15th September, 2017

A01630510

Victor Manuel García Rosales

Name of the course: TC2007

Métodos Cuantitativos y Simulación

The design of experiments DOE constructed

Brief description of the experiment, how was physically executed.

- The experiment was executed by teams conformed of 2 people
- First of all, we only cut and fold every single paper that we had. The results were several paper aircrafts with different features.
- We dropped every single aircraft from the 3rd floor of a building.
- Then we measured 3 different times, the total time that lasts the aircraft to fall down from the 3rd floor.
- Then we only get the average from that 3 different measurements.
- Our time is denominated as our dependent variable, so the independent variables are our features over the aircraft.
- The linear regression model is developed in the next following lines, with python & R.
- The results that we obtained were the coefficients and the coefficient determinant, as well as our residuals and our predicted values.
- Finally, we plot every single comparison between the average time & residuals, average time & predicted values, every feature & Time predicted.

Compute the linear regression model (y=B0+B1x1+B2x2.....) using both:

R Code

```
setwd("/Users/victormanuel/Downloads") #Set the directory
df <- read.csv("data-set.csv", header = FALSE)
names(df) <- c('Clip','Body', 'Wing', 'Papertype','Fly1','Fly2',
'Fly3', 'FlyAvg')
model <- lm(FlyAvg ~ Clip + Body + Wing + Papertype, df)
model #Define the model & then execute it

Call:
lm(formula = FlyAvg ~ Clip + Body + Wing + Papertype, data = df)

Coefficients:
(Intercept)         Clip         Body         Wing    Papertype
     6.7604       0.3708       0.6542       1.5792      -3.7625

summary(model)

Call:
lm(formula = FlyAvg ~ Clip + Body + Wing + Papertype, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-0.93125 -0.67708 -0.05625  0.33333  2.10208

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.7604     0.5099  13.259 4.14e-08 ***
Clip          0.3708     0.4560   0.813  0.43337
Body          0.6542     0.4560   1.434  0.17926
Wing          1.5792     0.4560   3.463  0.00531 **
Papertype    -3.7625     0.4560  -8.250 4.87e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9121 on 11 degrees of freedom
Multiple R-squared:  0.8827,	Adjusted R-squared:   0.84
F-statistic: 20.69 on 4 and 11 DF,  p-value: 4.453e-05

df$predicted <- predict(model) #Compare the predicted time vs the
residuals & plot them
```

```r
df$residuals <- residuals(model)
install.packages("dplyr")
library(dplyr)
df
plot(df$FlyAvg, df$residuals,ylab="Residuals", xlab="FlyAvg", pch =
16, cex = 1.3, col = "blue", main = "Residuals vs Avg Time")
abline(0,0, col = "red")
plot(df$predicted, df$residuals,ylab="Residuals", xlab="Predicted(Y)",
pch = 16, cex = 1.3, col = "blue", main = "Residuals vs Predicted")
abline(0,0, col = "red")

#Get an histogram of the residuals distribution & then plot it
r <- df$residuals
h <- hist(r, breaks = 10, density = 10, col = "lightgray", xlab =
"Residuals", main = "Histogram of residuals")
 xfit <- seq(min(r), max(r), length = 40)
 yfit <- dnorm(xfit, mean = mean(r), sd = sd(r))
 yfit <- yfit   * diff(h$mids[1:2]) * length(r)
 lines(xfit, yfit, col = "black", lwd = 2)
 residuals.stdres = rstandard(model)
 qqnorm(residuals.stdres, ylab = "Standardized Residuals", xlab =
"Normal Scores", pch = 16, cex = 1.3, col="blue", main = "" )
 qqline(residuals.stdres)
```

