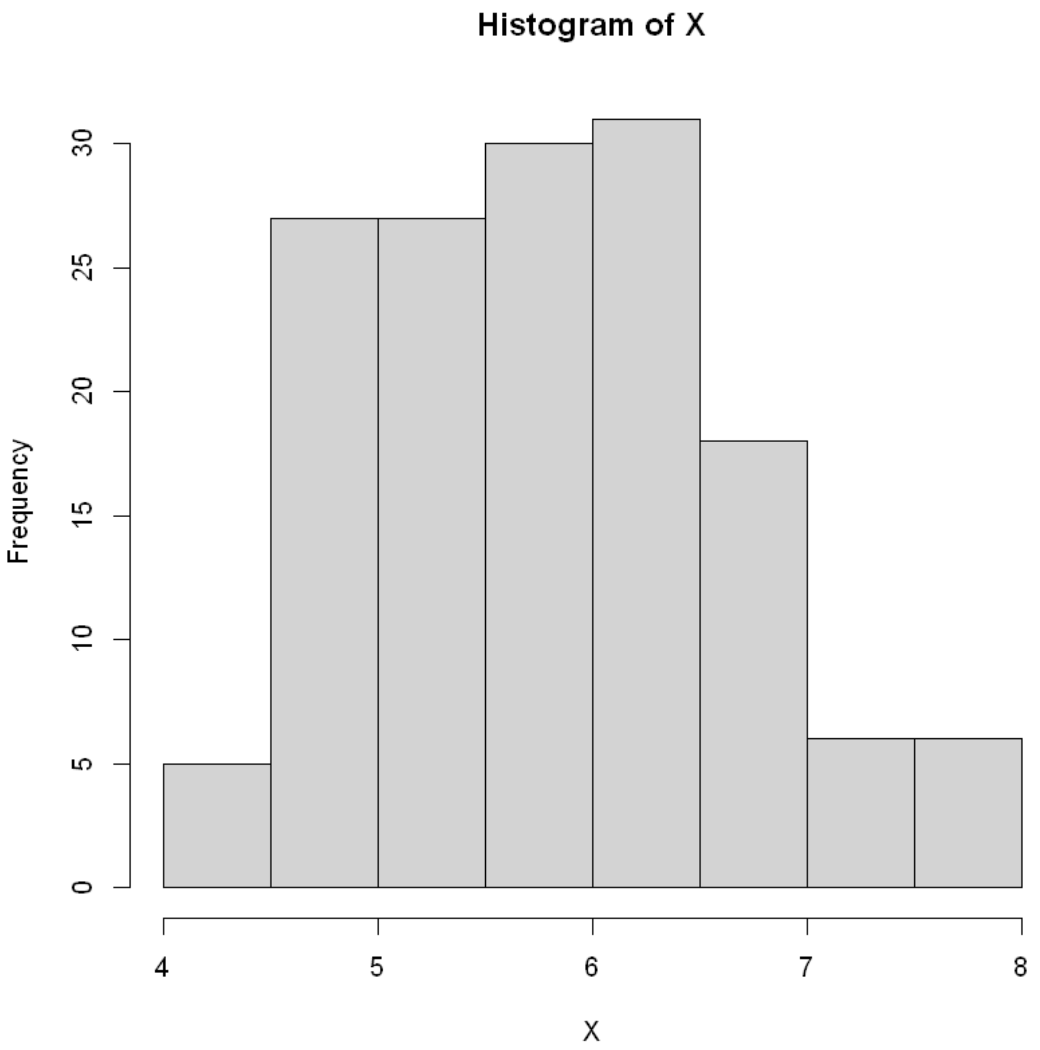
DescriptivStatistics

July 15, 2020

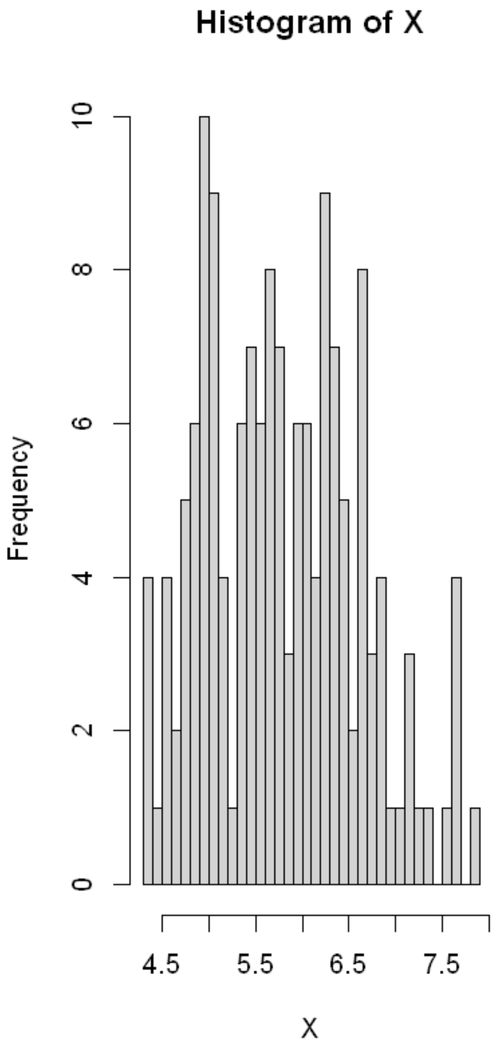
1 Reminder on descriptiv statistics

Comparison between the histogram and boxplot

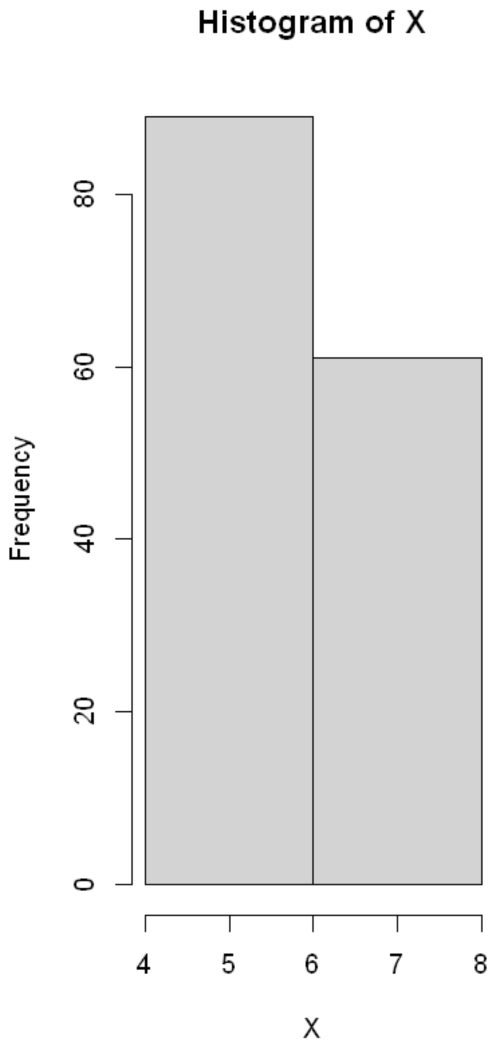
1.1 Histogram



It’s the default version of R. Possible to change the story by tuning the histogram.



