BLACKWELL ELECTRONICS

Fabiola Reyes Mosquera Barcelona, November 19, 2019.

INTRODUCTION

We have been requested by Blackwell Electronics to find predictions concerning how far a certain car travel based on speed.

Also to make predictions concerning the petal length through using the petal´s width by correcting the script which is in the plan of attack.

A summary of the experience using RStudio.

Your predictions concerning how far a certain car can travel based on speed. (From the R tutorial.)	**DONE**
Your predictions concerning the petal length through using the petal's width. (From the Find the Errors task.)	**DONE**
The errors/warning messages that you encountered and how you overcame them.	**DONE**
Was it straightforward to install R and RStudio?	**DONE**
Was the tutorial useful? Would you recommend it to others?	**YES, and I will because it has help me to do a better and faster analysis**
What are the main lessons you've learned from this experience?	**Call library,data summary, test set, Liner Regresion Model**
What recommendations would you give to other employees who need to get started using R and doing predictive analytics in R instead of Rapidminer?	**DONE**

Your predictions concerning how far a certain car can travel based on speed. (From the R tutorial.)	YES
Your predictions concerning the petal length through using the petal's width. (From the Find the Errors task.)	YES
The errors/warning messages that you encountered and how you overcame them.	Yes and helpe me to focus on the details
What recommendations would you give to other employees who need to get started using R and doing predictive analytics in R instead of Rapidminer?	I will tell the person who might used that is better to start with RapidMinner to have an idea of the models, because RM is more friendly.

Introduction of RStudio

Before to start, we got familiar with the program, the 3 boxes that we will be using such us: RScript, Console and the Environment. This consoles has different functions: The RScript is where we can save our codes and also were we should write the code before it appears in the Console. On the other hand, the environment is where variables and new variables can be observed. Also to get to know how save files in a directory or add new files in new directory's.

Get Start on R

Everything that we will be performing will be writing on the RScript after goes to the console, I have start with some codes to preprocess our data.

SCRIPT	FUNTION
library(readr)	Add the package readr
cars <- read.csv("C:\\Respaldo FR\\UBIQUM\\PROJECT3\\cars.csv")	Call my dataset
boxplot(cars$distance.of.car)	Identify Outliers
cars <- cars[cars$distance.of.car < 120,]	Eliminate outlier value
boxplot(cars$distance.of.car)	Confirm outlier is not any more

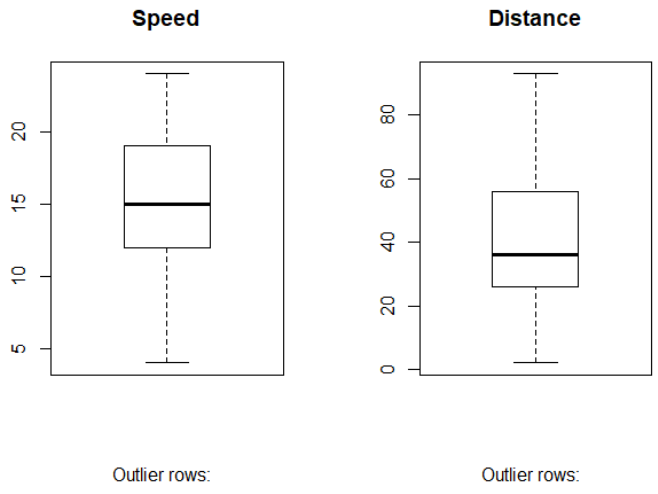


Figure 1-Boxplot

Now, we have 49 observations with 3 variables.

After this analysis we are ready to start with our model prediction, wich will have the following steps:

1.Define Training set

2.Testing

3.Define model

4.Training model

5.Prediction

1. Define Training Set and testing

In this step we will decide which model we will use, for the Car´s data set we have to predict the speed based on the distance, this is a regression problem because what we want to predict is "distance" which is a numerical variable.
Once we understand this, we are ready to continue with RStudio
We have to define training and testing set, so we can be able to define the percentage of our training set.

