# AssignementDS_1_YassineLAHBABI_RMD

Yassine Lahbabi

31/03/2020

# Introduction :

My answers to Assignment 1 for Data Statistics are found below.

```
# Clear everything in the workspace:
rm(list=ls())
```

## *Question 1: Simple Linear Regression*

### 1.a

In 1.a, we need to load the `salaries` dataset from the salaries.csv file. The commands needed to do this are:

```
salaries <- read.csv("salaries.csv")
```

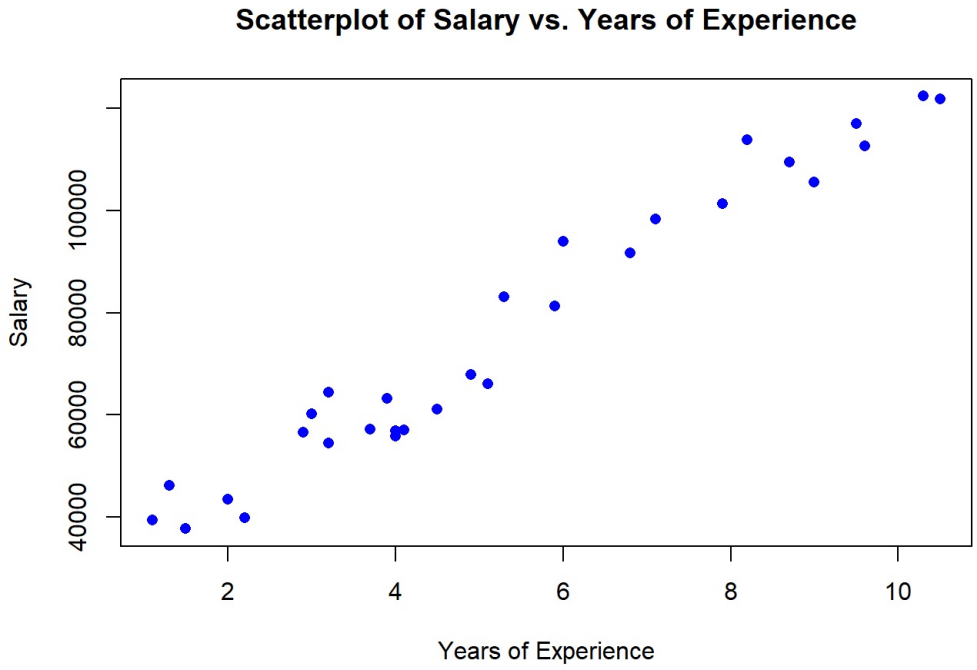
I then look at the top 6 rows to make sure the dataset has been read in correctly:

```
head(salaries)
```

```
##   YearsExperience Salary
## 1             1.1  39343
## 2             1.3  46205
## 3             1.5  37731
## 4             2.0  43525
## 5             2.2  39891
## 6             2.9  56642
```

I then need to explore and plot the data. In order to do this, I need to use the plot() function :

```
# We have 2 variables :
# 1- Salary which gives the salary of people.
# 2 - years of experience which gives the years that a person has as an experience in the professional domain.
plot(salaries$YearsExperience,salaries$Salary,
     pch = 16,
     col = "blue",
     main = "Scatterplot of Salary vs. Years of Experience",
     xlab = "Years of Experience",
     ylab = "Salary")
```




**Interpretation :** *Yes, i think there's a huge correlation between the 2 variables because we can see that there's a linear regression and a proportional relationship between both variables.*

## 1.b

In 1.b, we need to estimate the intercept and the slope, In order to do this, I need to use the formulas that we have seen before :

These are the pieces that we need :

Sxy = $\sum(x * y) - (\sum x . \sum y)/n$

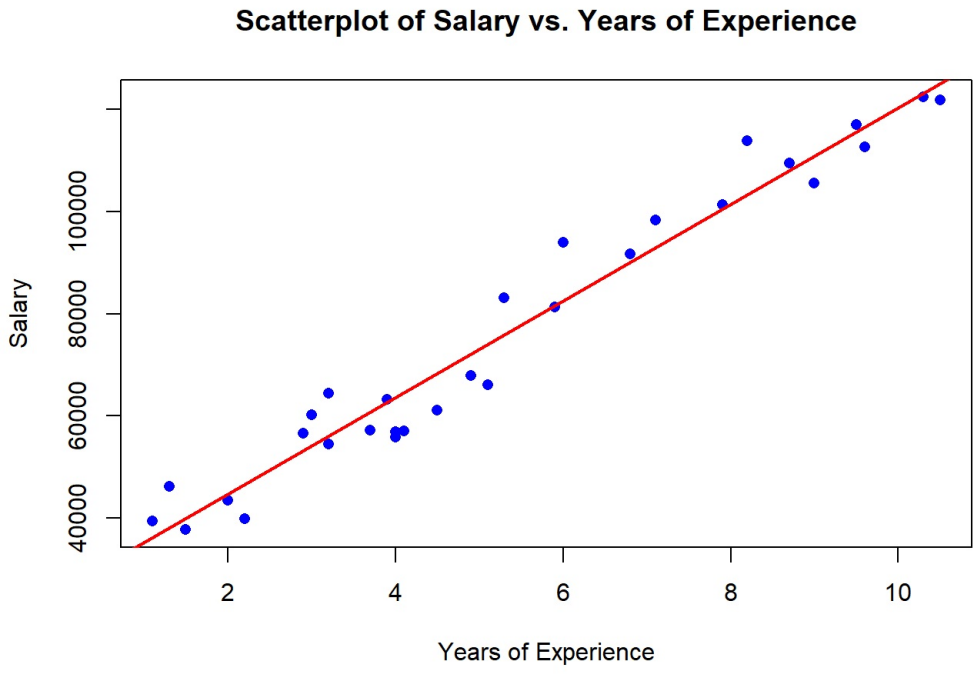
Sxx = $\sum x^2 - (\sum x^2)/n$

Syy = $\sum y^2 - (\sum y^2)/n$

To finally be able to calculate :