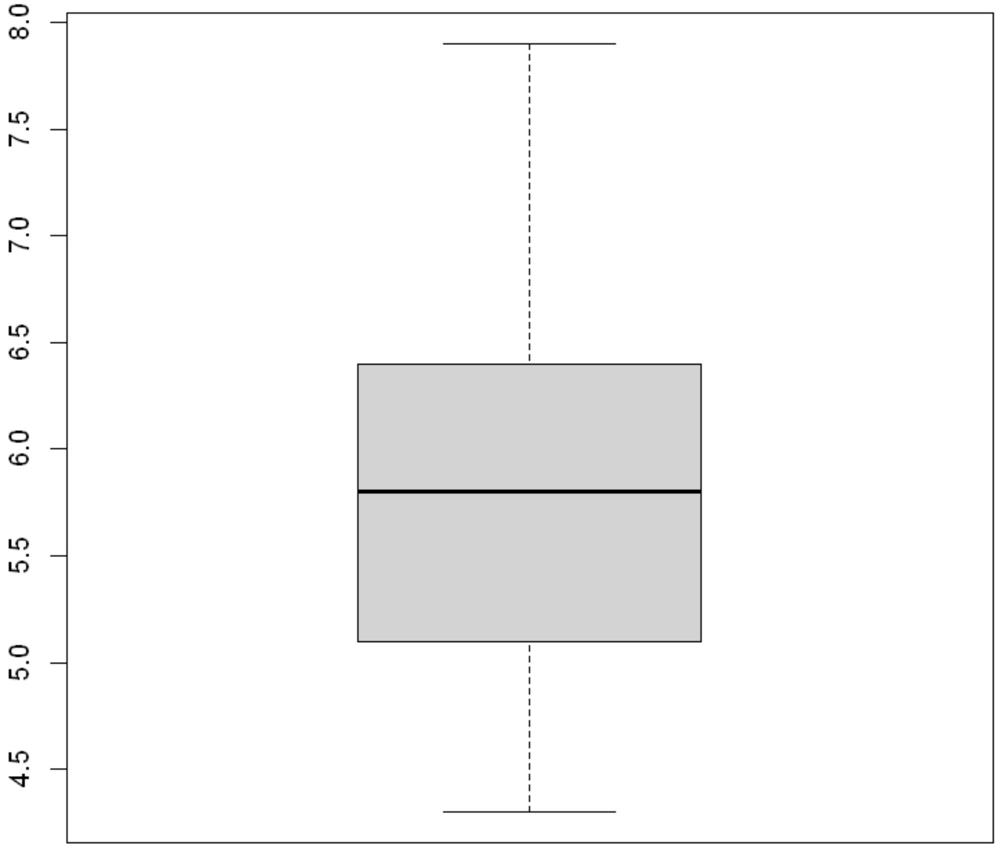
Histogram of X



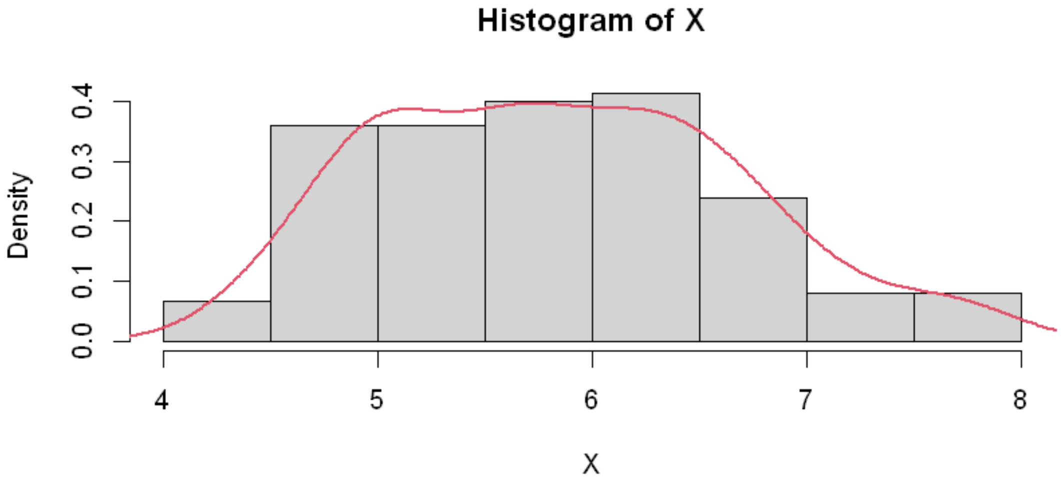
Histogram of X

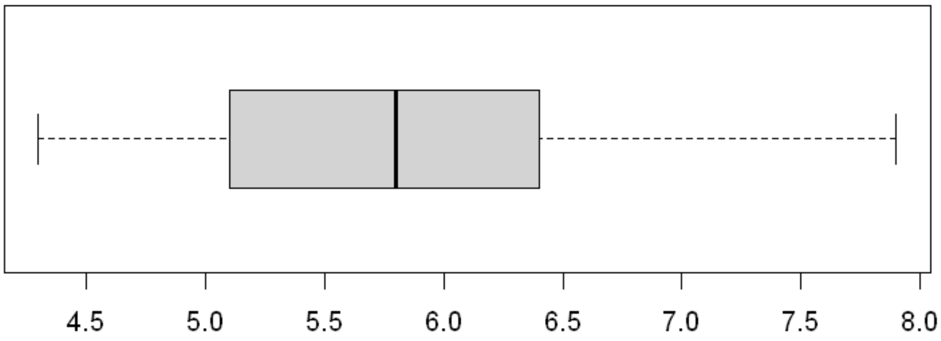
It’s possible to define specific function to calculate the best numbers of bins (by default “Strudge” function)