SCRIPT	**FUNTION**
set.seed(123)	Sequence of random numbers staring with the next 1,2,3
trainSize<-round(nrow(cars)*0.70)	70% for Train set
testSize<-nrow(cars)-trainSize	30% for Testing set
trainSize	Distance 34
testSize	Distance 15
training_indices<-sample(seq_len(nrow(cars)),size =trainSize)	Making the intervals [1:34]
trainSet<-cars[training_indices,]	Intervals
testSet<-cars[-training_indices,]	Intervals

All this, can be seen on the environment,

Data and Values

CARS, 49 of 3 variables-

TESTSET, 15 obs, 3 variables

TRAINSET, 34 obs, 3 variables.

2. Define Model-training model and prediction

Once, we have both sets, I will define my Liner Regression model by the definition of the variables and the selecting of the set (Trainset), in the first table we obtain negative values for the predicted distance, wich is why we will add a 0 in the intercept to get absolute values only.

SCRIPT	**FUNTION**
FRL<-lm(distance.of.car~ speed.of.car +, trainSet)	Determinated the name of the model, mention wich variable is gonna be predicted and from wich set the model
summary(FRL)	
Residuals:	
Min 1Q Median 3Q Max	
-7.942 -3.666 -0.295 1.884 9.951	
Coefficients:	
Estimate Std. Error t value Pr(>|t|)	
(Intercept) -30.459 2.596 -11.7 3.9e-13 ***	
speed.of.car 4.729 0.162 29.2 < 2e-16 ***	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1	
Residual standard error: 4.54 on 32 degrees of freedom	
Multiple R-squared: 0.964, Adjusted R-squared: 0.963	
F-statistic: 850 on 1 and 32 DF, p-value: <2e-16	
predictedDistance<-predict(FRL,testSet)	Get the predicted values of the Distance
1 2 6 16 18 20 22 24 32 33 36 38 41 -11.5 -11.5 12.1 31.0 31.0 35.8 35.8 40.5 54.7 54.7 59.4 59.4 64.1 44 46 73.6 83.0	15 Predicted Distance but as it can be observed, it has negative values

SCRIPT	FUNTION
FRL<-lm(distance.of.car~ speed.of.car +0, trainSet)	In order to avoid negative values, we add a 0 in the intercept
Residuals: **Min 1Q Median 3Q Max** -12.13 -9.27 -7.16 1.33 23.08 Coefficients: Estimate Std. Error t value Pr(>|t|) speed.of.car 2.91 0.11 26.4 <2e-16 *** --- Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 Residual standard error: 10.3 on 33 degrees of freedom Multiple R-squared: 0.955, Adjusted R-squared: 0.953 F-statistic: 698 on 1 and 33 DF, p-value: <2e-16	
predictedDistance<-predict(FRL,testSet)	Predict by using the FRL model with the trainset BUT in the Test set
predictedDistance 1 2 6 16 18 20 22 24 32 33 36 38 41 44 46 11.7 11.7 26.2 37.9 37.9 40.8 40.8 43.7 52.4 52.4 55.4 55.4 58.3 64.1 69.9	Predicted POSITIVE values

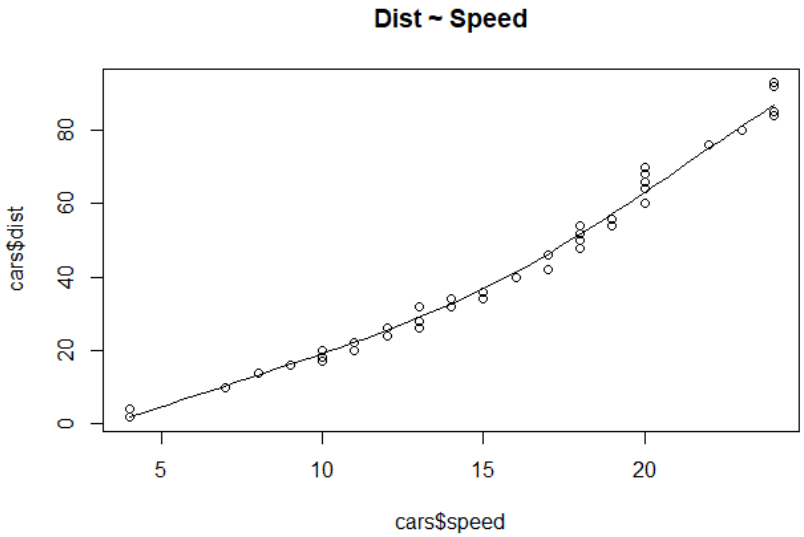
Based on this model, I will start analyzing the errors.