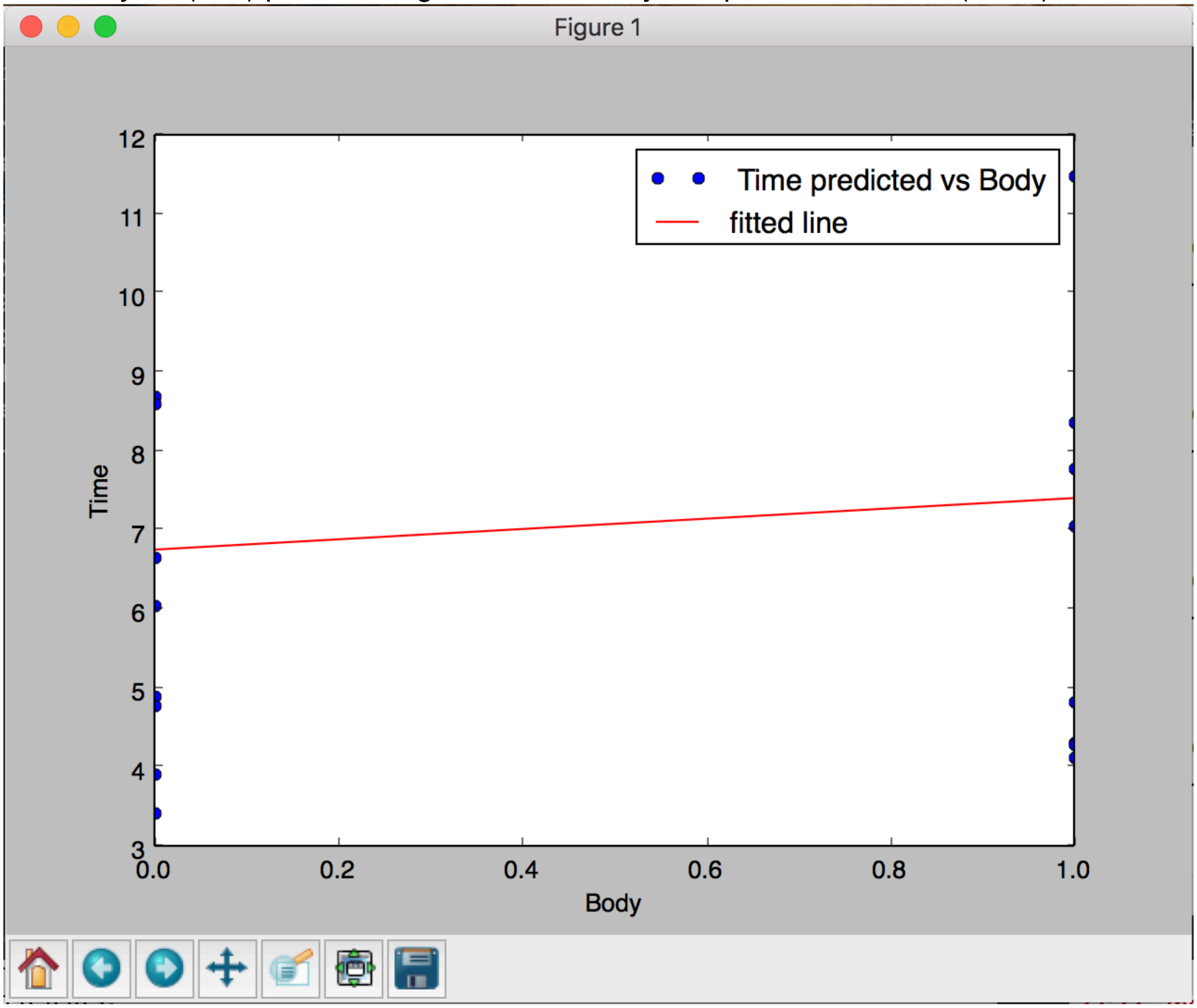
◦**Python**

```python
import matplotlib.pyplot as plt
import numpy as np
import csv
from sklearn import linear_model
# Load dataset, no header
filename = 'data-set.csv'
raw_data = open(filename, 'rt')
data = np.loadtxt(raw_data, delimiter=",")
# x y
matrix_X = data[:,0:4]
matrix_Y= data[:,7]
# Create MuLTIPLE linear regression object
modelM = linear_model.LinearRegression()
modelM.fit(matrix_X,matrix_Y)
# The coefficients
print('Coefficients:')
print('B0: \n', modelM.intercept_)
print('B1: \n', modelM.coef_[0])
print('B2: \n', modelM.coef_[1])
print('B3: \n', modelM.coef_[2])
```

```
print('B4: \n', modelM.coef_[3])
# R^2

print('R^2: \n', modelM.score(matrix_X, matrix_Y))
# Different plots of every feature vs the predicted time
plt.plot(data[:,0],matrix_Y, 'o', label = ' Time predicted vs
Clip')
plt.plot(data[:, 0], modelM.intercept_ +
modelM.coef_[0]*data[:,0], 'r', label = 'fitted line')
plt.xlabel('Clip')
plt.ylabel('Time')
plt.legend()
plt.show()
plt.plot(data[:,1],matrix_Y, 'o', label = ' Time predicted vs
Body')
plt.plot(data[:, 1], modelM.intercept_ +
modelM.coef_[1]*data[:,1], 'r', label = 'fitted line')
plt.xlabel('Body')
plt.ylabel('Time')
plt.legend()
plt.show()
plt.plot(data[:,2],matrix_Y, 'o', label = ' Time predicted vs
Wing')
plt.plot(data[:, 2], modelM.intercept_ +
modelM.coef_[2]*data[:,2], 'r', label = 'fitted line')
plt.xlabel('Wing')
plt.ylabel('Time')
plt.legend()
plt.show()
plt.plot(data[:,3],matrix_Y, 'o', label = ' Time predicted vs
Paper Type')
plt.plot(data[:, 3], modelM.intercept_ +
modelM.coef_[3]*data[:,3], 'r', label = 'fitted line')
plt.xlabel('Paper Type')
plt.ylabel('Time')
plt.legend()
plt.show()
```

• In a Layout (2, 2) plot the "Flight time" vs every independent variable (i.e., x)