1.2 Boxplot



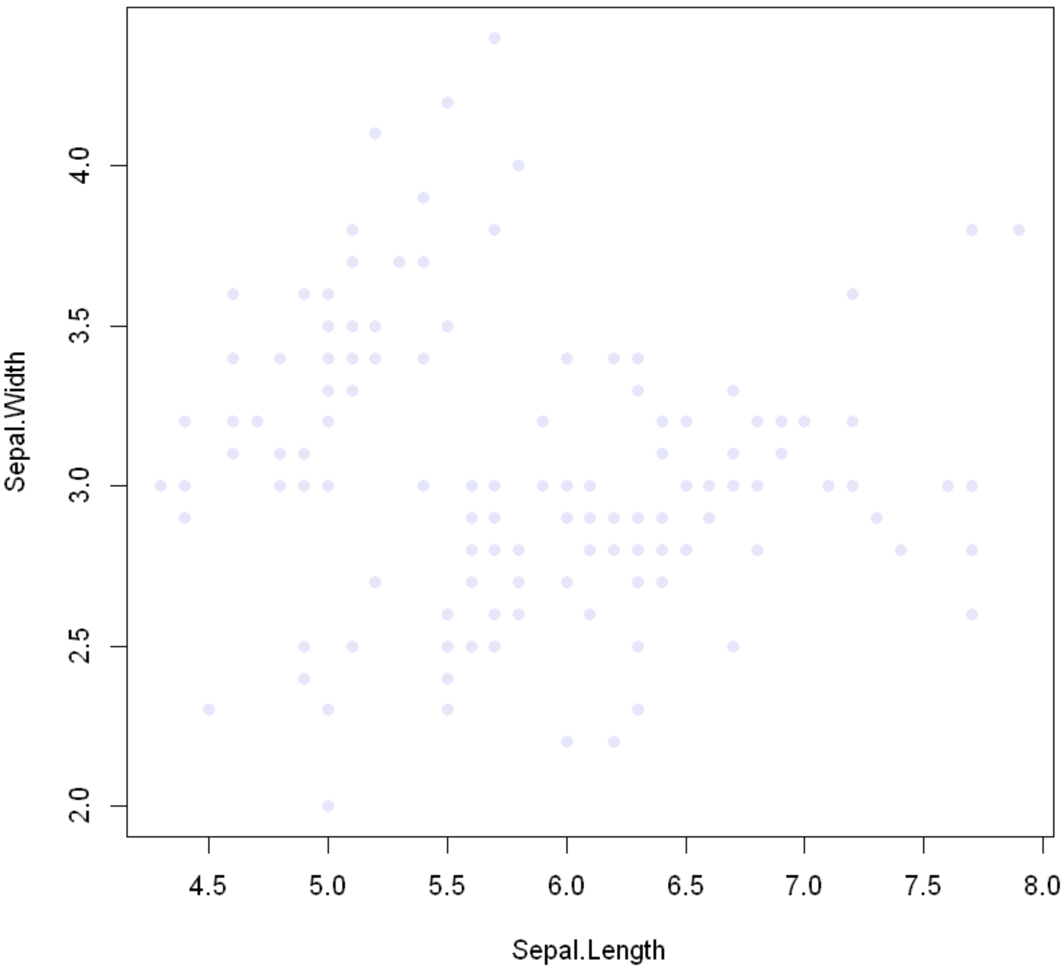
1.3 Histogram + Boxplot



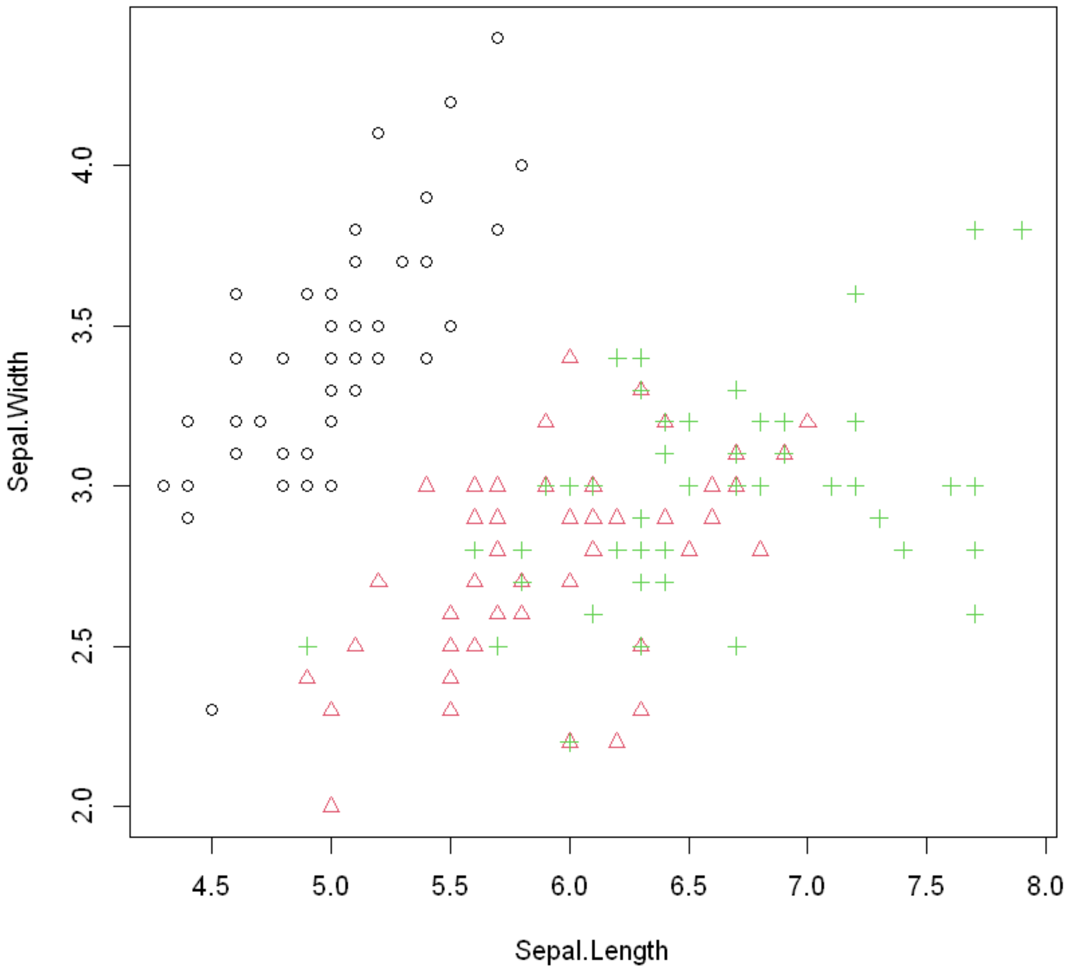


1.4 Scatter and Pair plot

1.4.1 Scatter plot



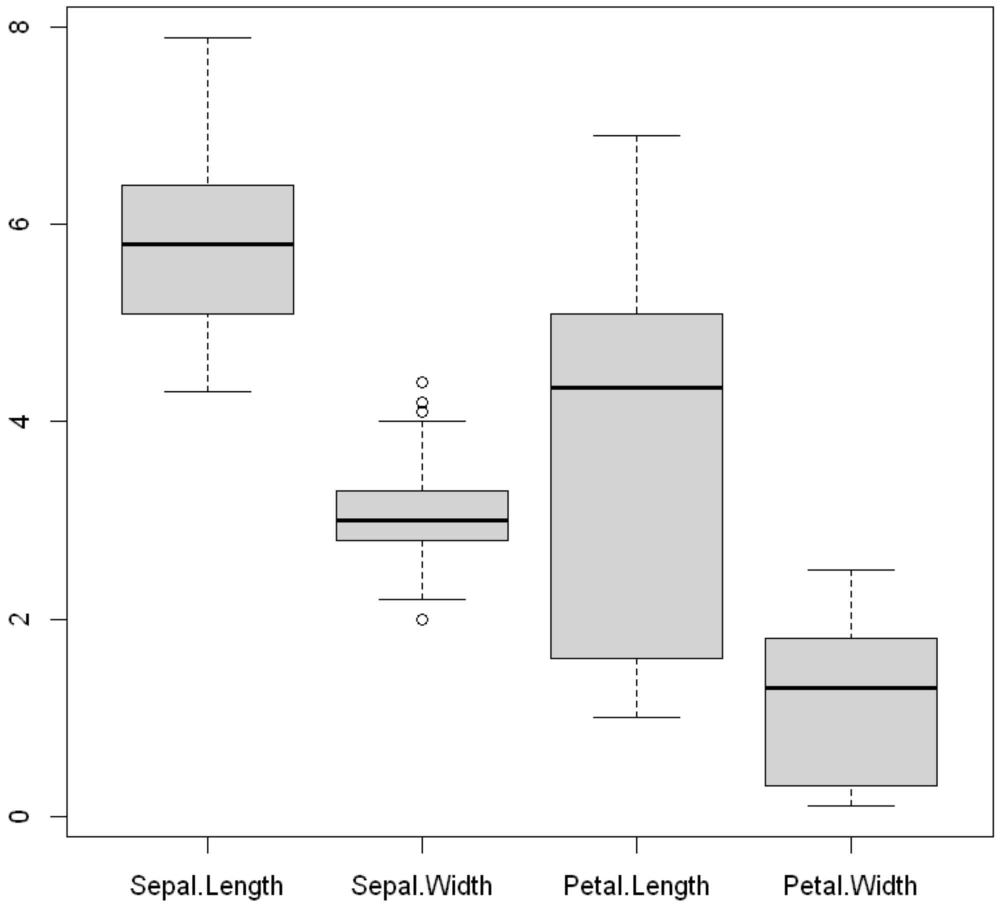
Add a categorical variable on colour and pattern