- The slope : b = S_xy / S_xx
- The Intercept : a = $\overline{y} - (\overline{x} . b)$

```
x = salaries$YearsExperience
y = salaries$Salary

# Finding n :
n <- length(x)

# Finding the pieces we need :
S_xy <- sum(x * y) - (sum(x) * sum(y)) / n
S_xx <- sum(x^2) - sum(x)^2 / n
S_yy <- sum(y^2) - sum(y)^2 / n

b <- S_xy / S_xx
a <- mean(y) - b*mean(x)
plot(salaries$YearsExperience,salaries$Salary,
     pch = 16,
     col = "blue",
     main = "Scatterplot of Salary vs. Years of Experience",
     xlab = "Years of Experience",
     ylab = "Salary")
abline(a, b, col = "red", lwd = 2)
```




```
b # Slope
```

```
## [1] 9449.962
```

```
a # Intercept
```

```
## [1] 25792.2
```

Now, we will check if we have the same results using the lm() function :

```
lm(y ~ x) # To check if we have approximately the same results.
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Coefficients:
## (Intercept)            x
##       25792         9450
```

We can actually see that we have the same results.

**Interpretation :**

*- The intercept means that for 0 year of experience we can hope to have a salary of 25792 euros for this company. - The slope means that for each additionnal year of experience you can earn approximately 9449 euros more.*

## 1.c

In 1.c, we are trying to conduct a hypothesis test to test if the salary is independant of a lecturer's years of experience.

We have :

- H0 : Hypothesis where B1 = 0.
- H1 : Hypothesis where B1 =/= 0.
- Significence level of alpha = 0.05

```
cor.test(x,y)
```

```
##
##  Pearson's product-moment correlation
##
## data:  x and y
## t = 24.95, df = 28, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9542949 0.9897078
## sample estimates:
##       cor
## 0.9782416
```

```
cat("We reject H0 and conclude that B1 is not equal to 0")
```

```
## We reject H0 and conclude that B1 is not equal to 0
```

To check our results :

```
model <- lm(y ~ x)
summary(model)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7958.0 -4088.5  -459.9  3372.6 11448.0
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  25792.2     2273.1   11.35 5.51e-12 ***
## x             9450.0      378.8   24.95  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5788 on 28 degrees of freedom
## Multiple R-squared:  0.957,  Adjusted R-squared:  0.9554
## F-statistic: 622.5 on 1 and 28 DF,  p-value: < 2.2e-16
```

**Conclusion :**

*We can fairly say that the employee's statement is wrong because we can clearly see that the p-value is < alpha = 0.05 and 0 is not in the confidence interval.* **Therefore, we reject H0.**

## 1.d

In 1.d, we have to calculate and interpret the R^2 value for the linear regression of Salary on Years of Experience to do so, we use the same formulas that we have used before:

```
# Using the same variances we used before :
r_squared = (S_xy^2) / (S_xx * S_yy)
r_squared
```

```
## [1] 0.9569567
```

To check our results :

```
model <- lm(y ~ x)
summary(model)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7958.0 -4088.5  -459.9  3372.6 11448.0
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  25792.2     2273.1   11.35 5.51e-12 ***
## x             9450.0      378.8   24.95  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5788 on 28 degrees of freedom
## Multiple R-squared:  0.957,  Adjusted R-squared:  0.9554
## F-statistic: 622.5 on 1 and 28 DF,  p-value: < 2.2e-16
```

**Conclusion :**

*This is a useful model because we can clearly see that we have a strong correlation as r_squared is equal to approximately 0.957 nearly approaching 1.*

## 1.e

In 1.e, we are using our model to calculate different predictions :

Let's start by initializing what we need :

```
sixhalfyear_lecturer  <- data.frame(x =6.5)
new_lecturer <- data.frame(x = 0)
```

***1 - Confidence Interval :***

- By hand :

```
confidence_interval <- function(x, y, pred_x){
  n <- length(y)
  lm.model <- lm(y ~ x)
  y.fitted <- lm.model$fitted.values
  pred_y <- b*pred_x + a

  # Finding the pieces that we need : SSE and MSE
  sse <- sum((y - y.fitted)^2)
  mse <- sse / (n - 2)

  t.val <- qt(0.975, n - 2)

  mean.se.fit <- (1 / n + (pred_x - mean(x))^2 / (sum((x - mean(x))^2))) # Standard error of the mean estimate

  # Mean Estimate Upper and Lower Confidence limits at 95% Confidence
  mean.conf.upper <- pred_y + t.val * sqrt(mse * mean.se.fit)
  mean.conf.lower <- pred_y - t.val * sqrt(mse * mean.se.fit)

  # Build data.frame of upper and lower limits calculated above, as well as the predicted y and beta 1 values
  upper <- data.frame(round(mean.conf.upper, 2))
  lower <- data.frame(round(mean.conf.lower, 2))
  fit <- data.frame(round(pred_y, 2))

  # Collect all into data.frame and rename columns
  results <- data.frame(cbind(lower, upper, fit), row.names = c('Confidence Interval'))
  colnames(results) <- c('Lower', 'Upper', 'Fit')

  return(results)
}

confidence_interval(x,y,6.5)
```

```
##                        Lower    Upper      Fit
## Confidence Interval 84864.56 89569.35 87216.96
```

- Checking if the values are good for 95% confidence interval :

```
predict(model,newdata = sixhalfyear_lecturer,interval = "confidence")
```

```
##        fit      lwr      upr
## 1 87216.96 84864.56 89569.35
```

