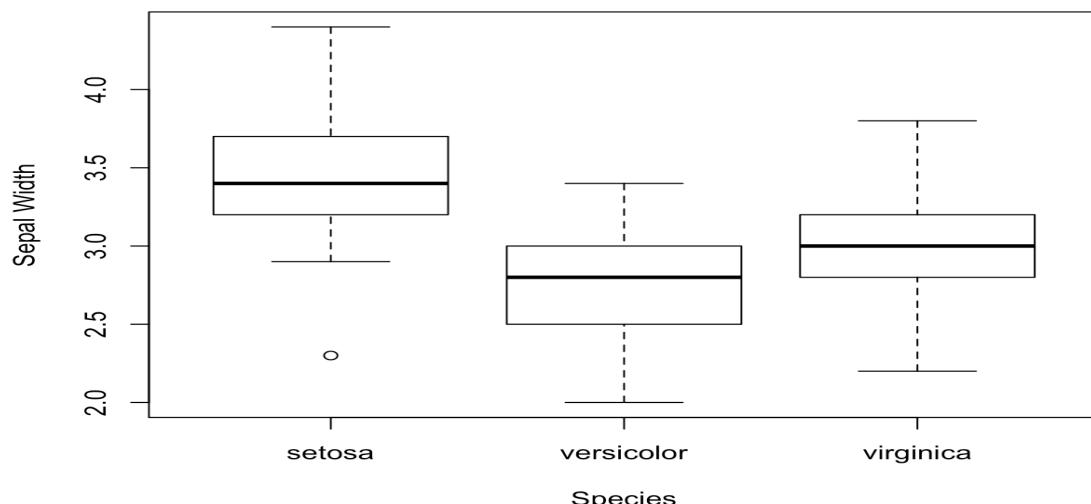
distribution of Petal Width for Species variables



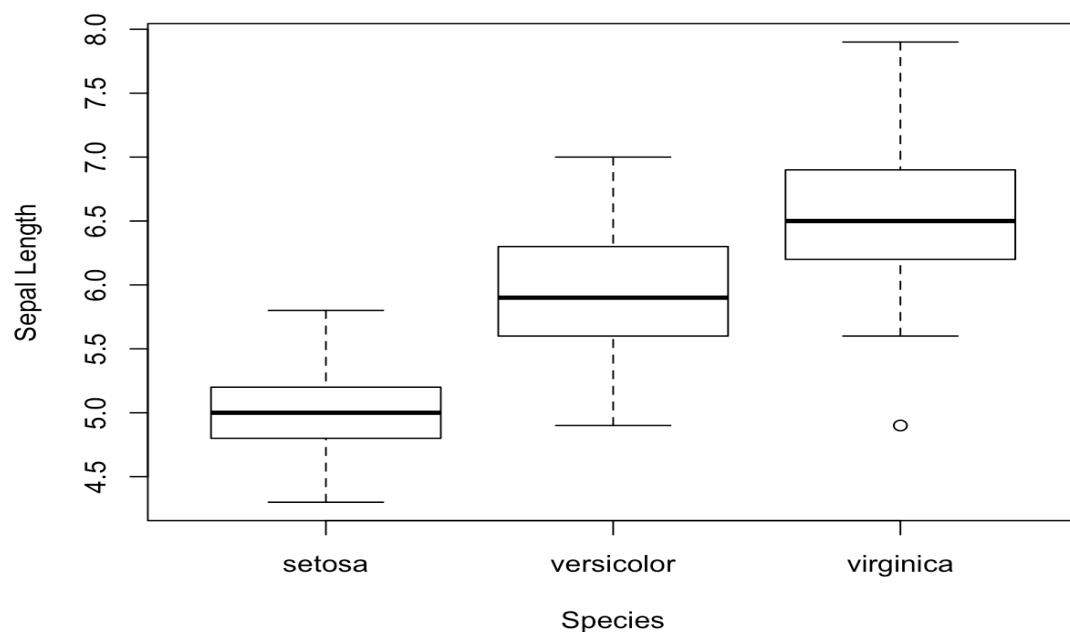
distribution of Petal length for Species variables