Here we can UNDERSTAND the datas

1.5 Multivariate data

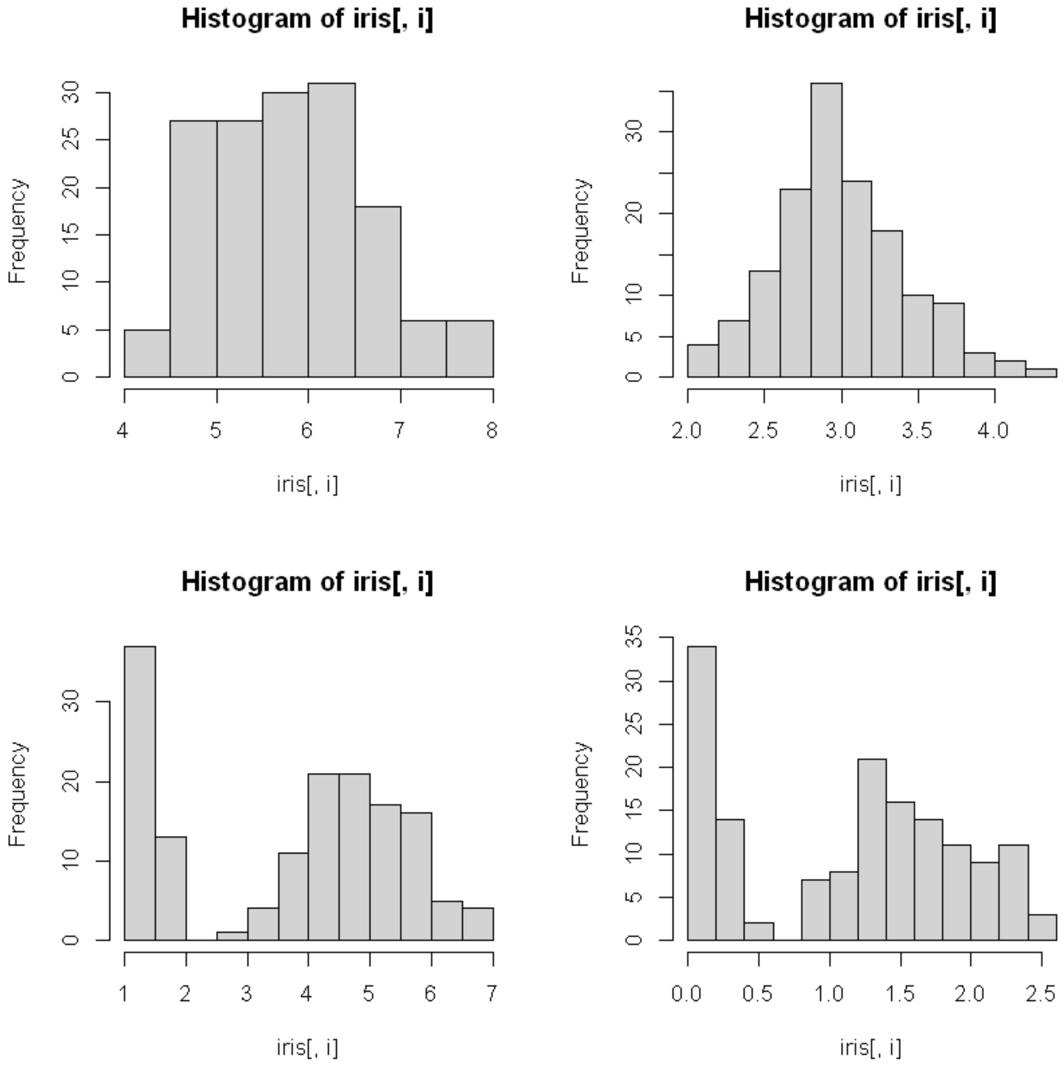
Iris : 4 continuous variable which are all measured in centimeters => boxplot possible



More variance on length then Width

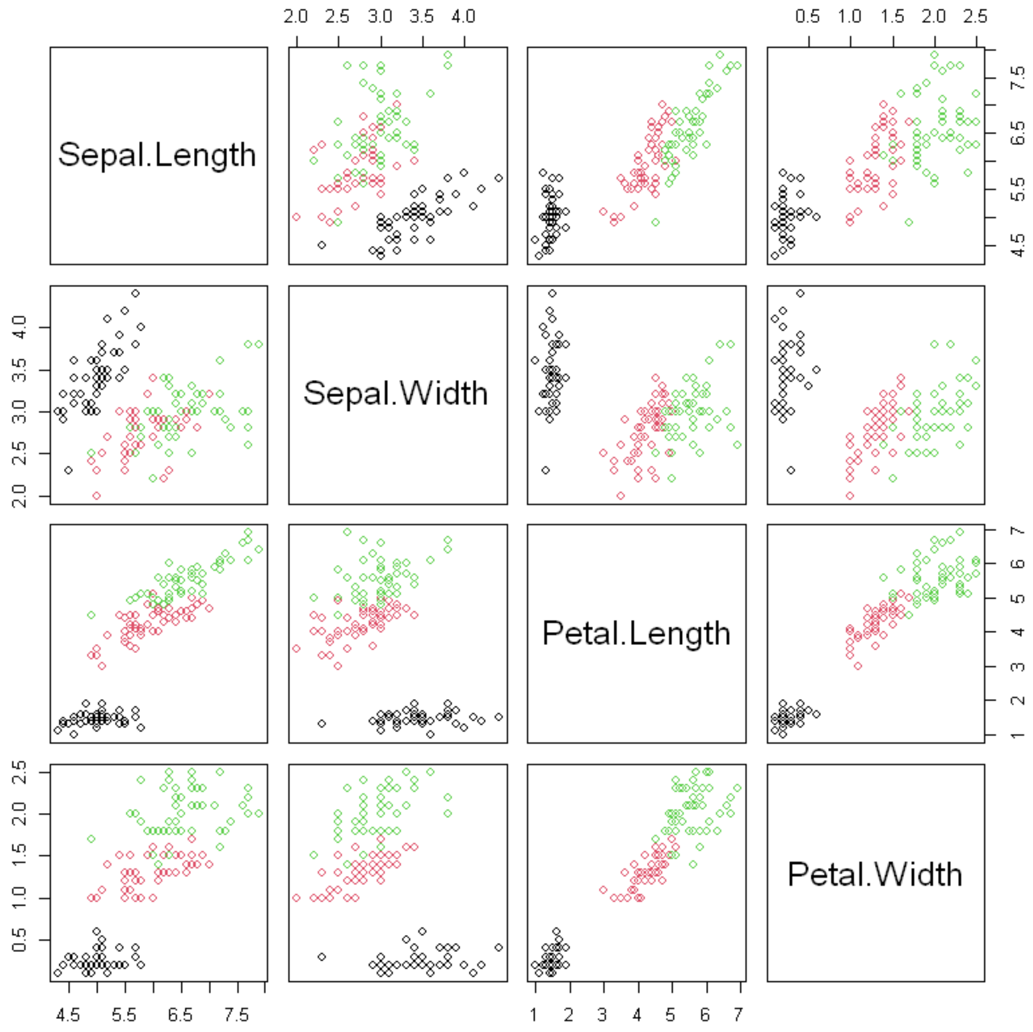
High variance on Petal.length ….