SCRIPT

```
> r_squared<-summary(FRL)$r.squared
> r_squared
[1] 0.955

> RMSE <- sqrt(mean(error^2))

> RMSE
[1] 8.72

> MAE <- mean(abs(error))

> MAE
[1] 7.81
```

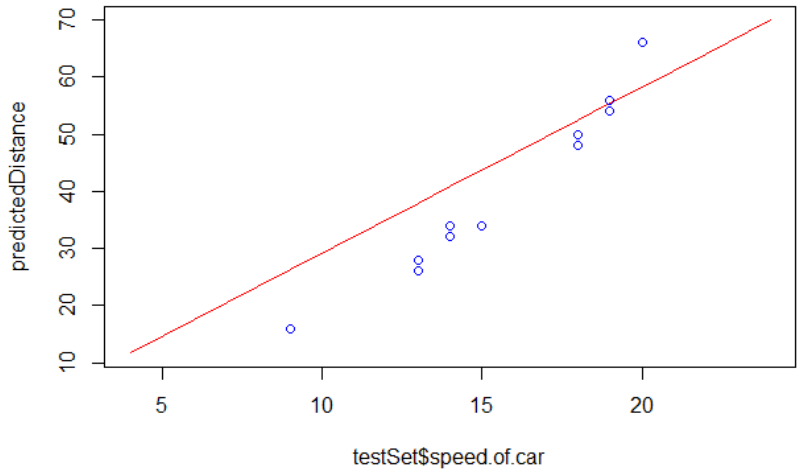
General Results

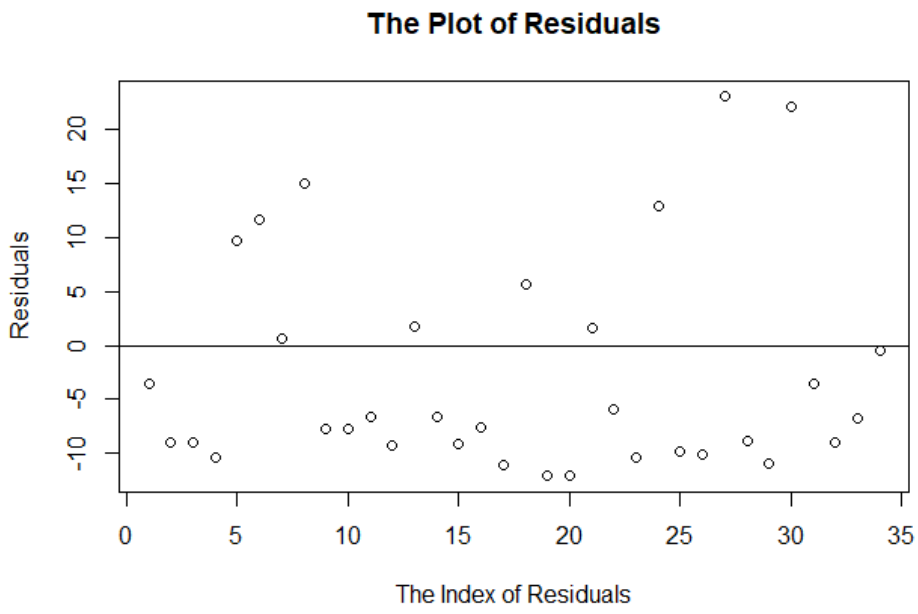
This scatter plot suggest us that there is a positive relation of the distance and speed, when the distance increase it might increase also the speed, also we verify the corraletion between this 2 variables and is positive and hight (0.97)

