SAE Régression

Fulbert_Chen_GRP15

15/05/2022

Ajustement linéaire

Initialisation

```
library(mangoTraining)

## Warning: le package 'mangoTraining' a été compilé avec la version R 4.1.3

View(auto_mpg)
which(is.na(auto_mpg$horsepower))

## [1]  33 127 331 337 355 375

x = auto_mpg$mpg
y = auto_mpg$acceleration
```

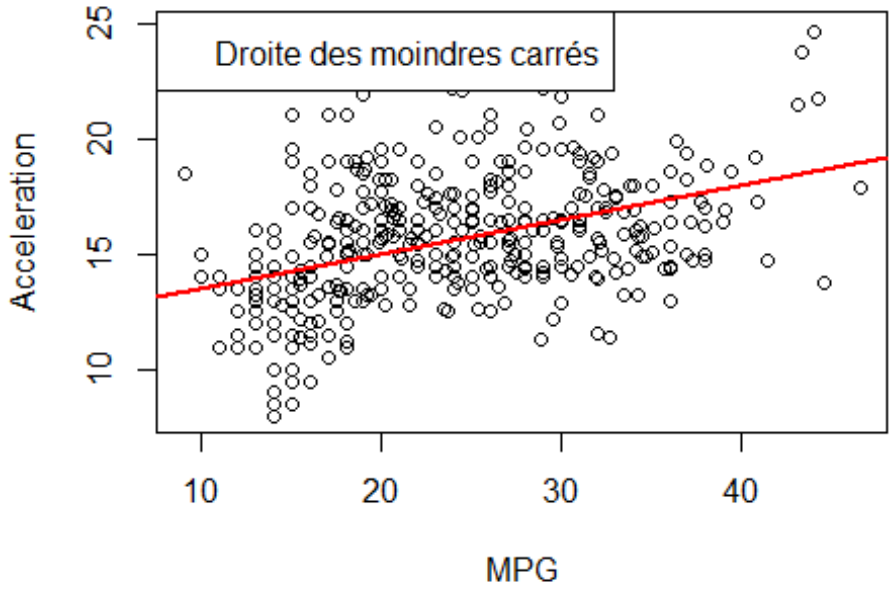
Corrélation

```
cor(x,y)

## [1] 0.4202889

(cor(x,y))^2

## [1] 0.1766428
```

Ici, on obtient 17,66% de la variance de l'accélération est expliqué par l'ajustement linéaire en fonction du mpg

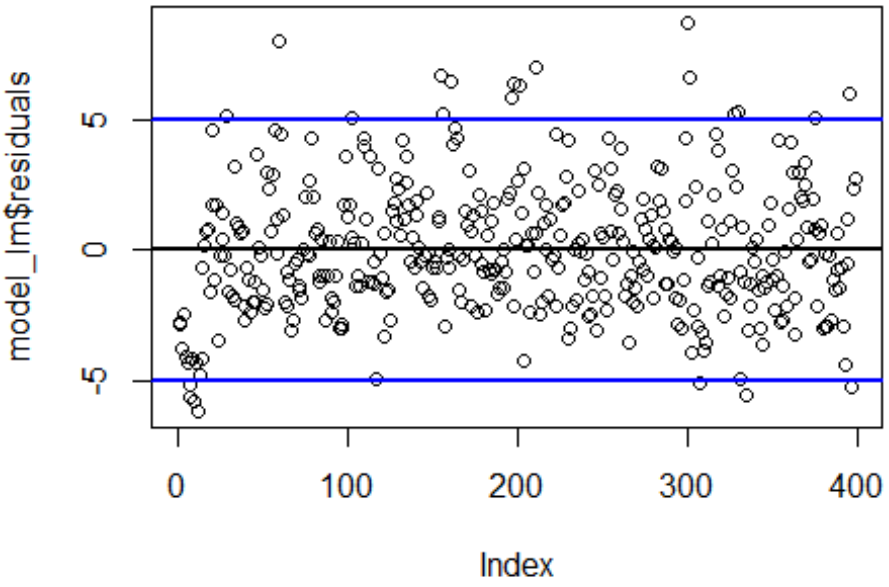
Graphique

```
model_lm = lm(y~x)
plot(auto_mpg$mpg,auto_mpg$acceleration,xlab = "MPG",ylab="Acceleration")
abline(model_lm,col="red",lwd=2)
legend ("topleft",legend=c("Droite des moindres carrés"))
```