Try to do the same with histogram …



** BUT difficult to read, and not the same bins, ...

1.6 Pair plot



We can see groups, linear dependency

1.7 Multivariate numerical indicators

Mean vector colMeans

Sepal.Length 5.84333333333333 **Sepal.Width** 3.05733333333333 **Petal.Length** 3.758
Petal.Width 1.19933333333333

covariance matrix

A matrix: 4 × 4 of type dbl

	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

The covariance matrix can't be easely interpreted

correlation matrix

A matrix: 4 × 4 of type dbl

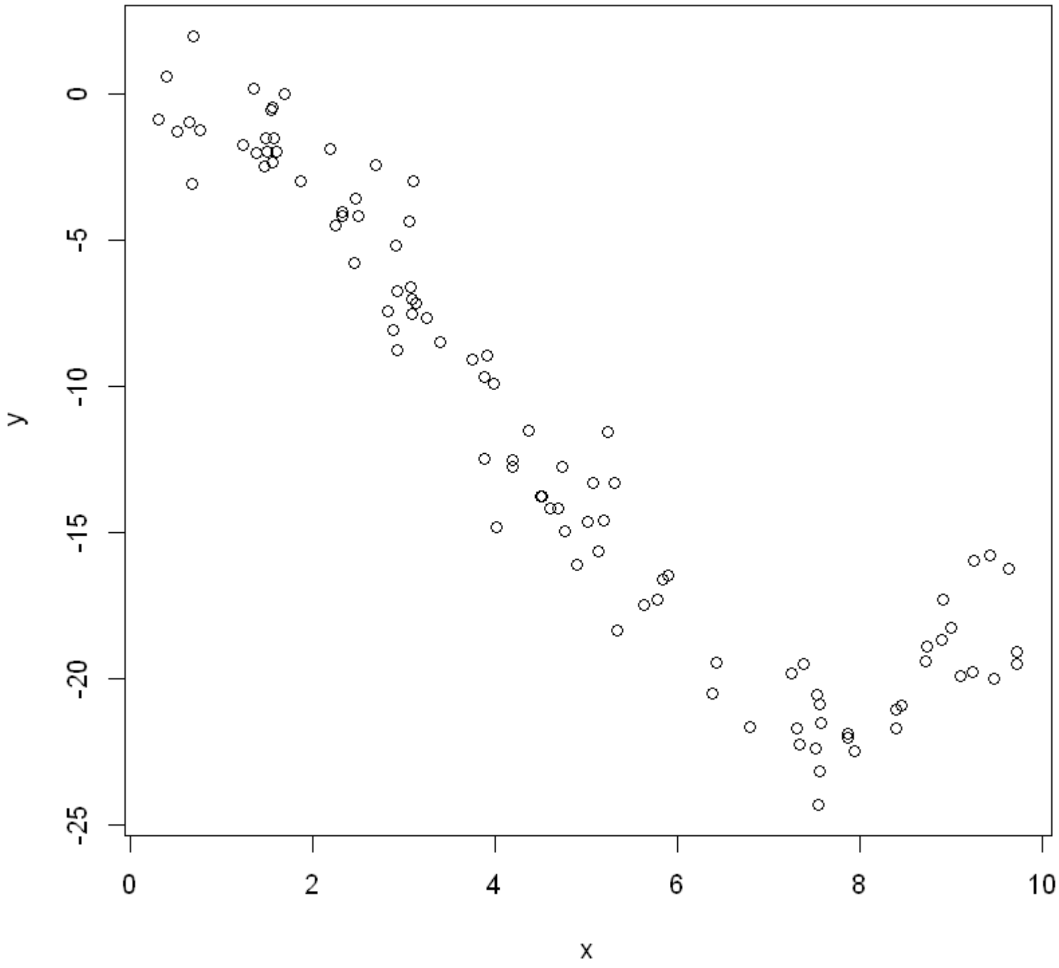
	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
Sepal.Length	1.0000000	-0.1175698	0.8717538	0.8179411
Sepal.Width	-0.1175698	1.0000000	-0.4284401	-0.3661259
Petal.Length	0.8717538	-0.4284401	1.0000000	0.9628654
Petal.Width	0.8179411	-0.3661259	0.9628654	1.0000000

Correlation matrix is easier to interprete. Here :

- petal.width and petal.length are highly correlated
- petal.length and sepal.length also