distribution of Sepal Width for Species variables

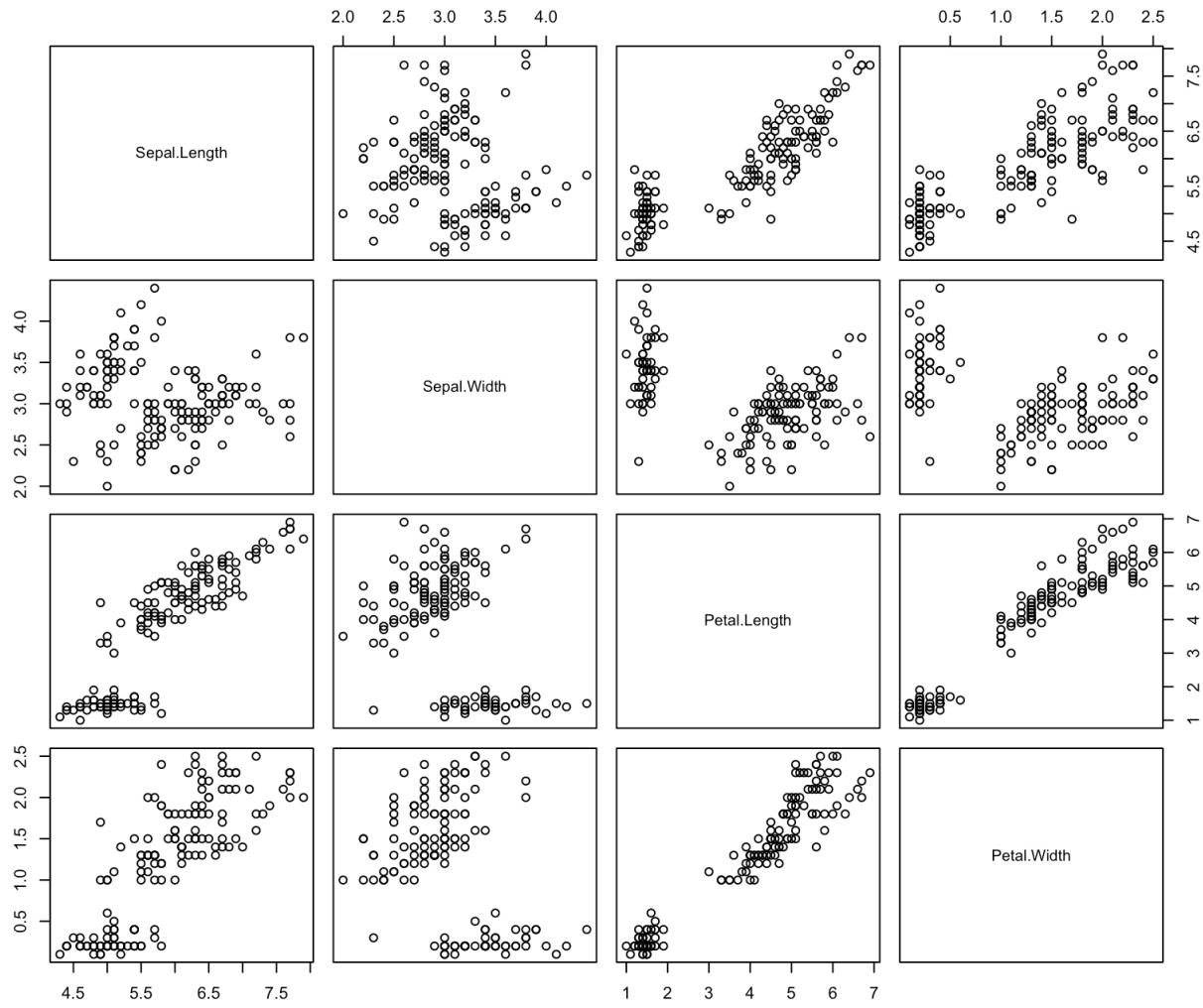


distribution of Sepal Length for Species variables



#plot

```
>  
> #get the plot for numeric columns:  
> pairs(iris[,1:4])  
>
```



we have some linear relationships like petal.length~ Petal.width.
and some none linear relationships like Sepal.length ~ Sepal.width.