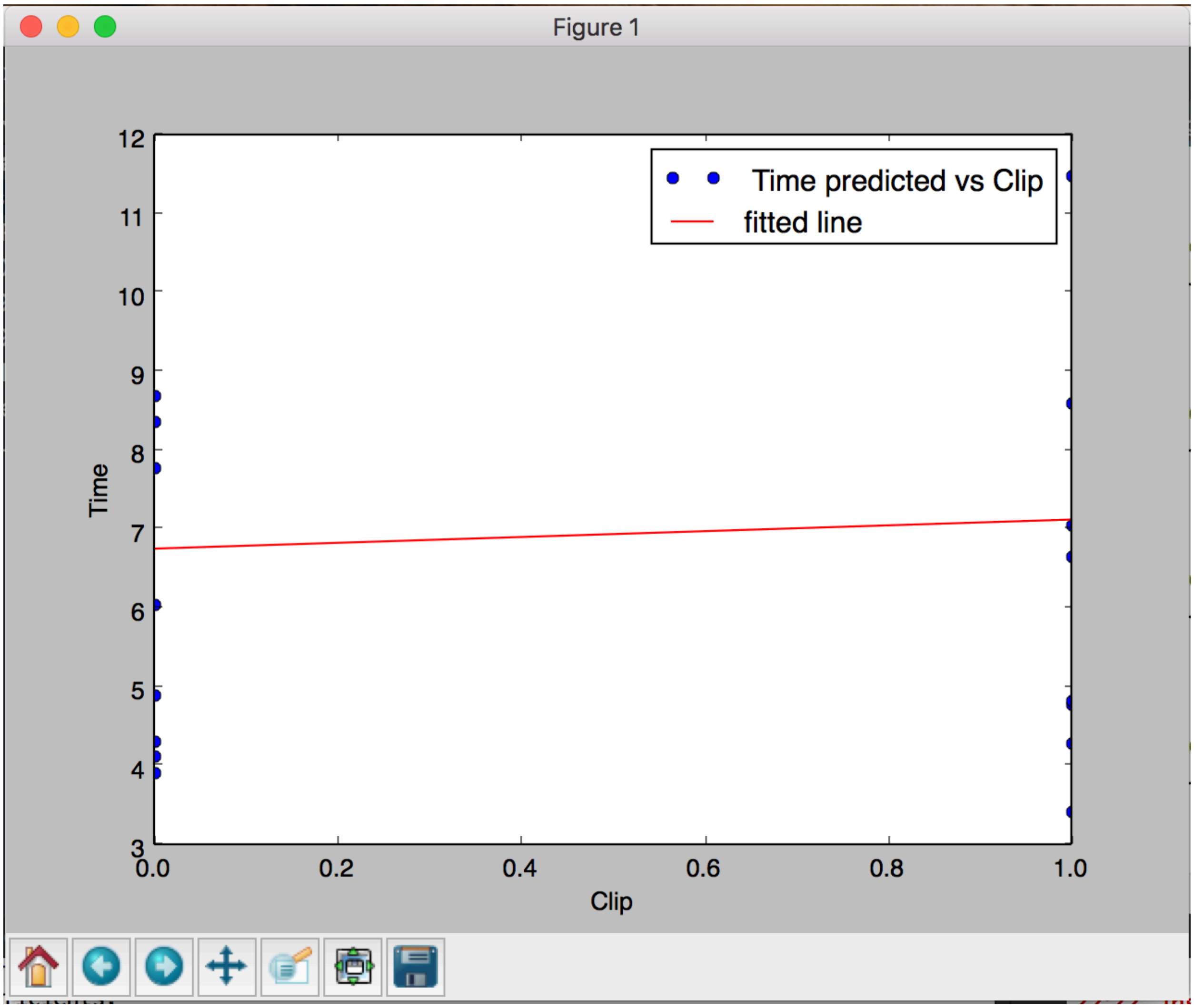


Figure 1
Time predicted vs Clip
fitted line
Time
Clip
12
11
10
9
8
7
6
5
4
3
0.0
0.2
0.4
0.6
0.8
1.0