2 The Learning process - Importance to evaluate

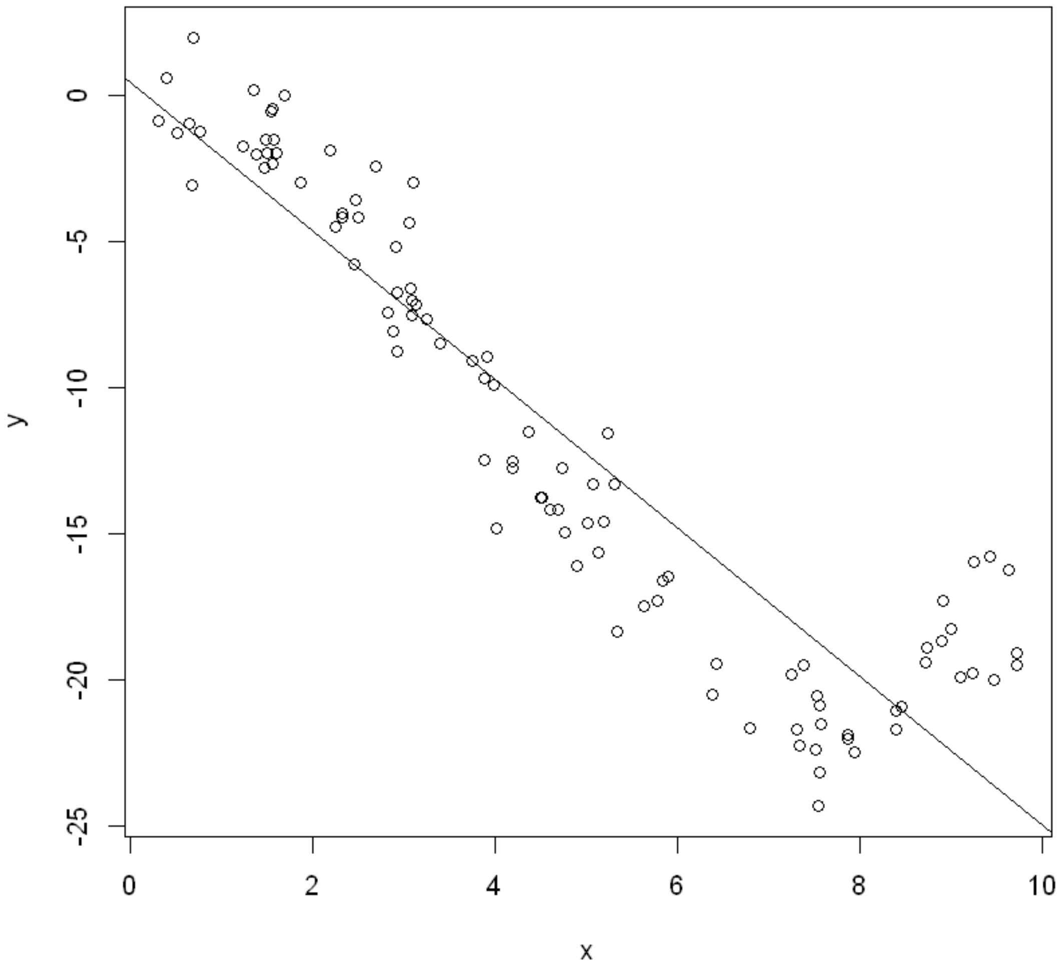
2.1 The Dataset to evaluate



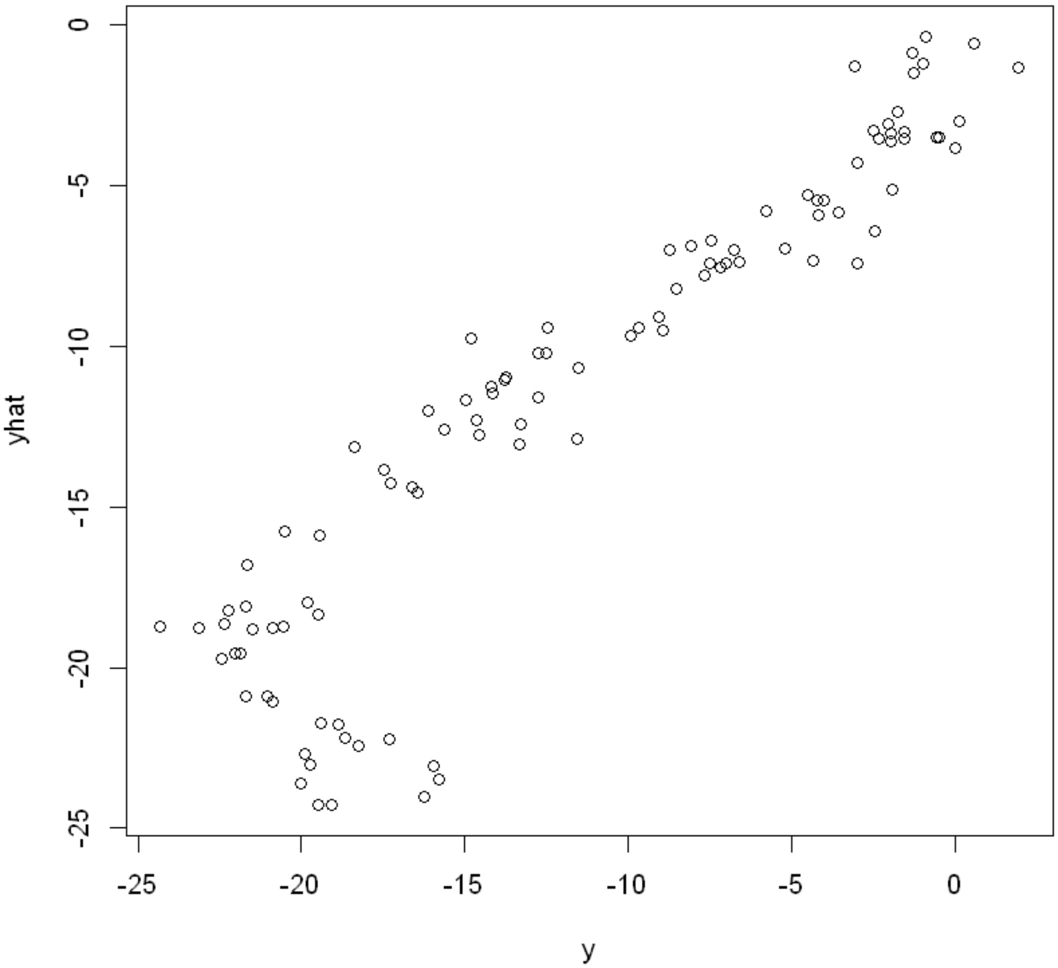
2.2 Learning step

```
Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
     0.4451      -2.5425
```


y
0
-5
-10
-15
-20
-25
x
0
2
4
6
8
10

2.3 Prediction step



Fit is not perfect : it should be a y=x line

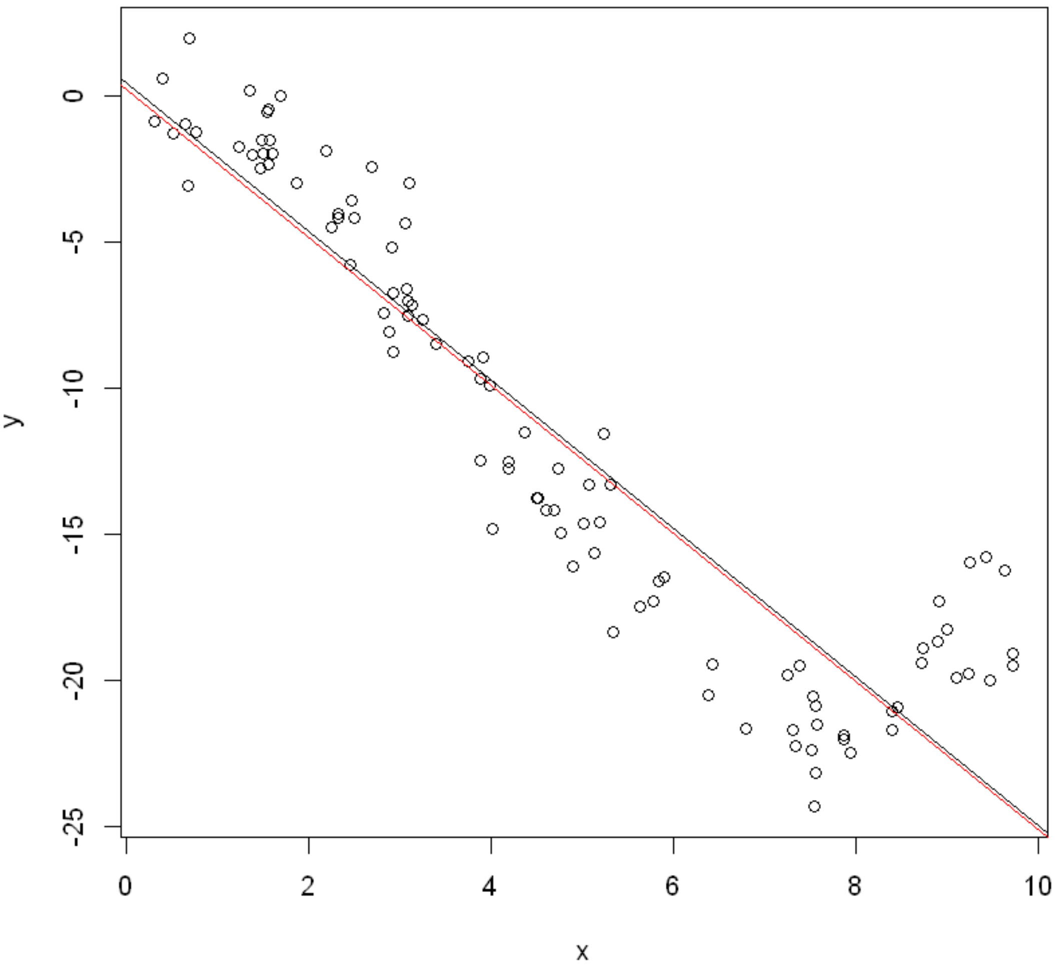
2.3.1 Calculation of Learning Error Step