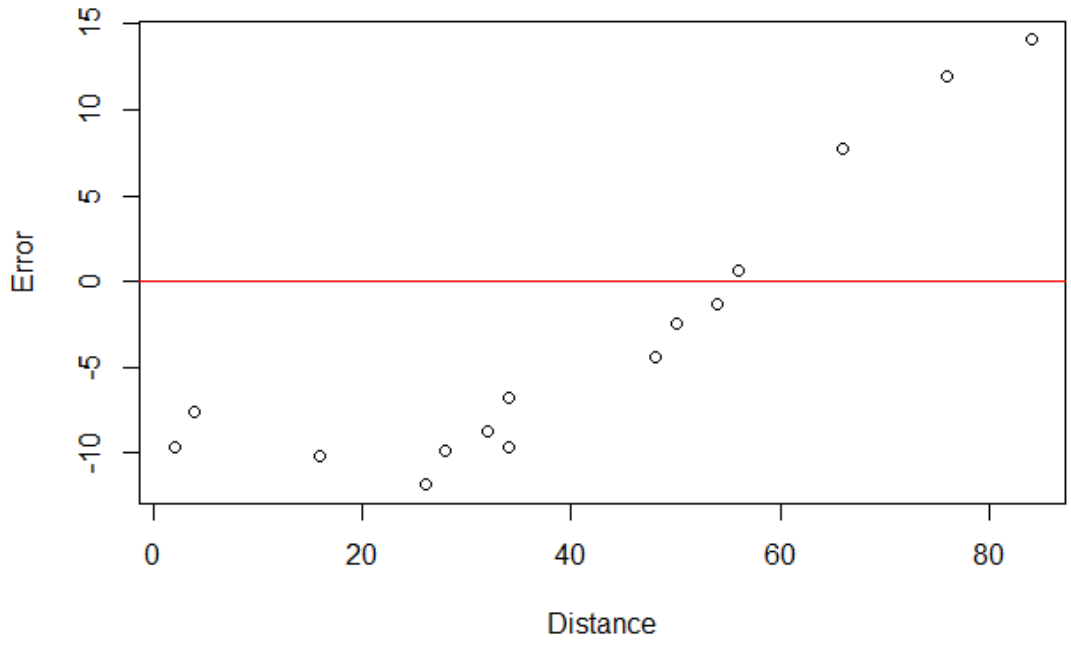


About my prediction I have the following values:

1	2	6	16	18	20	22	24	32	33	36	38	41	44	46
11.7	11.7	26.2	37.9	37.9	40.8	40.8	43.7	52.4	52.4	55.4	55.4	58.3	64.1	69.9







About this model, I can conclude that both variables are correlated and also that the models fits, as we can see the errors are well distributed and in a normal distribution, also are close to the line of the model meaning that I make “small” erros.

About Iris Data Set

We have corrected the Script.

SCRIPT	SCRIPT CORRECTED	FUNTION
install.packages(readr)	install.packages("readr")	Installation of the package
library("readr")	library("readr")	Call the library
IrisDataset <- read.csv(iris.csv)	iris <- read.csv("C:\\Respaldo FR\\UBIQUM\\PROJECT3\\iris.csv")	Call my data set from where is
attributes(IrisDataset)	attributes(iris)	For know whatever is in the data set, names, variables and class.
summary(risDataset)	summary(iris)	For know Min,Mean,Quartiles,Max of each varibale
str(IrisDatasets)	str(iris)	Data frame
names(IrisDataset)	names(iris)	Names of the variabñes
hist(IrisDataset$Species)	hist(iris$Species)	Change a categorical variable to numeric
plot(IrisDataset$Sepal.Length	plot(iris$Sepal.Length,iris$Species)	Graphic
qqnorm(IrisDataset)	qqnorm(iris$Sepal.Length)	Graphic
IrisDataset$Species<- as.numeric(IrisDataset$Species)	iris$Species<- as.numeric(iris$Species)	Change a categorical variable to numeric
set.seed(123)	set.seed(123)	Sequence of random numbers staring with the next 1,2,3
trainSize <- round(nrow(IrisDataset) * 0.2)	trainSize<- round(nrow(iris)*0.80)	To decide percentage por train test
testSize <- nrow(IrisDataset) - trainSet	testSize<-nrow(iris)-trainSize	Decide percentage por train test
trainSizes	trainSize	Distance 120
testSize	testSize	Distance30
trainSet <- IrisDataset[training_indices,]	trainSet<- iris[training_indices,]	Making Intervals
testSet <- IrisDataset[-training_indices,]	trainSet<- iris[training_indices,]	Making Intervals
set.seed(405)	set.seed(123)	Sequence of random numbers staring with the next 1,2,3
trainSet <- IrisDataset[training_indices,]	trainSet<- iris[training_indices,]	Making Intervals based on the new set.seed
testSet <- IrisDataset[-training_indices,]	trainSet<- iris[training_indices,]	Making Intervals based on the new set.seed
LinearModel<- lm(trainSet$Petal.Width ~ testingSet$Petal.Length)	LinearModeliris<- lm(Petal.Length~ Petal.Width,trainSet)	Modeling