***2 - Prediction Interval :***

- By hand:

```
prediction_interval <- function(x, y, pred_x){
  n <- length(y)
  lm.model <- lm(y ~ x)
  y.fitted <- lm.model$fitted.values
  pred_y <- b*pred_x + a

  # Finding the pieces that we need : SSE and MSE
  sse <- sum((y - y.fitted)^2)
  mse <- sse / (n - 2)

  t.val <- qt(0.975, n - 2)

  pred.se.fit <- (1 + (1 / n) + (pred_x - mean(x))^2 / (sum((x - mean(x))^2))) # Standard error of the prediction

  # Mean Estimate Upper and Lower Confidence limits at 95% Confidence
  pred.conf.upper <- pred_y + t.val * sqrt(mse * pred.se.fit)
  pred.conf.lower <- pred_y - t.val * sqrt(mse * pred.se.fit)

  # Build data.frame of upper and lower limits calculated above, as well as the predicted y and beta 1 values
  upper <- data.frame(round(pred.conf.upper, 2))
  lower <- data.frame(round(pred.conf.lower, 2))
  fit <- data.frame(round(pred_y, 2))

  # Collect all into data.frame and rename columns
  results <- data.frame(cbind(lower, upper, fit), row.names = c('Prediction Interval'))
  colnames(results) <- c('Lower', 'Upper', 'Fit')

  return(results)
}

prediction_interval(x,y,6.5)
```

```
##                        Lower    Upper      Fit
## Prediction Interval 75129.02 99304.89 87216.96
```

```
  * Checking if the values are good for 95% prediction interval :
```

```
predict(model,newdata = new_lecturer,interval = "predict")
```

```
##       fit      lwr      upr
## 1 25792.2 13053.91 38530.49
```

Now let's have a look at the point prediction :

- Point prediction :

```
predict(model,newdata = sixhalfyear_lecturer)
```

```
##        1
## 87216.96
```

**Conclusion :**

We can clearly tell to the new employee that : 1. The point prediction value is the salary they should expect to earn in the college. 2. The 95% confidence Interval gives an interval with values that takes incertitude in count to give the mean salary of people with 6.5 years of experience. 3. The 95% prediction Interval gives an interval with predicted values for the salary of a new lecturer.

## 1.f

In 1.f, we have to use our model to predict the salary of a lecturer with 30 years of experience who is about to join the company, to do so we can use the predict function along with our model.

```
thirtyyear_lecturer  <- data.frame(x =30)
# Point prediction :
predict(model,newdata = thirtyyear_lecturer)
```

```
##        1
## 309291.1
```

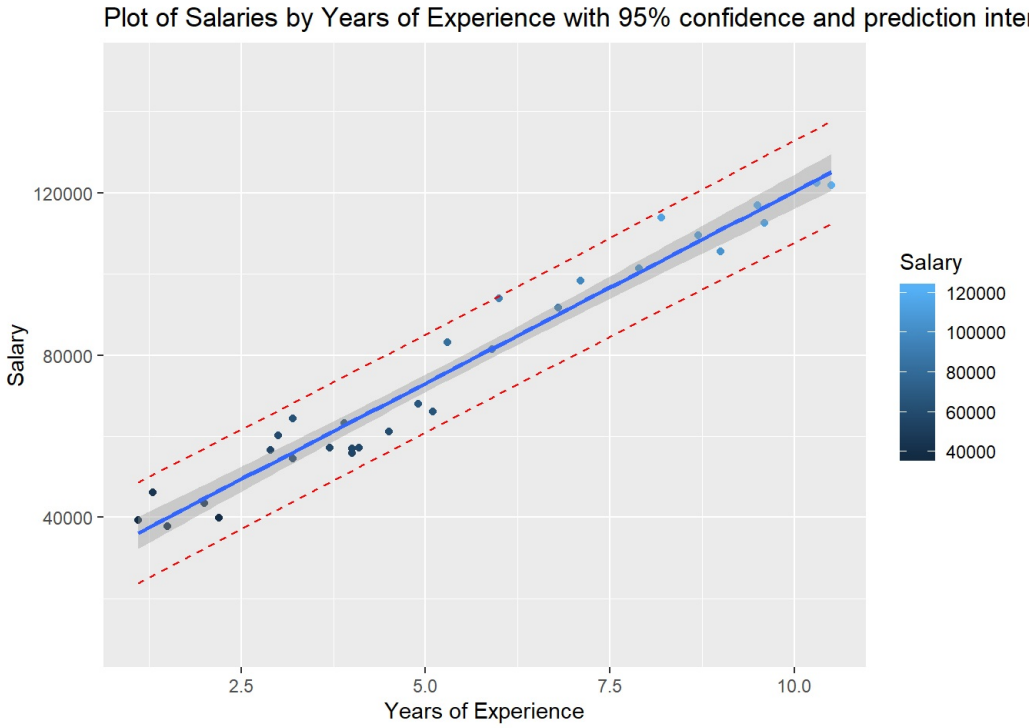
**Interpretation :**

Our model isn't efficient enough because it is quite simple. In reality we need a lot more factors like : discipline, rank, experiences,… So our model isn't perfect and accurate to have a precise idea of the salary. Because it can have really high salaries value when the years of experience are grinding with it, which is logical but to a certain point the values are just too big and do not represent the reality.