8.49549897408271

**Not good BUT optimistic !

2.4 The learning process - with “minimal setup”

2.4.1 Learning step



2.4.2 Evaluation Step

2.4.3 Error level Step

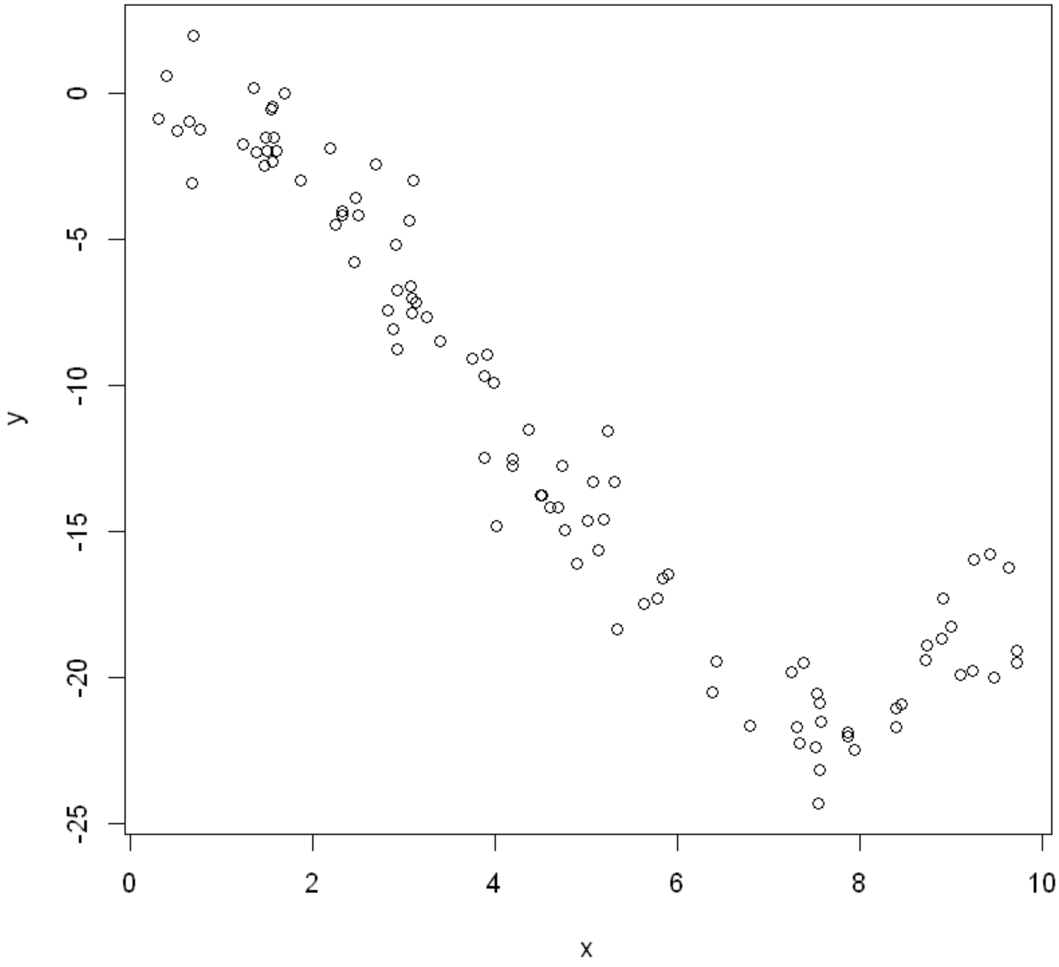
7.21923485650683

**The error increases from 11 to 12

2.5 Evaluation with a more complexe model - naive method

Let’s try a complexe model (x^6)

2.12485180557466



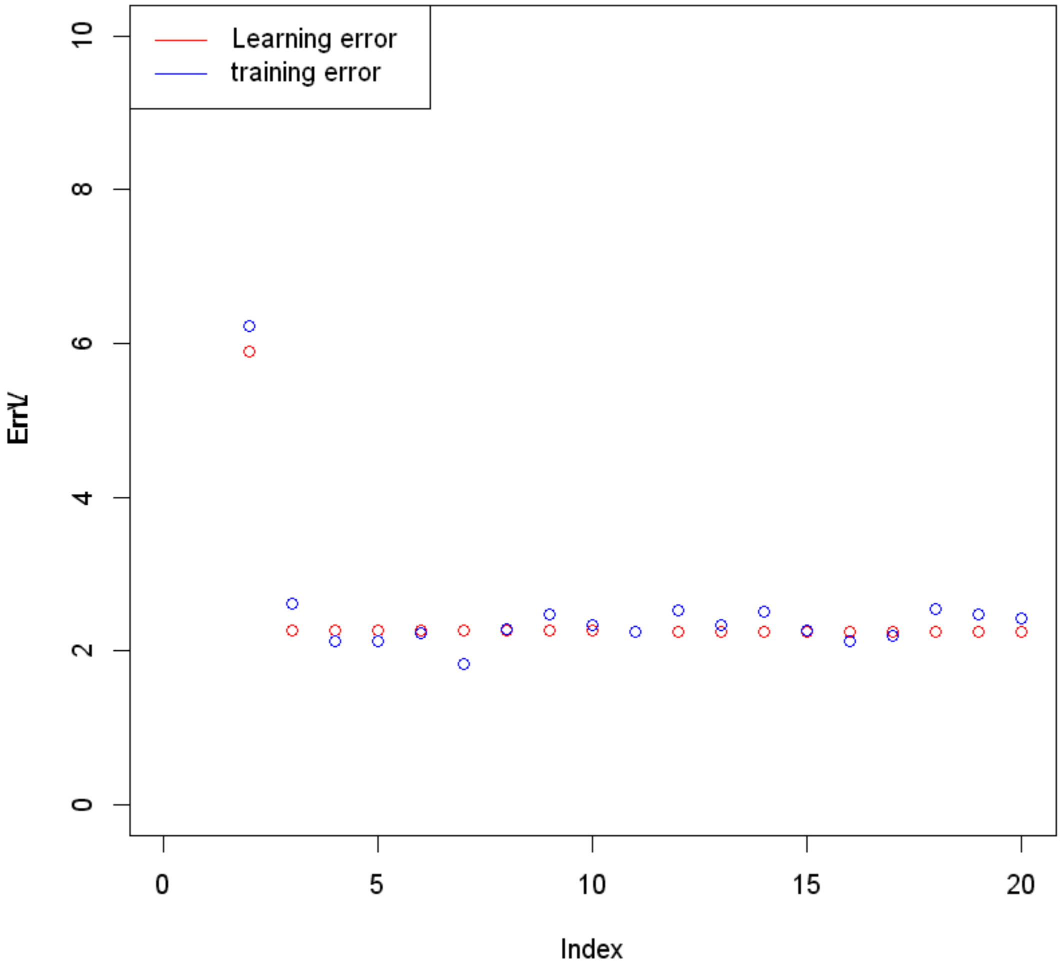
Now the error is really lower on the learning set ... but still to optimistic