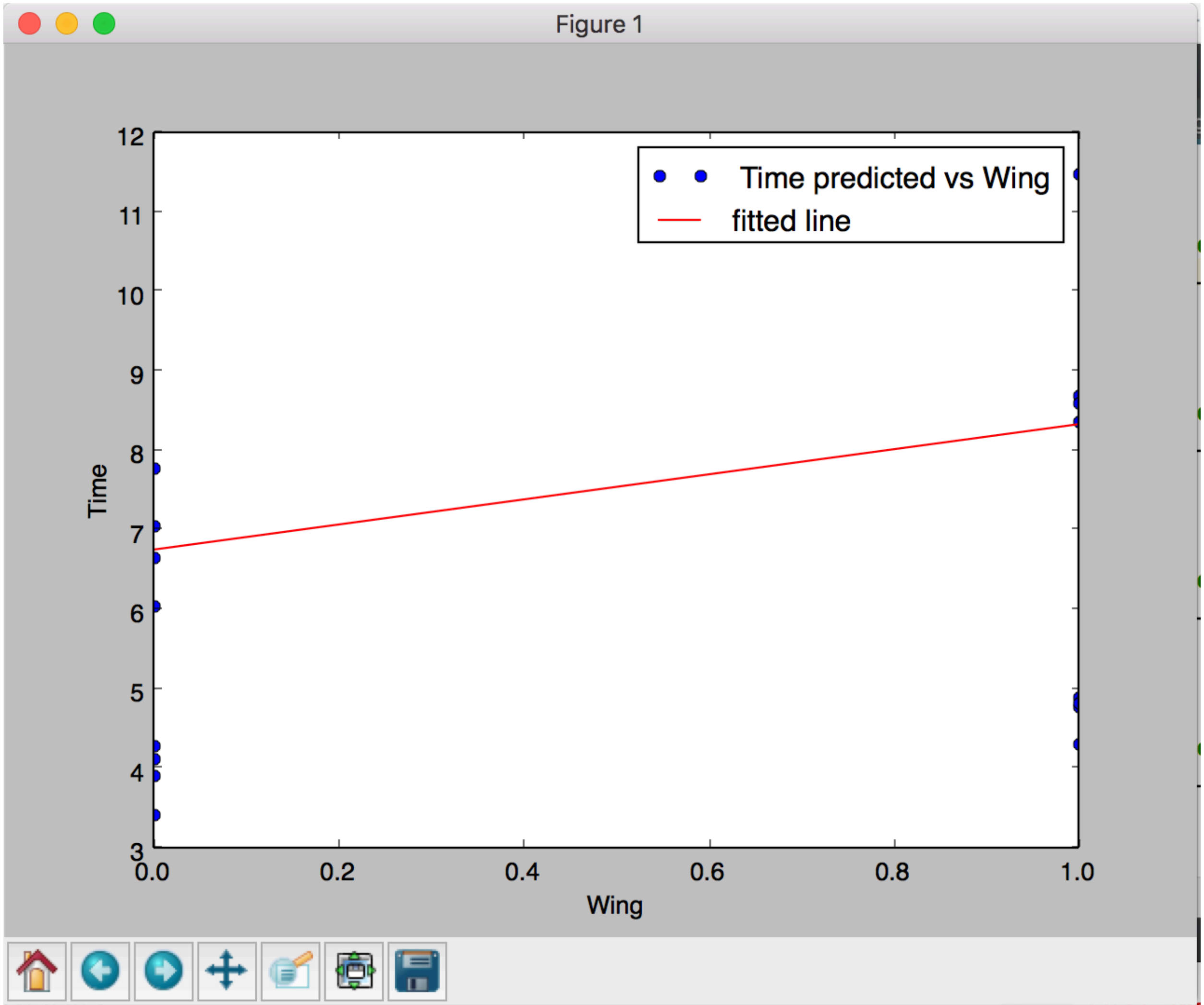