covariance and correlation between variables

```
> # Check the covariance and correlation between variables
> cov(dataset[,1:4])
   Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    0.6856935 -0.0424340   1.2743154  0.5162707
Sepal.Width     -0.0424340  0.1899794  -0.3296564 -0.1216394
Petal.Length    1.2743154 -0.3296564   3.1162779  1.2956094
Petal.Width     0.5162707 -0.1216394   1.2956094  0.5810063
> cor(dataset[,1:4])
   Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    1.0000000 -0.1175698   0.8717538  0.8179411
Sepal.Width     -0.1175698  1.0000000  -0.4284401 -0.3661259
Petal.Length    0.8717538 -0.4284401   1.0000000  0.9628654
Petal.Width     0.8179411 -0.3661259   0.9628654  1.0000000
> |
```

We can note that it has different linear correlation:

positive linear correlation when the result of the second table is positive like:

Sepal.length~Petal.length... etc.

negative linear correlation when the result of the second table is negative like:

Sepal.length~ Sepal.width... etc.

#T test

```
> # Subset setosa data
> setosa <- dataset[dataset$Species == "setosa", ]
> # Subset versicolor data
> versicolor <- dataset[dataset$Species == "versicolor", ]
> # Subset virginica data
> virginica <- dataset[dataset$Species == "virginica", ]
>
> t.test(Petal.Length, mu=0 , alternative = "less")

One Sample t-test

data: Petal.Length
t = 26.073, df = 149, p-value = 1
alternative hypothesis: true mean is less than 0
95 percent confidence interval:
 -Inf 3.996566
sample estimates:
mean of x
3.758
```

```

> #Do t test for spesific Species
> # t test for Petal Lenght for different Species
> t.test(x = versicolor$Petal.Length, y = virginica$Petal.Length)

Welch Two Sample t-test

data: versicolor$Petal.Length and virginica$Petal.Length
t = -12.604, df = 95.57, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-1.49549 -1.08851
sample estimates:
mean of x mean of y
4.260      5.552

> # we noticed that versicolor and virginica also have different significantly different petal lengths.
> t.test(x = setosa$Petal.Length, y = virginica$Petal.Length)

Welch Two Sample t-test

data: setosa$Petal.Length and virginica$Petal.Length
t = -49.986, df = 58.609, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-4.253749 -3.926251
sample estimates:
mean of x mean of y
1.462      5.552

> #we noticed that setosa and virginica have significantly different petal lengths.
>

```

In the result above:

t is the t-test statistic value,
df is the degrees of freedom,
p-value is the significance level of the t-test.
conf.int is the confidence interval of the mean at 95%.
sample estimates are he mean value of the sample.

Results of t test:

- in the first example: we can NOT reject the null hypothesis cause the P value is larger than significance level 0.05.
- In the second and the third tests we can reject the null hypothesis and accept the alternative hypothesis.
- Related to the second test, we noticed that versicolor and virginica also have different significantly different petal lengths
- Related to the third test, we noticed that setosa and virginica have significantly different petal lengths.

#ANOVA

The question we will address is: are there differences in each variable among the three species?

```
> summary(aov(Petal.Length~Species))
      Df Sum Sq Mean Sq F value Pr(>F)
Species      2  437.1   218.55    1180 <2e-16 ***
Residuals  147   27.2     0.19
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Sepal.Length~Species))
      Df Sum Sq Mean Sq F value Pr(>F)
Species      2   63.21   31.606    119.3 <2e-16 ***
Residuals  147   38.96    0.265
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Sepal.Width~Species))
      Df Sum Sq Mean Sq F value Pr(>F)
Species      2   11.35    5.672    49.16 <2e-16 ***
Residuals  147   16.96    0.115
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Petal.Width~Species))
      Df Sum Sq Mean Sq F value Pr(>F)
Species      2   80.41   40.21     960 <2e-16 ***
Residuals  147    6.16    0.04
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> |
```

We can note that the species do have significantly different among all variables because of the small value of p-value.

#Linear Regression Module

```
> set.seed(12)
> library(caTools)
> #split the data with ration of 70%
> split<- sample.split(dataset, SplitRatio=0.7)
> #create the subsets for studing and testing
> studied_set <- subset(dataset, split="TRUE")
> test <- subset(dataset, split="FALSE")
`|
```

Spilt our dataset by ration of 70% and create two subsets one for testing an another for predicting values.

We used functions from caTools to split our dataset.