1.g

In 1.g, we have to produce a nice plot of the final model, that can be suitable for inclusion in a work presentation.

```
# Plotting using ggplot2.
library(ggplot2)

temp_var <- predict(model,interval ="prediction")
new_df <- cbind(salaries,temp_var)

# 95% confidence and prediction intervals.
ggplot(new_df, aes(YearsExperience, Salary,color = Salary))+
        geom_point() +
        geom_line(aes(y=lwr), color = "red", linetype = "dashed")+
        geom_line(aes(y=upr), color = "red", linetype = "dashed")+
        geom_smooth(method=lm, se=TRUE) +
        xlab("Years of Experience")+
        ylab("Salary")+
        ggtitle("Plot of Salaries by Years of Experience with 95% confidence and prediction intervals")+
        scale_y_continuous(name="Salary", limits=c(10000, 150000))
```




# *Question 2: Multiple Linear Regression*

## *Introduction :*

For insurers, it's important to develop models that accurately forecast medical expenses so that they can make money and profit through it. Through this project, we will try to make the most accurate model we can and build different models to best predict the medical cost of individuals given their basic information.The final prediction result can be used as a benchmark for the insurance company to establish appropriate insurance claim coverage for their contractors.

For this project we will use these libraries (ggplot2,psych and car) and load the data from the "medical-expenses.csv" file that you will find inside this project's folder :

```
# Loading libraries that we will need :
library(psych)
library(ggplot2)

# ----- STEP 1 : Load the dataset  ------

insurance <- read.csv("medical-expenses.csv")
```

## *Data Exploration :*

```
head(insurance)
```

```
##   age    sex  bmi children smoker    region expenses
## 1  19 female 27.9        0    yes southwest 16884.92
## 2  18   male 33.8        1     no southeast  1725.55
## 3  28   male 33.0        3     no southeast  4449.46
## 4  33   male 22.7        0     no northwest 21984.47
## 5  32   male 28.9        0     no northwest  3866.86
## 6  31 female 25.7        0     no southeast  3756.62
```

```
tail(insurance)
```

```
##      age    sex  bmi children smoker    region expenses
## 1333  52 female 44.7        3     no southwest 11411.69
## 1334  50   male 31.0        3     no northwest 10600.55
## 1335  18 female 31.9        0     no northeast  2205.98
## 1336  18 female 36.9        0     no southeast  1629.83
## 1337  21 female 25.8        0     no southwest  2007.95
## 1338  61 female 29.1        0    yes northwest 29141.36
```

```
str(insurance)
```

```
## 'data.frame':    1338 obs. of  7 variables:
##  $ age     : int  19 18 28 33 32 31 46 37 37 60 ...
##  $ sex     : Factor w/ 2 levels "female","male": 1 2 2 2 2 1 1 1 2 1 ...
##  $ bmi     : num  27.9 33.8 33 22.7 28.9 25.7 33.4 27.7 29.8 25.8 ...
##  $ children: int  0 1 3 0 0 0 1 3 2 0 ...
##  $ smoker  : Factor w/ 2 levels "no","yes": 2 1 1 1 1 1 1 1 1 1 ...
##  $ region  : Factor w/ 4 levels "northeast","northwest",..: 4 3 3 2 2 3 3 2 1 2 ...
##  $ expenses: num  16885 1726 4449 21984 3867 ...
```

We can clearly now differenciate between 2 types of variables and analyze their relationships :

### Dependant variable :

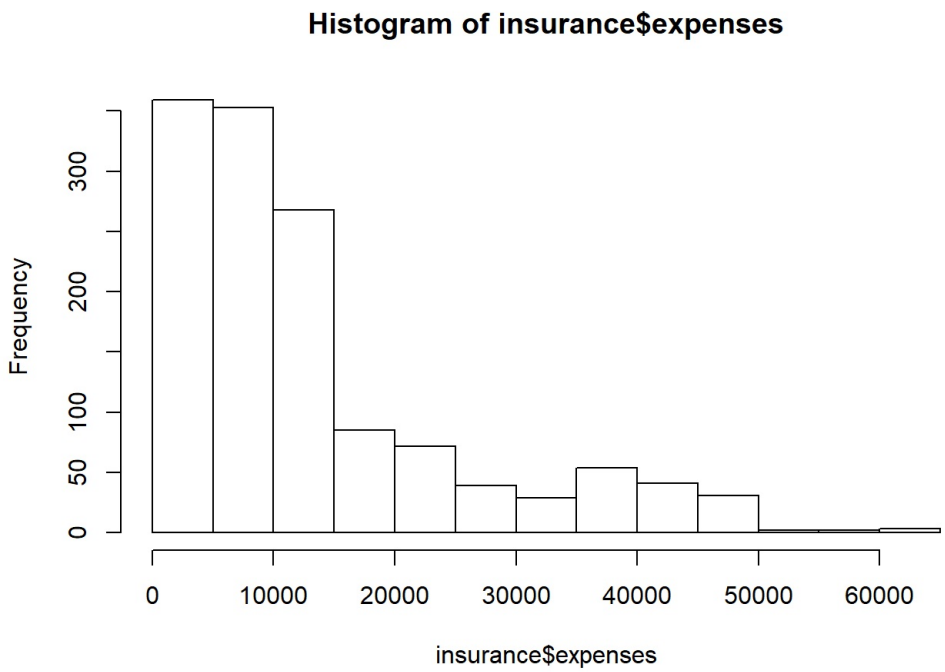
Expenses : numeric, indicates individual medical cost billed by health insurance.

```
summary(insurance$expenses)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1122    4740    9382   13270   16640   63770
```

We can see that the distribution of expenses is highly right skewed as the mean value is greater that the median. And we can confirm that visually in this histogram :

```
hist(insurance$expenses)
```




### Independant variables :

- Age : integer, indicates the age of the beneficiary.