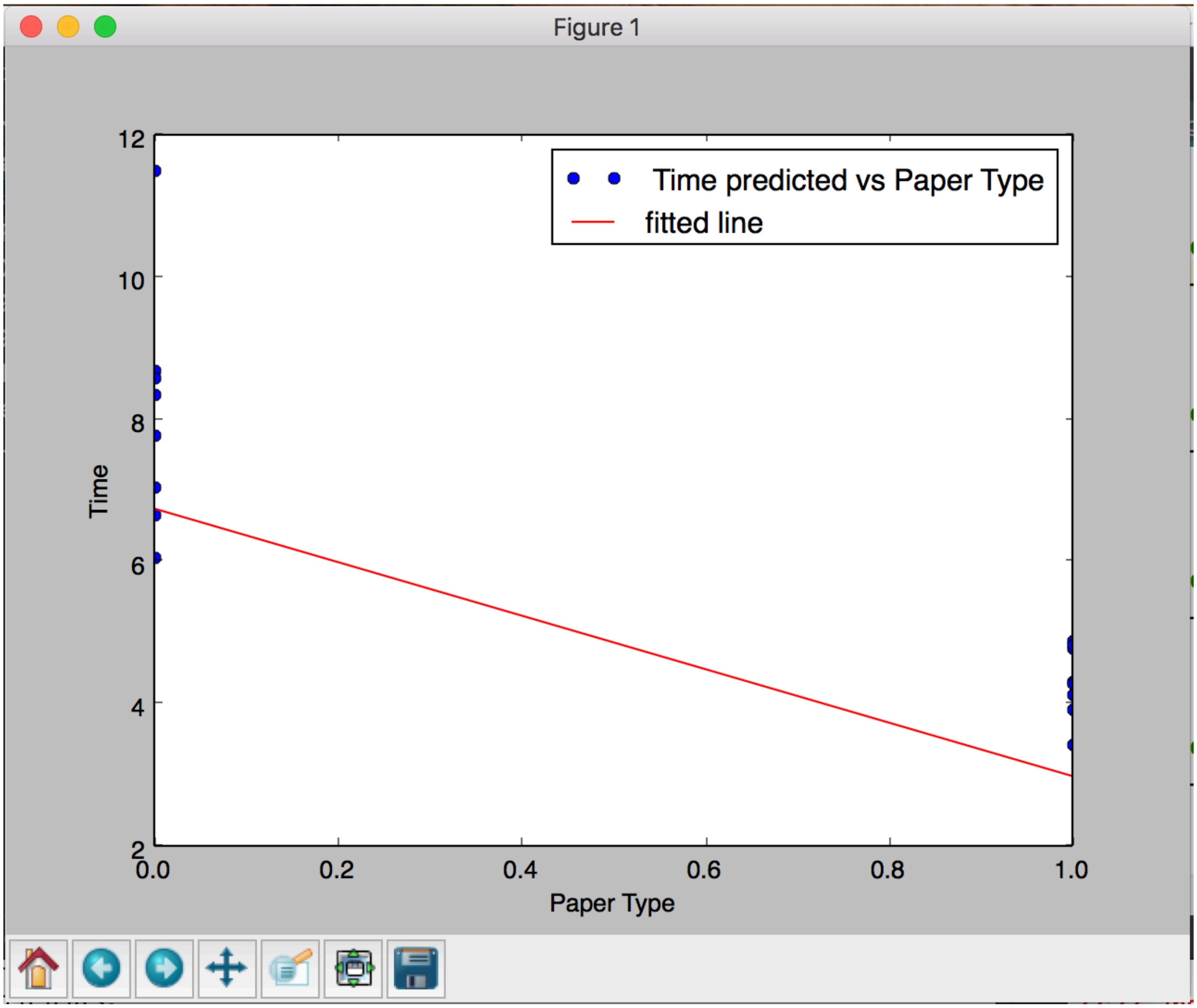
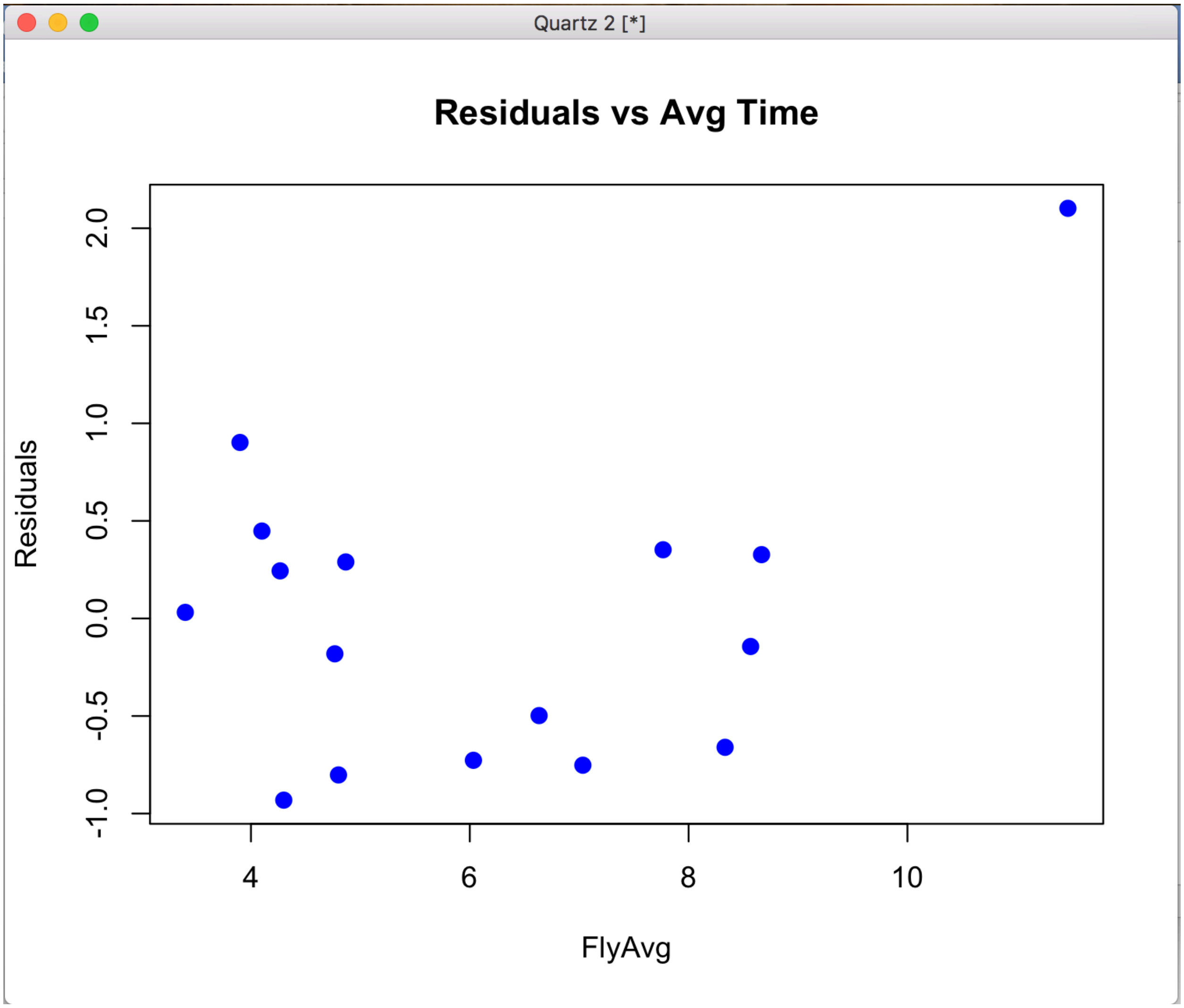