Then we create the module of Petal Length and check the module summary:

```
> mod_Petal_Length <- lm(Petal.Length~. , data = dataset)
> #check the summary result of this module
> summary(mod_Petal_Length)

Call:
lm(formula = Petal.Length ~ ., data = dataset)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.78396 -0.15708  0.00193  0.14730  0.65418 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.11099   0.26987 -4.117 6.45e-05 ***
Sepal.Length  0.60801   0.05024 12.101 < 2e-16 ***
Sepal.Width   -0.18052   0.08036 -2.246  0.0262 *  
Petal.Width    0.60222   0.12144  4.959 1.97e-06 ***
Speciesversicolor 1.46337   0.17345  8.437 3.14e-14 ***
Speciesvirginica 1.97422   0.24480  8.065 2.60e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

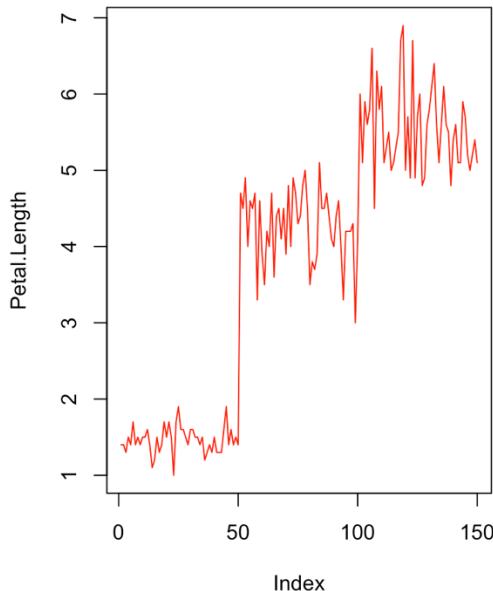
Residual standard error: 0.2627 on 144 degrees of freedom
Multiple R-squared:  0.9786,    Adjusted R-squared:  0.9778 
F-statistic: 1317 on 5 and 144 DF,  p-value: < 2.2e-16
```

Generate the predictions module using predict function and display them

```
> #prediction
> pred_Petal_Length <- predict(mod_Petal_Length, test)
> #prediction value
> pred_Petal_Length
   1     2     3     4     5     6     7     8     9     10    11    12    13    14
1.478453 1.447113 1.289407 1.246659 1.399600 1.709089 1.252724 1.435704 1.161163 1.368839 1.624750 1.314103 1.326091 1.022088
  15    16    17    18    19    20    21    22    23    24    25    26    27    28
1.813795 1.801229 1.709089 1.538675 1.849321 1.484518 1.678907 1.562792 1.156397 1.695224 1.314103 1.507914 1.556149 1.539253
  29    30    31    32    33    34    35    36    37    38    39    40    41    42
1.557306 1.289407 1.368260 1.799351 1.370717 1.595288 1.429061 1.471809 1.721655 1.278577 1.143110 1.496505 1.477874 1.390500
  43    44    45    46    47    48    49    50    51    52    53    54    55    56
1.107006 1.658541 1.544740 1.446535 1.424296 1.228607 1.563949 1.453757 4.873857 4.569276 4.891331 4.064098 4.702286 4.095437
  57    58    59    60    61    62    63    64    65    66    67    68    69    70
4.550645 3.500575 4.624590 3.869709 3.633586 4.301378 4.205487 4.380809 4.016584 4.709508 4.118976 3.993624 4.628199 3.968349
  71    72    73    74    75    76    77    78    79    80    81    82    83    84
4.445940 4.338639 4.634842 4.278417 4.502989 4.666760 4.824466 4.908227 4.380231 3.950875 3.925601 3.865379 4.114068 4.476558
  85    86    87    88    89    90    91    92    93    94    95    96    97    98
3.997375 4.350191 4.769730 4.550502 3.998532 4.027993 3.949719 4.362757 4.132120 3.579428 4.052689 3.999110 4.077385 4.381388
  99    100   101   102   103   104   105   106   107   108   109   110   111   112
3.664346 4.095437 5.603497 5.046475 5.903170 5.254151 5.598588 6.207173 4.414930 5.862157 5.569563 6.096545 5.442039 5.411278
  113   114   115   116   117   118   119   120   121   122   123   124   125   126
5.720768 5.082001 5.329533 5.561905 5.357700 6.183777 6.460627 5.017449 5.865908 4.967044 6.243856 5.290256 5.605810 5.747199
  127   128   129   130   131   132   133   134   135   136   137   138   139   140
5.211403 5.114497 5.513670 5.662860 6.001232 6.184933 5.573893 5.091537 4.945818 6.388418 5.525222 5.278847 5.053697 5.763516
  141   142   143   144   145   146   147   148   149   150
5.822582 5.883961 5.046475 5.805108 5.846699 5.780412 5.386583 5.478144 5.404199 4.992896
|
```