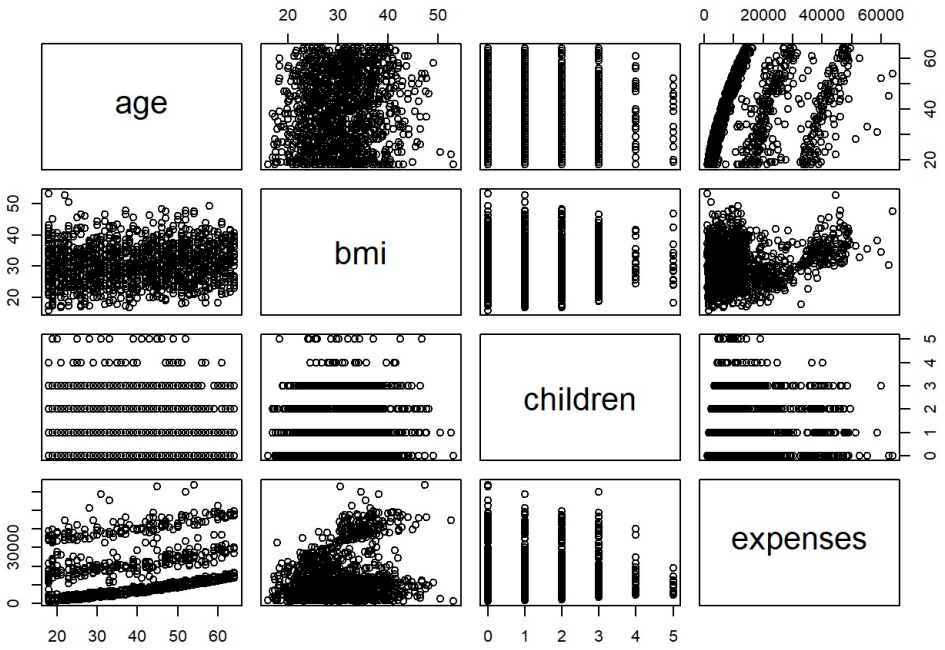
- Sex : factor, indicates the gender of the beneficiary.
- Bmi : numeric, indicates the body mass index.
- Children : integer, indicates the number of children in this family covered by health insurance.
- Smoker : factor, indicates wether or not the beneficiary smoke.
- Region : factor, indicates the beneficiary's residential area.

### Correlation and Scatterplot matrix :

We want now to have a look at the relationship between variables and use tools that provides a visualization of the correlation of variables :

***- The scatterplot matrix :***

```
pairs(insurance[c("age", "bmi", "children", "expenses")])
```




***- The correlation matrix :***

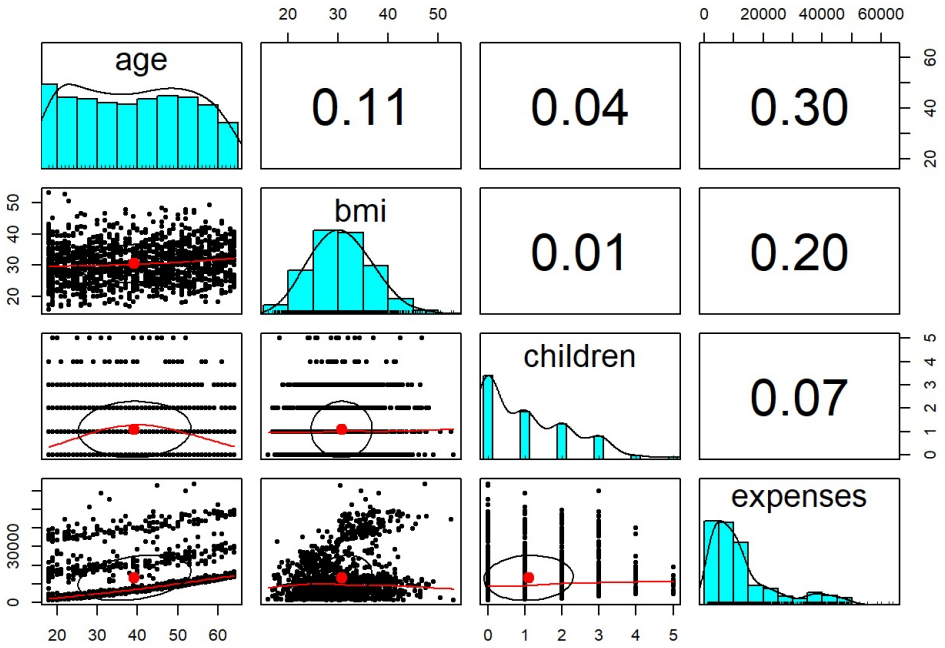
```
round(cor(insurance[c("age", "bmi", "children", "expenses")]),2)
```

```
##             age  bmi children expenses
## age        1.00 0.11     0.04     0.30
## bmi        0.11 1.00     0.01     0.20
## children   0.04 0.01     1.00     0.07
## expenses   0.30 0.20     0.07     1.00
```

Having a more neater look at these data :

***- Panel of correlation matrix :***

```
 pairs.panels(insurance[c("age", "bmi", "children", "expenses")])
```




*Methodology :*

According to the steps of multiple linear regression, we first look at the correlation matrix that we presented before, then we fit the simple linear regression using every variable.

From the R^2 of every simple linear regression model, we can analyze and see which variable have relatively high R^2 value.

Next, we consider mutiple linear regression model. We first put all of the variables into the multiple regression model, and then, improve the model by introducing new variables or interaction term.