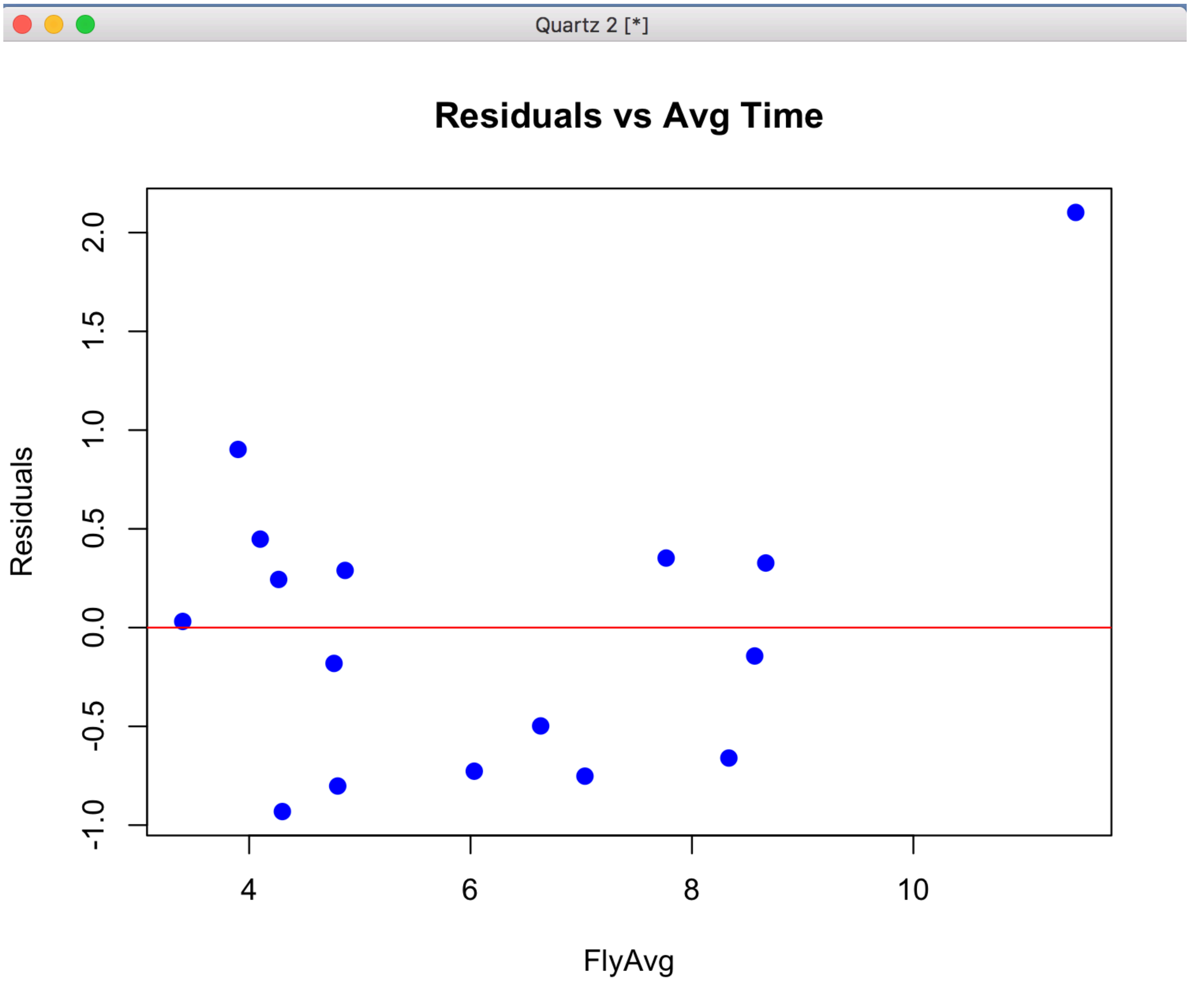