summary(LinearModel)	**summary(LinearModeliris)**	**Residuals:** **Min 1Q Median 3Q Max** **-1.3153 -0.3266 -0.0269 0.2761 1.4067** **Coefficients:** **Estimate Std. Error t value Pr(>|t|)** **(Intercept) 1.0825 0.0869 12.5 <2e-16 *** Petal.Width 2.2220 0.0599 37.1 <2e-16 *** --- Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 Residual standard error: 0.506 on 118 degrees of freedom Multiple R-squared: 0.921, Adjusted R-squared: 0.92 F-statistic: 1.37e+03 on 1 and 118 DF, p-value: <2e-16**
prediction<-predict(LinearModeltestSet)	**predictedPetal.Width<-predict(LinearModeliris,testSet)**	**1 2 3 11 18 19 28 33 36 48 55 56 57 58 59 61** **1.53 1.53 1.53 1.53 1.75 1.75 1.53 1.30 1.53 1.53 4.42 3.97 4.64 3.30 3.97 3.30** **62 65 66 70 77 83 84 98 100 105 113 125 131 141** **4.42 3.97 4.19 3.53 4.19 3.75 4.64 3.97 3.97 5.97 5.75 5.75 5.30 6.42**
predictions	**predictedPetal.Width**	

About the Regression model for iris, we have the next results:

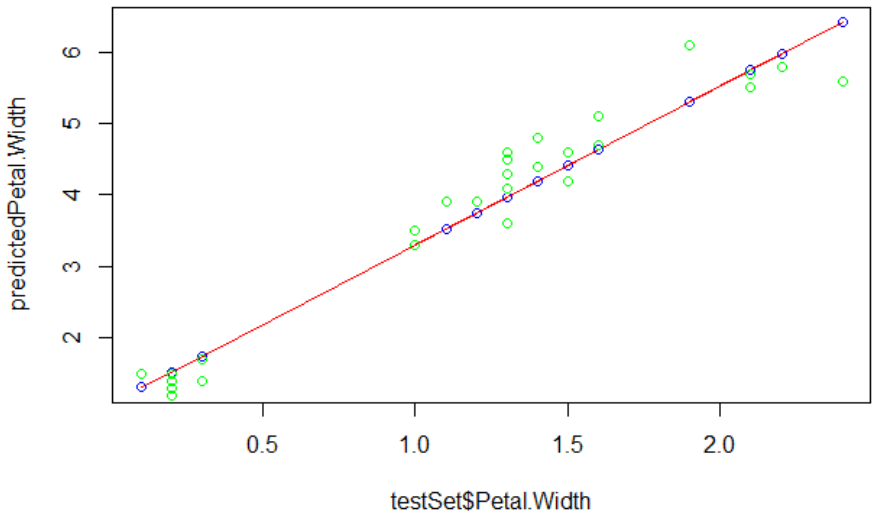
Predicted values:

1 2 3 11 18 19 28 33 36 48 55 56 57 58 59 61
1.53 1.53 1.53 1.53 1.75 1.75 1.53 1.30 1.53 1.53 4.42 3.97 4.64 3.30 3.97 3.30

62 65 66 70 77 83 84 98 100 105 113 125 131 141
4.42 3.97 4.19 3.53 4.19 3.75 4.64 3.97 3.97 5.97 5.75 5.75 5.30 6.42

Erros:

SCRIPT	FUNTION
RMSE <- sqrt(mean(error^2)) > RMSE [1] 0.347 > > MAE <- mean(abs(error)) > MAE [1] 0.27 > > MAPE<-mean(abs(error/testSet$Petal.Length)) > MAPE [1] 0.08492977	



As we can see in the graphic, I compared the predicted value vs the test set and add the lineal model so it can be inferred that the distance are smaller and maybe is a good model.