Résidus

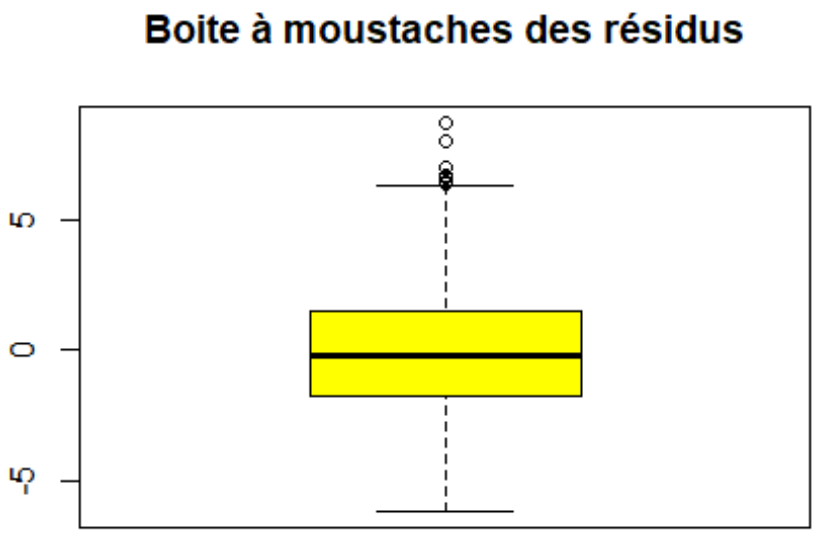
```
plot(model_lm$residuals)
SD = function(x)
{
  SD = sqrt(var(x)*(length(x)-1)/length(x))
            return(SD)
}
SD(model_lm$residuals)

## [1] 2.499155

#Résultat de l'écart type
abline(0,0,lwd=2)
abline(2*SD(model_lm$residuals),0,lwd= 2,col="blue")
abline(-2*SD(model_lm$residuals),0,lwd=2,col="blue")
```



Ajustement polynomial de degré 2

Boîte à moustache des résidus

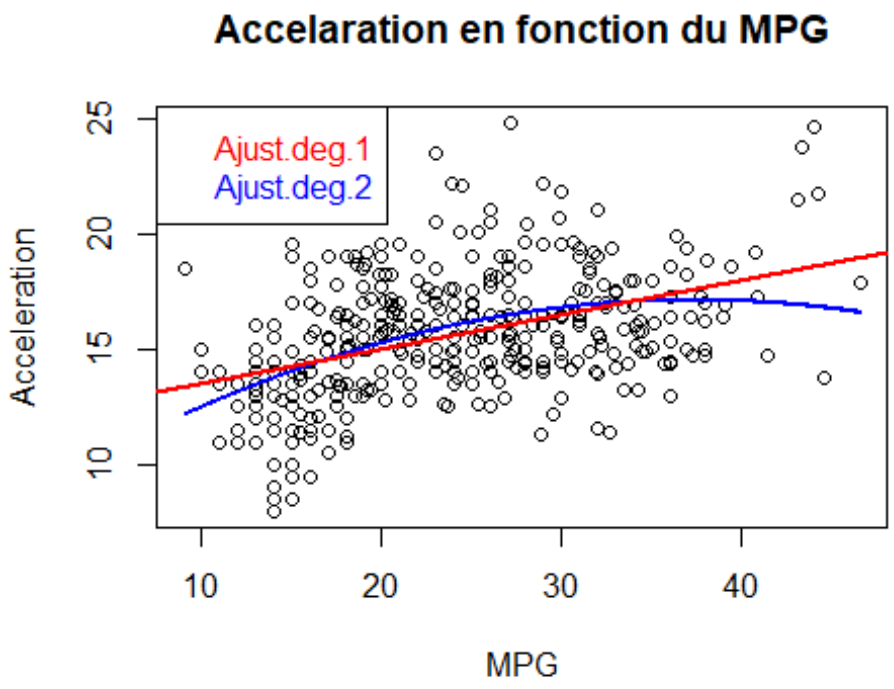
```
boxplot(model_lm$residuals,col="yellow",main="Boite à moustaches des résidus"
)
```