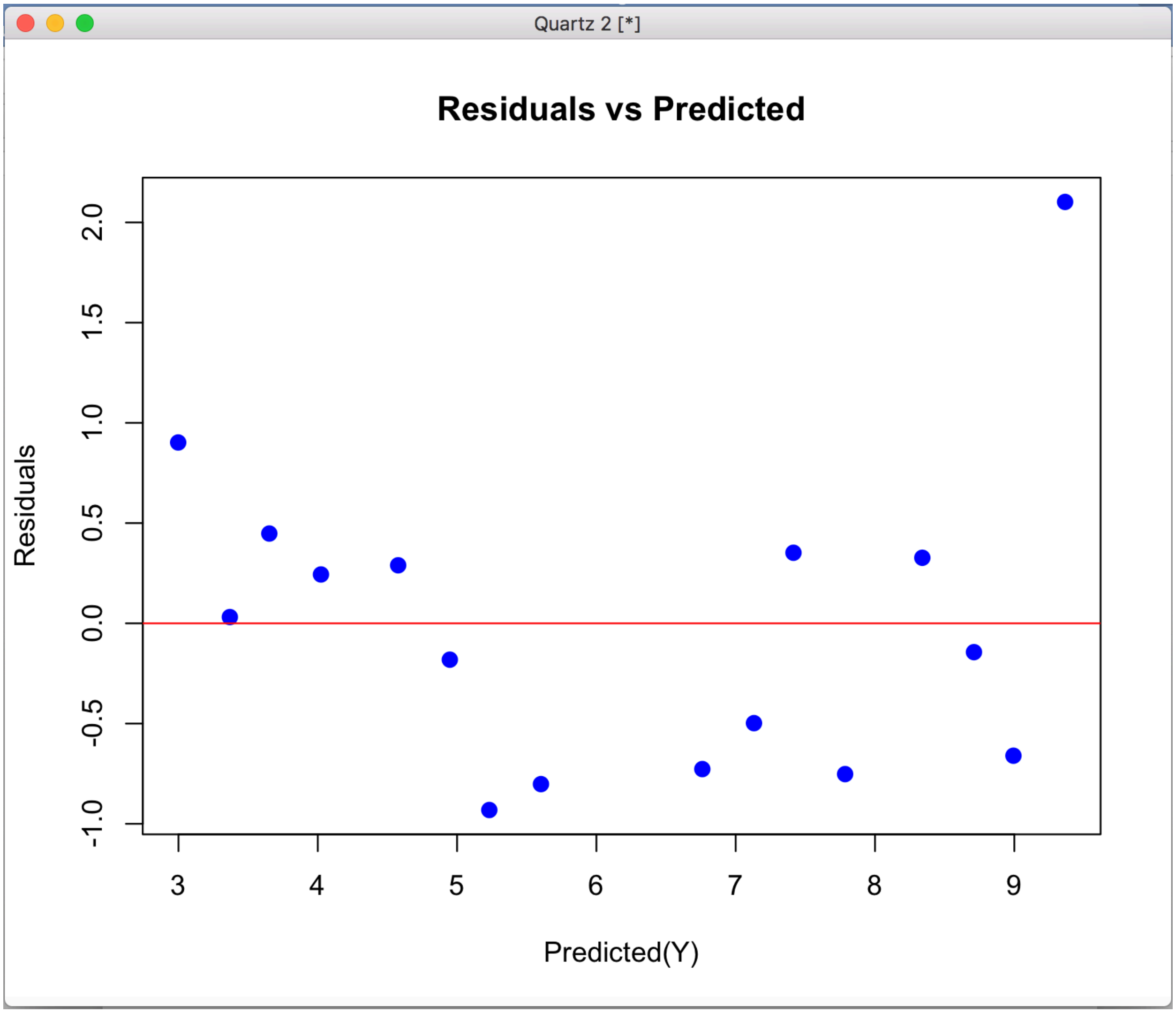
Figure 1
Time predicted vs Wing
fitted line
Time
Wing
12
11
10
9
8
7
6
5
4
3
0.0
0.2
0.4
0.6
0.8
1.0

