MTH2302B - H2017

TD11 - Exercice 4

Observations

In [1]:

```
# Données
X1=c(30,30,30,30,50,50,50,50,70,70,70,70,90,90,90,90)
Y=c(66.83,72.77,68.04,66.11,84.45,83.06,77.92,84.00,93.94,86.71,92.38,87.60,86.27,90.19,88.97,84.95)

# Tracer le nuage de points
plot(X1,Y)
```

...

In [2]:

```
# Nombre de mesures
n = length(X1)

# Moyennes échantillonnales
x_bar = mean(X1)
y_bar = mean(Y)

# Somme des carrés corrigée
S_xx = sum((X1-x_bar)^2)
S_yy = sum((Y-y_bar)^2)
# Somme des produits croisés corrigée
S_xy = sum( (X1-x_bar) * (Y-y_bar))
```

a) Régression linéaire simple

In [3]:

```
# Régression linéaire avec lm
linReg = lm(Y~X1)
summary(linReg)

beta_hat = unname(coefficients(linReg))
S_beta = unname(summary(linReg)$coefficients[,2])
```

...

In [4]:

```
anova(linReg)
```

...

b) Analyse des résidus

In [5]:

```
par(mfrow=c(2,2))
Y_hat= fitted.values(linReg)

# Calcul des résidus
res=residuals(linReg)

plot(linReg)

# Test de normalité
shapiro.test(res)

# Conclusion ?
```

...

1.c) Test de signification du modèle

On teste $H_0 : \beta_1 = 0$ contre $H_1 : \beta_1 \neq 0$

In [6]:

```
summary(linReg)
```

...

2.c) Intervalle de confiance pour la pente de la droite de régression

Pour un niveau de confiance $1 - \alpha$, l'intervalle de confiance pour β_1 est donné par :

$$\beta_1 \in \hat{\beta}_1 \pm t_{\alpha/2,n-2} \sqrt{\frac{MSE}{S_{XX}}}$$

In [7]:

```
# Intervalle de confiance à 95% pour beta1
confint(linReg,parm='X1',level = 0.95)
```

...

d) Intervalle de prévision

Pour un niveau de confiance $1 - \alpha$, l'intervalle de prévision pour $Y|x = x_0$ est donné par :

$$Y_0 \in \hat{y}_0 \pm t_{\alpha/2,n-2} \sqrt{MS_E \left(1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{S_{XX}}\right)}$$

In [8]:

```
# Intervalle de prévision à 95% pour y en x=60
alpha= 0.05
x0 = 60
y0_hat = beta_hat[1] + beta_hat[2] * x0

MSE = sum(residuals(linReg)^2) / (n-2)
l =y0_hat - qt(alpha/2,n-2,lower.tail=FALSE) * sqrt(MSE * (1+1/n + (x0 - x_bar)^2 / S_xx))
u =y0_hat + qt(alpha/2,n-2,lower.tail=FALSE) * sqrt(MSE * (1+1/n + (x0 - x_bar)^2 / S_xx))
cat(l, u)
```

70.5507 93.72305

In [9]:

```
# Tracer les intervalles de prévision à 95%
alpha = 0.05

x_prev = seq(min(X1),max(X1),by=(max(X1)-min(X1))/20)
y_prev = beta_hat[1] + beta_hat[2] * x_prev

MSE = sum(residuals(linReg)^2) / (n-2)
l_prev =y_prev - qt(alpha/2,n-2,lower.tail=FALSE) * sqrt(MSE * (1+1/n + (x_prev - x_bar)^2 / S_xx))
u_prev =y_prev + qt(alpha/2,n-2,lower.tail=FALSE) * sqrt(MSE * (1+1/n + (x_prev - x_bar)^2 / S_xx))

plot(X1,Y)
lines(X1,fitted.values(linReg),col='blue')
lines(x_prev,l_prev,lty=2,col='red')
lines(x_prev,u_prev,lty=2,col='red')

remove(x_prev,y_prev,l_prev,u_prev)
```

...

e) Régression polynomiale du second degré

1. Mise en forme des données

In [10]:

```
X0 = rep(1,n)
X2 = X1^2
k=2

# Formation de la matrice X
X = matrix(c(X0,X1,X2), nrow = n,ncol = 3)
C = solve( t(X) %*% X)
```

2. Ajustement du modèle de régression

In [11]:

```
# Régression avec lm
qReg = lm(Y ~ X1+X2)
summary(qReg)

# Coefficients de régression et leur écart-type echantillonal
beta_hat = unname(coefficients(qReg))
S_beta = unname(summary(qReg)$coefficients[,2])

cat('beta_hat =', beta_hat, '\n')
cat('S_beta =', S_beta)
```

...

3. Courbe de régression

In [12]:

```
# valeurs de la régression
Y_hat = fitted.values(qReg)

# Courbe de regression
plot(X1,Y)

x_reg = seq(min(X1),max(X1),by = 3)
y_reg = beta_hat[1] + beta_hat[2] * x_reg + beta_hat[3] * x_reg^2
lines(x_reg,y_reg,col='blue')

remove(x_reg)
remove(y_reg)
```

...

4. Table d'analyse de la variance

In [13]:

```
anova(qReg)
```

...

f) Test de signification global

On teste $H_0 : \beta_1 = \beta_2 = 0$ contre H_1 : au moins un des $\beta_j \neq 0$

In [14]:

```
summary(qReg)
```

...

g) Comparaison des modèles

In [15]:

```
summary(linReg)
summary(qReg)
```

...

h) Rendement optimal

In [16]:

```
beta = unname(coefficients(qReg))

x_opt = -beta[2] / (2*beta[3])
x_opt
```

75.8404368269376