***Building a model :***

```
# We are going to train the model on the training dataset,
# and predict the values on the test dataset :

# Split data into training(70%) and testing(30%) sets
set.seed(1313)
n <- length(insurance$expenses)
index <- sample(1:n,floor(n*0.7))
length(index)
```

```
## [1] 936
```

```
train <- insurance[index,]
test <- insurance[-index,]
mod1 <- lm(expenses~.,data = train)
```

We can now have a look at informations related to our training model :

```
summary(mod1)
```

```
##
## Call:
## lm(formula = expenses ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12015.0  -2796.9   -905.8   1392.1  29137.7
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -12034.07    1186.84 -10.140   <2e-16 ***
## age                265.99      14.26  18.657   <2e-16 ***
## sexmale            106.85     401.88   0.266   0.7904
## bmi                326.92      34.36   9.513   <2e-16 ***
## children           373.98     167.37   2.234   0.0257 *
## smokeryes        24333.93     494.98  49.161   <2e-16 ***
## regionnorthwest   -439.44     572.17  -0.768   0.4427
## regionsoutheast   -859.59     566.24  -1.518   0.1293
## regionsouthwest   -988.32     564.74  -1.750   0.0804 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6112 on 927 degrees of freedom
## Multiple R-squared:  0.7577, Adjusted R-squared:  0.7556
## F-statistic: 362.3 on 8 and 927 DF,  p-value: < 2.2e-16
```

The Pr(>|t|) value can indicate which variable is relevant or not, as we will be using a significance level of α = 0.05.

We can see that mostly all variables are relevant, except the sex variable, because it nearly has no importance to expenses.

```
AIC(mod1)
```

```
## [1] 18987.35
```

We will now try to improve the model by removing the sex variable as we have found our AIC value that measures the quality of our model. The lower the AIC, the better the model is.

```
# Improving the model.
mod2 <- lm(expenses~ age +bmi+children+smoker+region, data = train)
AIC(mod2)
```

```
## [1] 18985.42
```

We can clearly see that the AIC is lower even if it has not reduced by a lot.

### *Testing the model :*

```
deviance(mod2)
```

```
## [1] 34633210823
```

We will now use the ANOVA function which is a function that makes an analysis of variance table and returns reisduals of the same degree as the deviance function. We can now do a hypothesis test to test if our model is good enough :

```
anova(mod2, test = "Chisq")
```

```
## Analysis of Variance Table
##
## Response: expenses
##            Df     Sum Sq    Mean Sq   F value    Pr(>F)
## age         1 1.3237e+10 1.3237e+10  354.6978 < 2.2e-16 ***
## bmi         1 3.1102e+09 3.1102e+09   83.3372 < 2.2e-16 ***
## children    1 5.5328e+08 5.5328e+08   14.8251  0.000126 ***
## smoker      1 9.1226e+10 9.1226e+10 2444.4197 < 2.2e-16 ***
## region      3 1.3933e+08 4.6443e+07    1.2445  0.292386
## Residuals 928 3.4633e+10 3.7320e+07
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- H0 : deviance nearly approaching the Chi-Squared value. => This model is good enough.
- H1 : No relationship between both deviance and Chi-Squared value.

We can also notice that the Pr(>F), The Fisher Test, value show that all the variables are below the significance level of α = 0.05.

*Conclusion :* **We fail to reject H0. We can then say that the model is good.**

We will now check the multicollinearity :

```
library(car)
```

```
## Error: package or namespace load failed for 'car' in loadNamespace(i, c(lib.loc, .libPaths()), versionCheck = v
I[[i]]):
##  namespace 'rlang' 0.4.4 is already loaded, but >= 0.4.5 is required
```

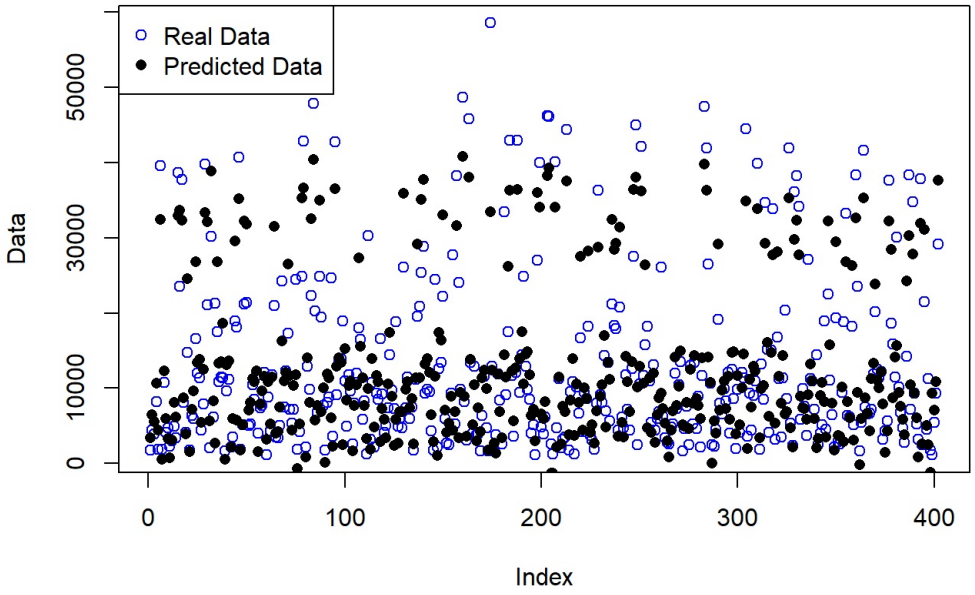
```
vif(mod2)
```