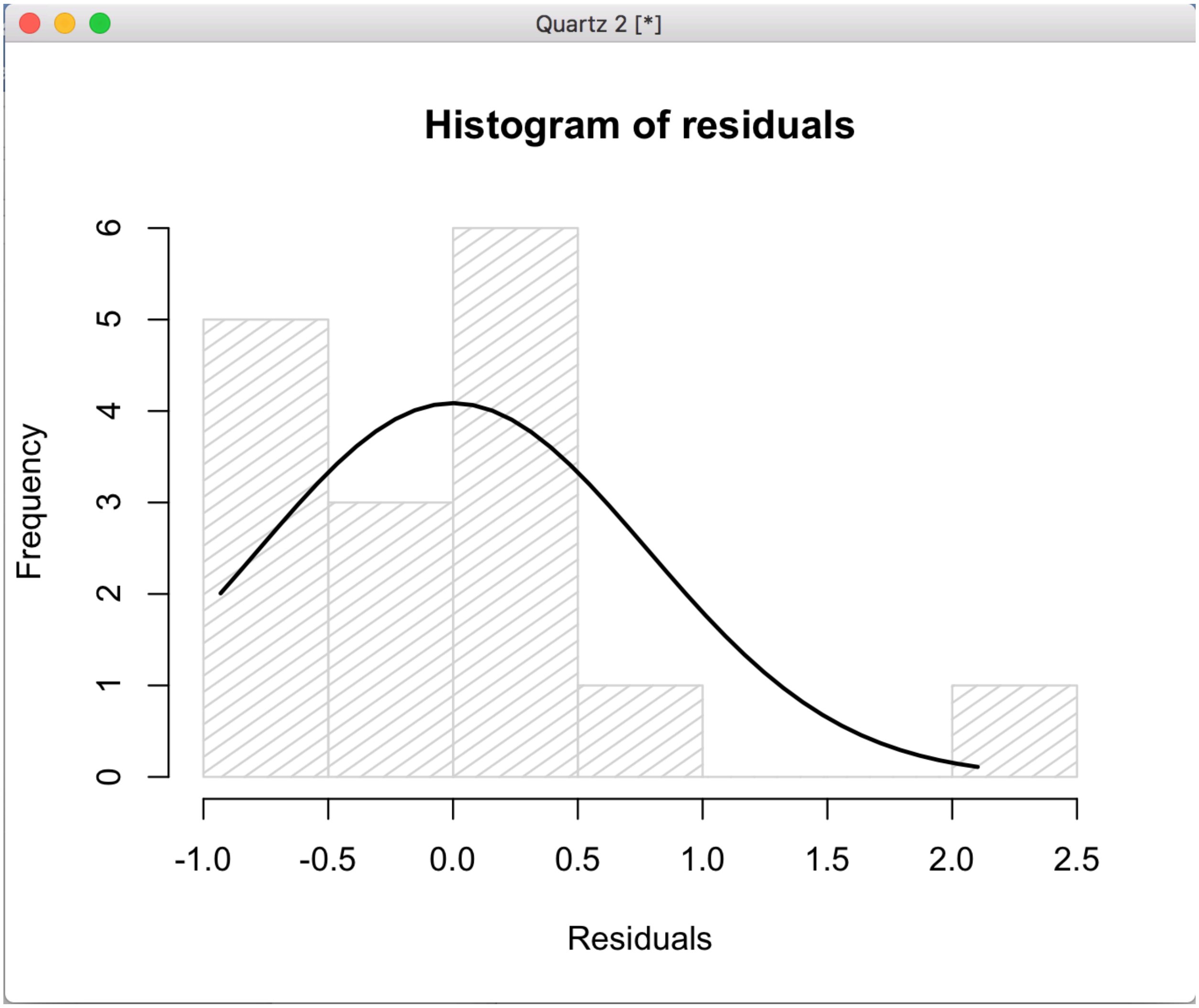


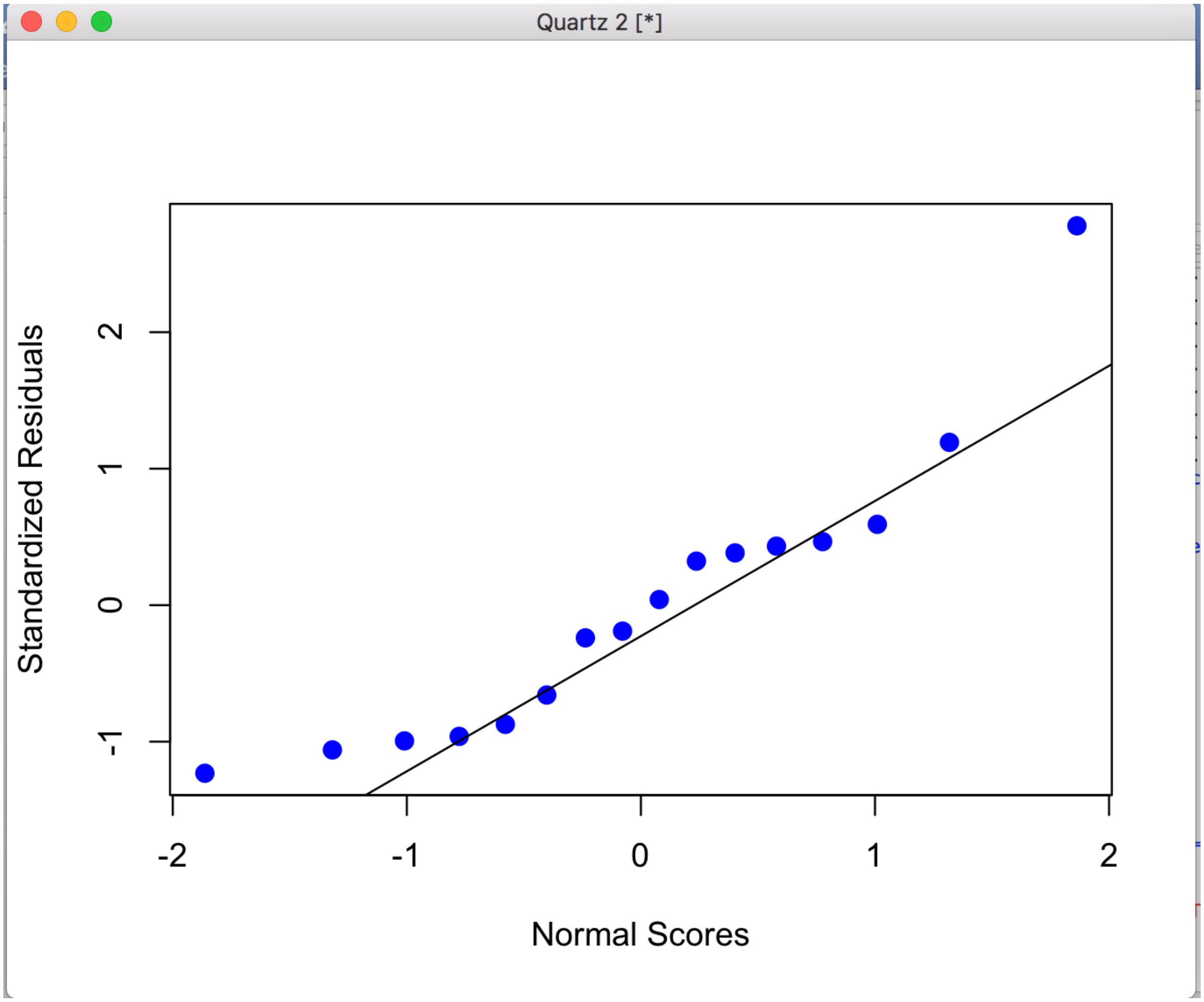
- Add to the data set the “Flight time predicted” and the “Residuals” (as seen in class, MCS_9100_ResidualAnalysisFromLinearRegression.pdf, slide 15)
- Analysis of Residuals as seen in class
  - R^2
  - Plot of Residuals vs predicted.


Quartz 2 [*]
Residuals vs Avg Time
Residuals
FlyAvg
2.0
1.5
1.0
0.5
0.0
-0.5
-1.0
4
6
8
10


Quartz 2 [*]
Residuals vs Avg Time
Residuals
FlyAvg
-1.0
-0.5
0.0
0.5
1.0
1.5
2.0
4
6
8
10


Quartz 2 [*]
Residuals vs Predicted
Residuals
Predicted(Y)
2.0
1.5
1.0
0.5
0.0
-0.5
-1.0
3
4
5
6
7
8
9


Quartz 2 [*]
Histogram of residuals
Frequency
0
1
2
3
4
5
6
-1.0
-0.5
0.0
0.5
1.0
1.5
2.0
2.5
Residuals



Explanation of the results of analysis of residuals, and normality of data

What we can say is that our model nearly matches the data, so it's very close to 1 (the coefficient of termination R^2) which means that is not perfect, but is actually a good model. The normality of the data tends to 0 because that's where the actually points (data), trend to be in.

Our residuals (errors) are minimal in our sample, maybe there are 2 or 3 exceptions, but the actual standard is nearly close to 0. So it's good in this case.