Graphique

```
reg2 = lm(y~x+I(x^2))
summary(reg2)

##
## Call:
## lm(formula = y ~ x + I(x^2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.7669 -1.7669 -0.2669  1.4683  8.3133
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.538151   1.109482   7.696 1.14e-13 ***
## x            0.459656   0.092543   4.967 1.01e-06 ***
## I(x^2)      -0.006155   0.001802  -3.415 0.000703 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.472 on 395 degrees of freedom
## Multiple R-squared:  0.2003, Adjusted R-squared:  0.1962
## F-statistic: 49.45 on 2 and 395 DF,  p-value: < 2.2e-16

xseq = seq(min(auto_mpg$mpg),max(auto_mpg$mpg),by = 0.01)
coef_p2 = reg2$coefficients
plot(x,y,main="Accelaration en fonction du MPG",xlab="MPG",ylab="Acceleration
```

```
")
lines(xseq,coef_p2[1]+coef_p2[2]*xseq+coef_p2[3]*xseq^2,col="blue",lwd=2)
abline(model_lm,col="red",lwd=2)
legend("topleft",legend=c("Ajust.deg.1","Ajust.deg.2"),
       text.col=c("red","blue"))
```



r-squared_2 = 0.2003 Ici, on obtient 20,03% de la variance de l'accélération est expliqué par l'ajustement polynomial de degrée 2 en fonction du mpg

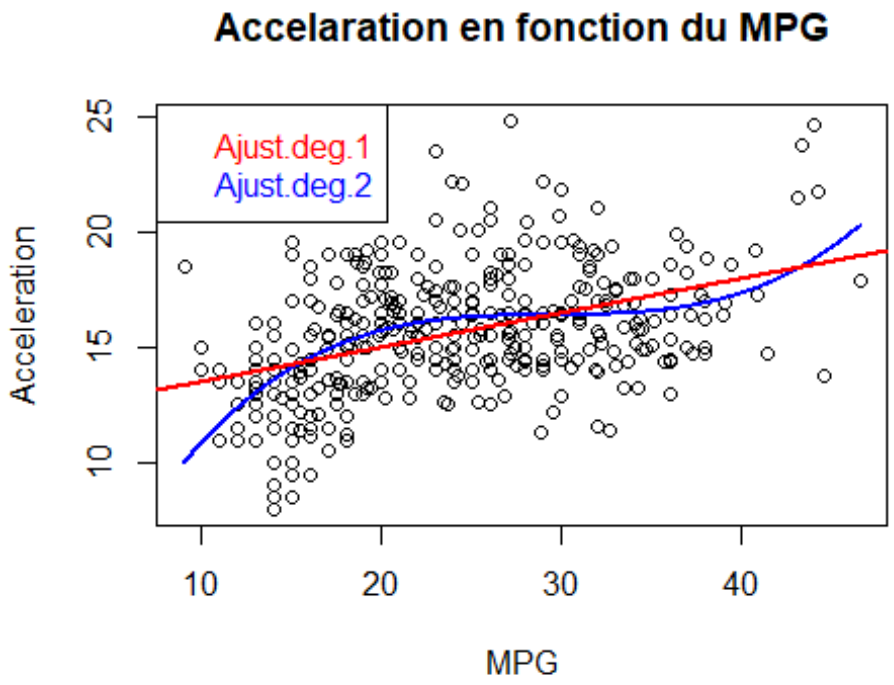
Ajustement polynomial de degré 3

Graphique

```
reg3 = lm(y~x+I(x^2)+I(x^3))
summary(reg3)

##
## Call:
## lm(formula = y ~ x + I(x^2) + I(x^3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.5980 -1.6520 -0.1766  1.3620  8.5078
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.5780312  2.9198520  -0.883    0.378
## x            1.9243287  0.3683714   5.224 2.85e-07 ***
```

```
## I(x^2)      -0.0652940  0.0145232  -4.496 9.13e-06 ***
## I(x^3)       0.0007409  0.0001806   4.102 4.97e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.424 on 394 degrees of freedom
## Multiple R-squared:  0.233,  Adjusted R-squared:  0.2272
## F-statistic:  39.9 on 3 and 394 DF,  p-value: < 2.2e-16
```

```
xseq = seq(min(auto_mpg$mpg),max(auto_mpg$mpg),by = 0.01)
coef3 = reg3$coefficients
plot(x,y,main="Accelaration en fonction du MPG",xlab="MPG",ylab="Acceleration
")
lines(xseq,coef3[1]+coef3[2]*xseq+coef3[3]*xseq^2+coef3[4]*xseq^3,col="blue",
lwd=2)
abline(model_lm,col="red",lwd=2)
legend("topleft",legend=c("Ajust.deg.1","Ajust.deg.2"),
       text.col=c("red","blue"))
```



r-squared_3 = 0.233

Ici, on obtient 23.3% de la variance de l'accélération est expliqué par l'ajustement polynomial de degrée 3 en fonction du mpg