```
## Error in vif(mod2): impossible de trouver la fonction "vif"
```

We can see that we do not have a VIF value larger than 10. This indicates no multicollinearity.

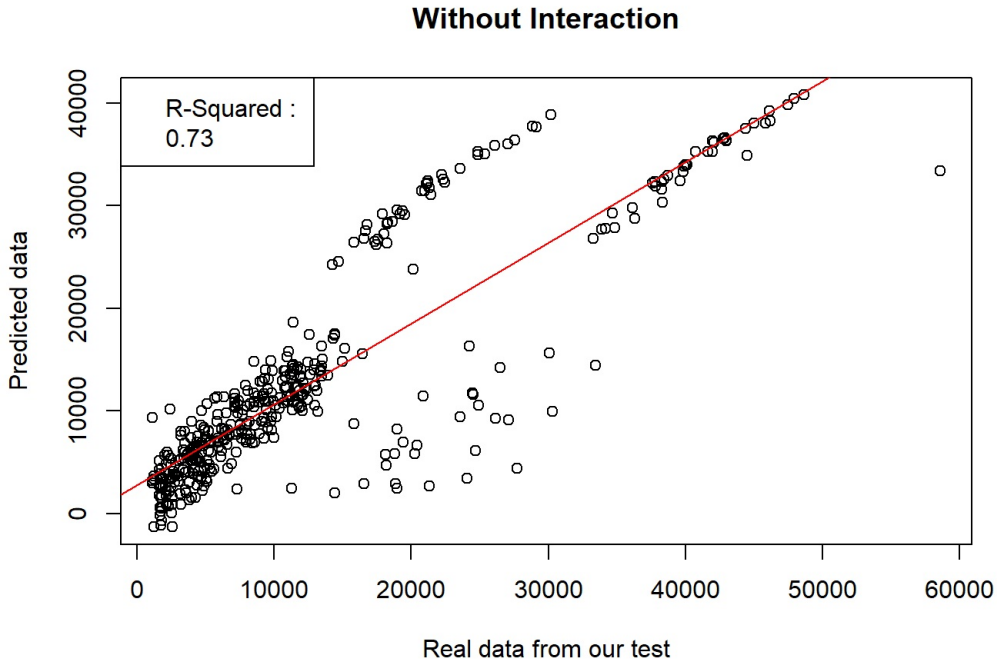
We will now plot the predicted values angainst the real values to see how our model is performing :

```
coeffs <- mod2$coefficients

predicted.data <- predict(mod2, newdata = test)
real.data <- test$expenses

{plot(real.data,ylab = 'Data', col = 'blue')
points(predicted.data, col = 'black', pch = 16)
legend('topleft',legend = c('Real Data','Predicted Data'), col = c('blue','black'), bty = n,pch = c(1,16))}
```




We can see that the predicted values are almost close to the real values. Let's see and plot our R-squared value :

```
mod.interactionFalse <- lm(predicted.data ~ real.data)

{plot(real.data, predicted.data, xlab = 'Real data from our test', ylab ='Predicted data', title('Without Interact
ion'))
abline(a = mod.interactionFalse$coefficients[1], b = mod.interactionFalse$coefficients[2],col = 'red')
legend('topleft',legend = c('R-Squared : ',round(summary(mod.interactionFalse)$r.squared,2)))}
```




We find 75% of accuracy without interactions, which is a good value but we can improve it.

***Improving our model performance :***

We will now improve our model performance by going through some logical improvements :

1- Adding non-linear relationships :

In linear regression, the relationship between an independant variable and the dependent variable is assumed to be linear, yet this may not necesarily be true. For example, the effect of age on medical expenditures may not be constant throughout all age values; the treatment may become disproportionately expensive for the oldest populations.

```
train$age2 <- train$age^2
test$age2 <- test$age^2
```

2 - Converting a numeric variable to a binary indicator :

Logically, medical expenses will only be affected by bmi if it is abnormal i.e. >30, this will make our model more efficient.

```
train$bmi30 <- ifelse(train$bmi <= 30, 1, 0)
test$bmi30 <- ifelse(test$bmi <= 30, 1, 0)
```

3 - Adding Interactions :

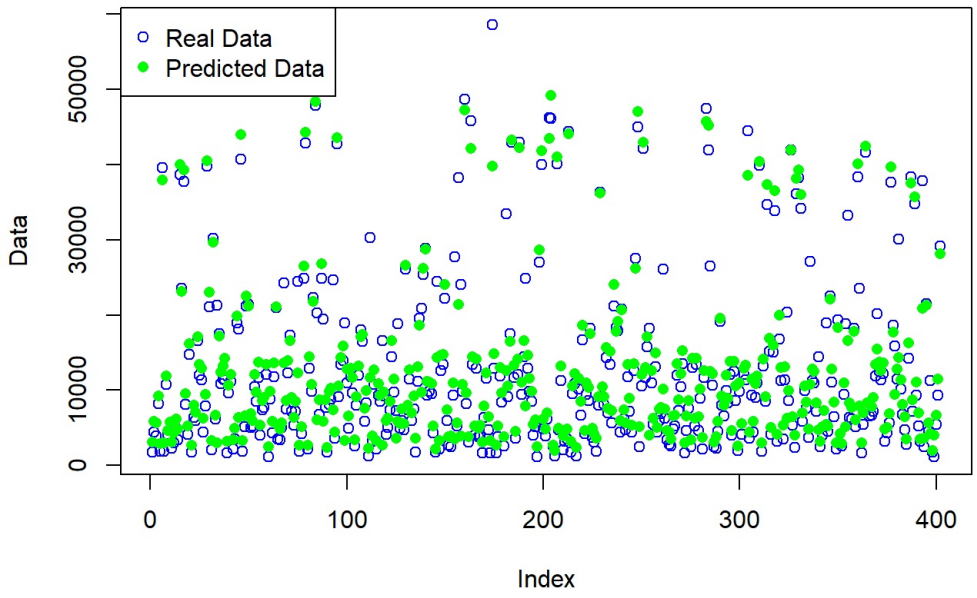
We can say that smoking and obesity may have harmful effects separately, but it is reasonable to assume that their combined effect may be worse than the sum of each one alone.

We finally now have this improved model :