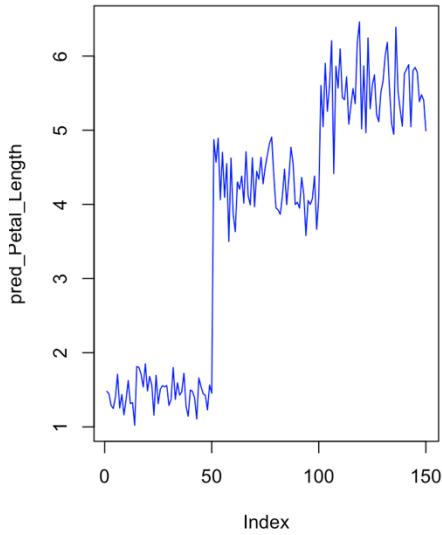
Plot the Petal.Length values

```
> plot(Petal.Length,type='l',lty=1.8, col=2)
>
```



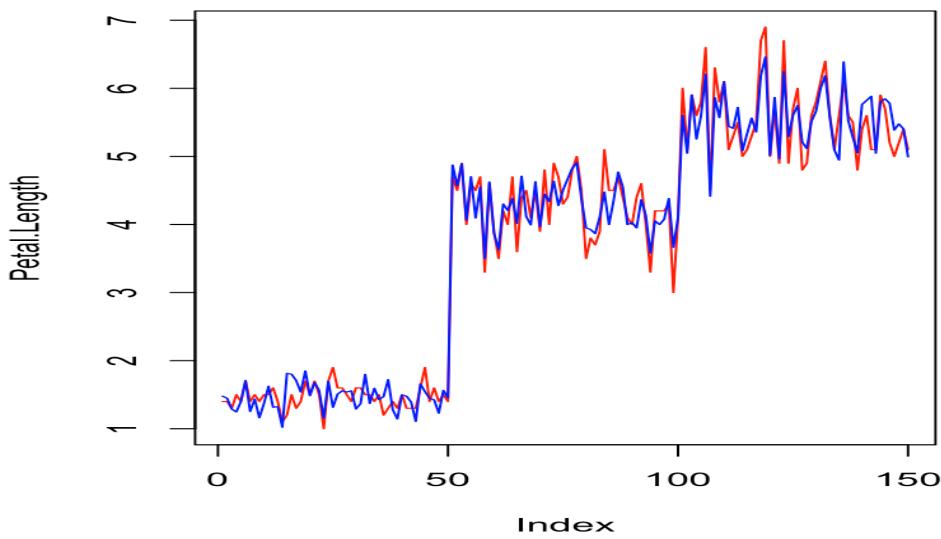
Plot the predicted values only

```
> plot(pred_Petal_Length,type='l',lty=1.8, col="blue")
>
```



Display both of them to check the errors and difference

```
> plot(Petal.Length,type='l',lty=1.8, col=2)
> lines(pred_Petal_Length,type='l', col="blue")
```



We did the same steps to create the second model to predicate the value of Petal.Width

```
#create the module to predicate the Petal.Width variable
mod_Petal_Width <- lm(Petal.Width~. , data = dataset)
#check the summary result of this module
summary(mod_Petal_Width)
#prediction
pred_Petal_Width <- predict(mod_Petal_Width, test)
#prediction value
pred_Petal_Width
#check the Petal.Width values only
plot(Petal.Width,type='l',lty=1.8, col=2)

#check the predicted values only
plot(pred_Petal_Width,type='l',lty=1.8, col="blue")

#comparing predicted VS actual values of Petal.Width
plot(Petal.Width,type='l',lty=1.8, col=2)
lines(pred_Petal_Width,type='l', col="blue")
```