2.5.1 with a the training set + evaluation

2.96609553450826

This error is more realistic

2.5.2 searching the best model : increasing the number of variable x^i of the model



20

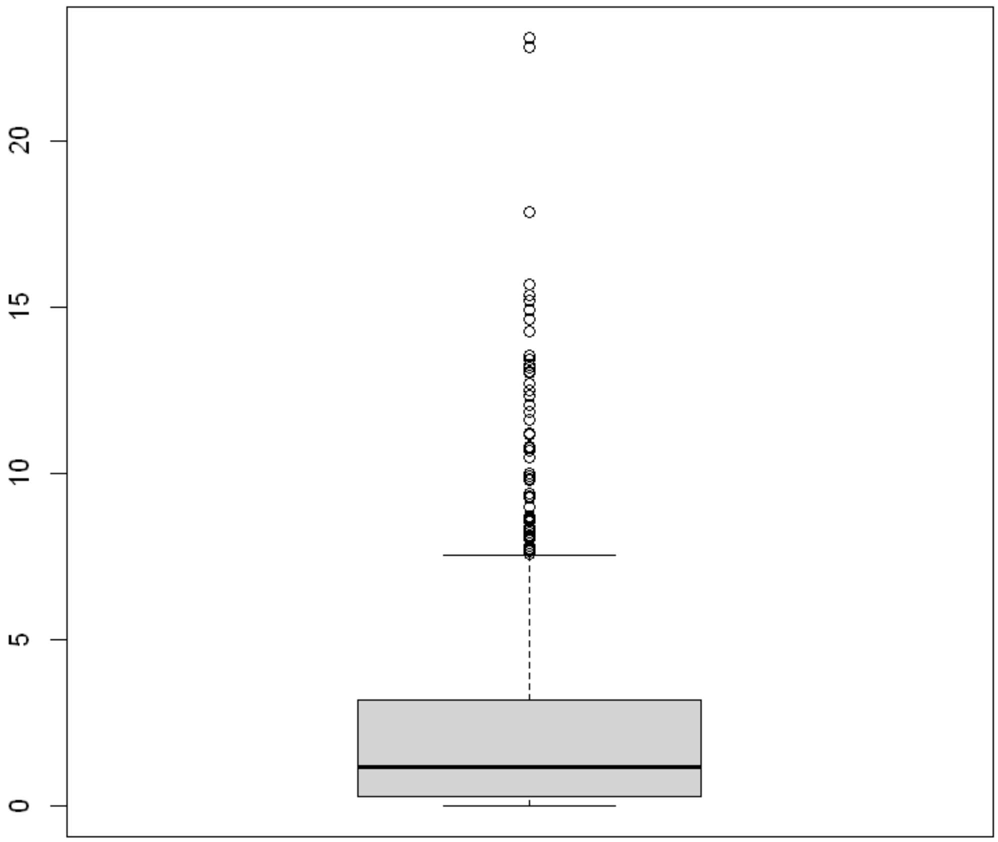
7

2.5.3 For Learning Error, the best model is with x^20 (in fact the model fit better and better when increasing the complexity

2.5.4 For training Error, the best model is with x^6, which is more realistic

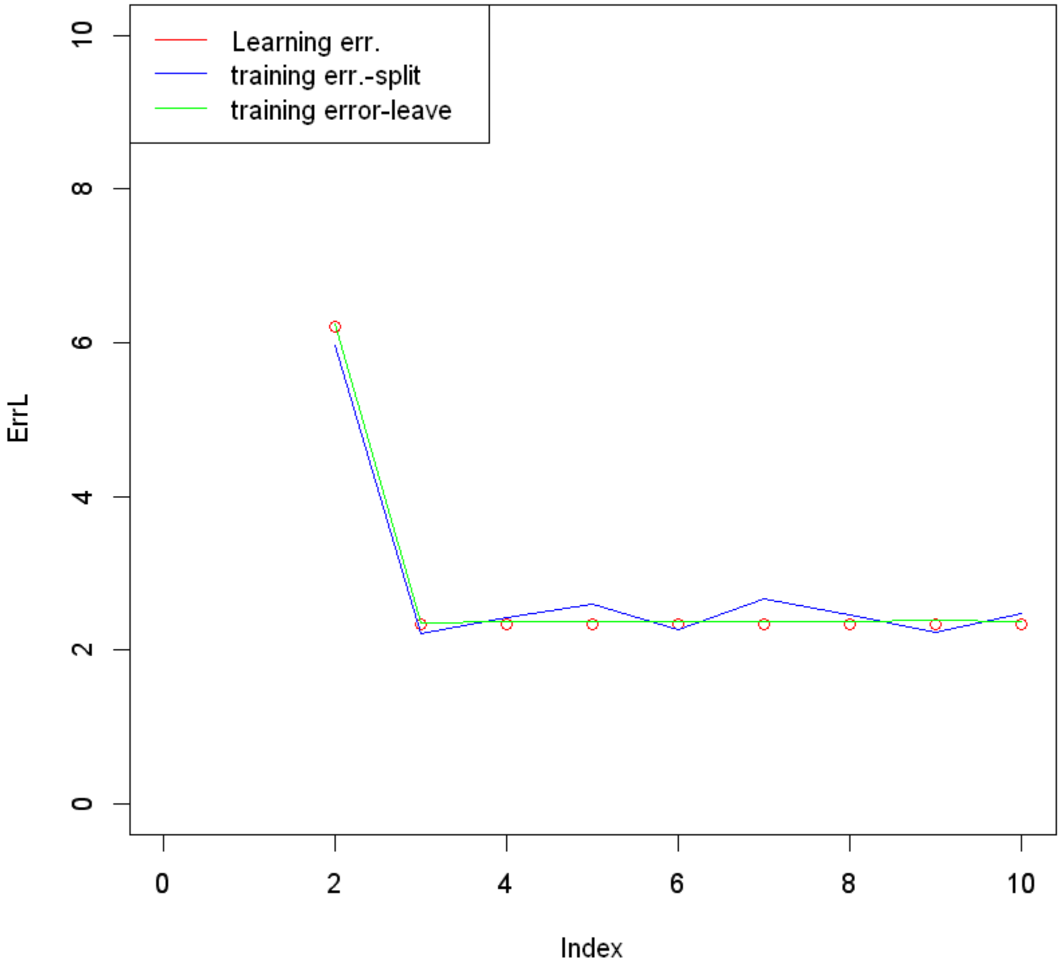
2.6 Learning process - with “leave one out method”

2.30038251159155



**The error found here : 2.31, can be relied on.

2.6.1 Comparing errors between evaluation done on training, done by split method, done by leave one out method


Learning err.
training err.-split
training error-leave
ErrL
Index