```
finalmod <- lm(expenses ~ age + age2 + children + bmi + smoker + bmi30*smoker + region,data = train)
```

Let's now see how is this new model in predicting the real values :

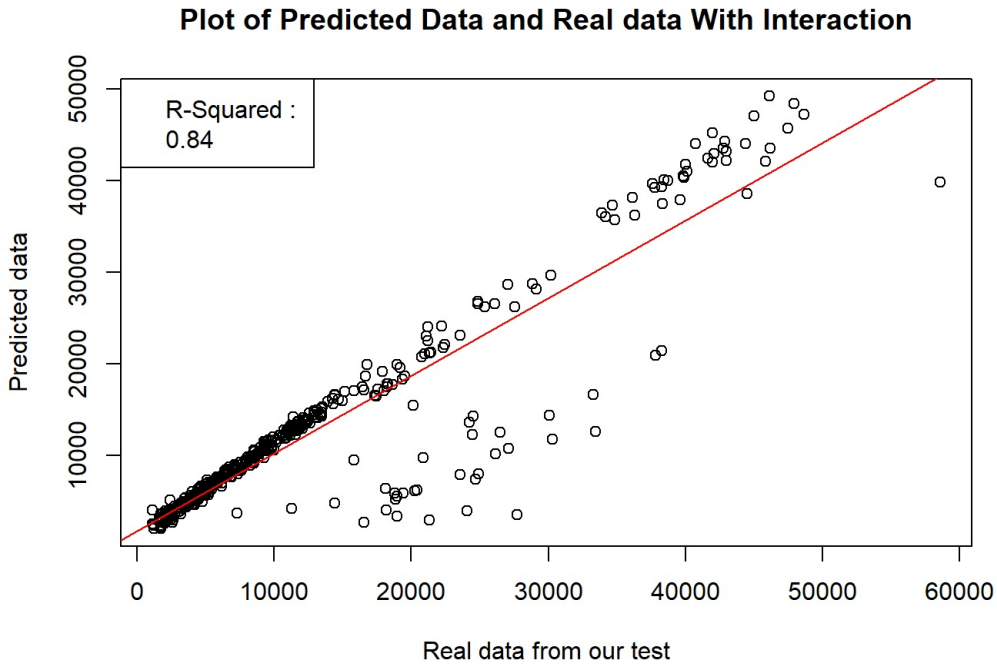
```
coeffs <- finalmod$coefficients

predicted.data.inter <- predict(finalmod, newdata = test)
real.data <- test$expenses

{plot(real.data,ylab = 'Data', col = 'blue')
points(predicted.data.inter, col = 'green', pch = 16)
legend('topleft',legend = c('Real Data','Predicted Data'), col = c('blue','green'), bty = n,pch = c(1,16))}
```




We can see that there's a huge improvement in predicting the real values.

Let's now have a look at our R-Squared plot to find the accuracy of our model :

```
mod.interactionTrue <- lm(predicted.data.inter ~ real.data)

{plot(real.data, predicted.data.inter, xlab = 'Real data from our test', ylab ='Predicted data', title('Plot of Pr
edicted Data and Real data With Interaction'))
abline(a = mod.interactionTrue$coefficients[1], b = mod.interactionTrue$coefficients[2],col = 'red')
legend('topleft',legend = c('R-Squared : ',round(summary(mod.interactionTrue)$r.squared,2)))}
```




Relative to our first model, the R-squared value has improved from 0.73 to about 0.84. Our model is now explaining 86 percent of the variation in medical treatment costs.

***Test Prediction :***

We will now try to predict the expected medical expenses for a new customer with these specifications :

A man aged 30 with 2 children, who is a smoker with a BMI of 32 and who live in the southeast of Ireland.

```
###
prediction.custom <- function(age,age2,regionNW,regionSE,regionSW,children,smoker,bmi){
  result <- age*coeffs[2] + age2*coeffs[3] + regionNW*coeffs[4] + regionSE*coeffs[5] + regionSW*coeffs[6] + childr
en*coeffs[7] + smoker*coeffs[8] + (1-smoker)*bmi*coeffs[9]+ smoker*bmi*coeffs[10] + coeffs[1]

  return(result)
}

result <- -(prediction.custom(30,900,0,1,0,2,1,32))
result
```

```
##      age
## 31275.11
```

```
###
```

So this man, will have to pay ~31000 for his medical expenses.

## Results :

1- From the correlation matrix that we've seen, we can see that although the correlation coefficient in the matrix does not reflect strong relationships between these variables, some correlations do exist. For example, age and bmi has moderate relationship, which means that with the increase of age, body mass index will also increase. Besides of that, age and charges all have moderate correlation.

2- When we put all variables with our first model we obtain a 73% R-Squared value for our multiple linear regression model but we can clearly see that we can improve our model as high values are hardly being accurately predicted.

3- We have no multicollinearity, and we did a hypothesis test to check if our model was good using the deviance value.

4- When we put the variables in simple linear regression model individually, we can see that bmi and smoker can have an interaction to improve the quality of our model.

## Conclusion :

For the linear regression model, our final model includes variables of age,age2, bmi, children, region, smoker and bmi30*smoker with R2 = 0.84 that was improved from our initial R2 = 0.75. We applied 3 tests after building the linear regression model and found that although there is no multicollinearity problem, the residuals do not have constant variance or normal distribution. Finally, we got R2 = 0.84, which performs slightly better than our first linear regression model. At the same time, by Variable Importance provided by our first test, **we can clearly conclude that smokers (smoke or not), BMI and age are the main variables that influence the medical expenses of an individual person most** and these are the variables that are been taken in count to predict our expenses along with our final model. Region, gender and children (the number of children covered in the insurance) show little influence on the individual medical charges. The results of descriptive statistical analysis.