Here are the results:

```
> mod_Petal_Width <- lm(Petal.Width~. , data = dataset)
> #check the summary result of this module
> summary(mod_Petal_Width)

Call:
lm(formula = Petal.Width ~ ., data = dataset)

Residuals:
    Min      1Q  Median      3Q      Max 
-0.59239 -0.08288 -0.01349  0.08773  0.45239 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.47314   0.17659  -2.679  0.00824 **  
Sepal.Length -0.09293   0.04458  -2.084  0.03889 *   
Sepal.Width   0.24220   0.04776  5.072 1.20e-06 ***  
Petal.Length  0.24220   0.04884  4.959 1.97e-06 ***  
Speciesversicolor 0.64811   0.12314  5.263 5.04e-07 ***  
Speciesvirginica 1.04637   0.16548  6.323 3.03e-09 ***  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1666 on 144 degrees of freedom
Multiple R-squared:  0.9538, Adjusted R-squared:  0.9522 
F-statistic: 594.9 on 5 and 144 DF,  p-value: < 2.2e-16

> |
```



```
> pred_Petal_Width <- predict(mod_Petal_Width, test)
> pred_Petal_Width
     1      2      3      4      5      6      7      8      9      10 
0.23968609 0.13717259 0.17997912 0.21349301 0.27319950 0.38134705 0.26193287 0.24897970 0.15941936 0.18561292 
     11     12     13     14     15     16     17     18     19     20 
0.28446638 0.29178671 0.14646595 0.12027191 0.24729220 0.42612662 0.28446590 0.23968609 0.32924691 0.33656652 
     21     22     23     24     25     26     27     28     29     30 
0.26024682 0.31234647 0.21349181 0.26390686 0.36444758 0.17631980 0.27319999 0.25461302 0.20617268 0.25263998 
     31     32     33     34     35     36     37     38     39     40 
0.21912657 0.21180624 0.39993330 0.37205297 0.18561292 0.12787874 0.17829235 0.28249287 0.15941912 0.23968633 
     41     42     43     44     45     46     47     48     49     50 
0.22475917 -0.01941457 0.20785921 0.29742003 0.43344767 0.14646595 0.36078681 0.21349277 0.29375975 0.20053936 
     51     52     53     54     55     56     57     58     59     60 
1.43783408 1.44515368 1.47134797 1.18971210 1.36320042 1.41332704 1.52710767 1.10015031 1.37812710 1.29025209 
     61     62     63     64     65     66     67     68     69     70 
1.04241734 1.37051955 1.11902523 1.44881421 1.22885786 1.36883326 1.47106050 1.28293248 1.22153994 1.20463854 
     71     72     73     74     75     76     77     78     79     80 
1.56428137 1.25505215 1.38178787 1.42459416 1.32405297 1.35390657 1.38376091 1.48993494 1.40966700 1.12268407 
     81     82     83     84     85     86     87     88     89     90 
1.16549157 1.14127128 1.23449191 1.50654863 1.48964723 1.53076723 1.44149412 1.21224634 1.37417935 1.23815219 
     91     92     93     94     95     96     97     98     99     100 
1.35925339 1.44881397 1.23449215 1.06663690 1.32573950 1.38910627 1.36488623 1.34263970 1.03312277 1.31644589 
    101    102    103    104    105    106    107    108    109    110 
2.24022913 1.92339308 2.06900179 2.04646779 2.10054169 2.19207699 1.81327153 2.12307617 1.96085472 2.25346928 
    111    112    113    114    115    116    117    118    119    120 
1.97943976 1.91607347 2.00000073 1.86002606 1.94761312 2.03717370 2.02788082 2.40076429 2.15856430 1.75948583 
    121    122    123    124    125    126    127    128    129    130 
2.08758804 1.91775928 2.15856382 1.82848568 2.13039481 2.13236881 1.83777880 1.91973255 2.01295438 2.03548814 
    131    132    133    134    135    136    137    138    139    140 
2.04112218 2.30951670 2.01295438 1.90114630 1.99239438 2.06168218 2.16756803 2.06139423 1.90480563 1.99070713 
    141    142    143    144    145    146    147    148    149    150 
2.05773443 1.91804626 1.92339308 2.14532198 2.13039481 1.93663323 1.80426588 1.95521996 2.12842081 1.98675985
```

