TD11_ex3

November 21, 2018

1 MTH2302B - H2017

2 TD11 - Exercice 3

2.1 a) Observations

In [ ]:
```
# Données
X=c(14,18,40,43,45,112)
Y=c(280,350,470,500,560,1200)

# Tracer le nuage de points
plot(X,Y) #Est-ce qu'un modèle linéaire est plausible ?
```

In [ ]:
```
# Nombre de mesures
n = length(X)

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

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

2.2 b) Régression linéaire simple

Modèle de régression : $Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$

2.2.1 0. Régression linéaire en R avec lm

In [ ]:
```
# Ajuster un modèle de régression avec la fonction lm
linModel <- lm(Y~ X)

# afficher les resultats numeriques
summary(linModel)
```

In [ ]:
```
anova(linModel)
```

2.2.2 1. Coefficients de régression

$\hat{\beta}_1 = \frac{S_{XY}}{S_{XX}}$
$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X}$

In [ ]:
```
# Calcul des coefficients de régression
beta1_hat = (S_xy / S_xx)
beta0_hat= y_bar - beta1_hat* x_bar

cat('beta0_hat =', beta0_hat, '\n')
cat('beta1_hat =', beta1_hat, '\n')
```

In [ ]:
```
# Tracer la droite de régression
Y_hat = beta0_hat + beta1_hat * X

plot(X,Y, col = 'black') #données expérimentales (points noirs)
lines(X,Y_hat,col='blue') #droite des moindres carrés (en bleu)
```

In [ ]:
```
# Estimation de sigma^2

SST = S_yy
#SSE = sum( (Y-Y_hat)^2) #method 1
SSR = sum( (Y_hat-y_bar)^2)
SSE = SST - SSR #method 2

MSR = SSR / 1
MSE = SSE / (n-2)

# Estimateur de la variance des résidus
sigma_hat = sqrt(MSE)
cat('sigma_hat =', sigma_hat)
```

In [ ]:
```
# Variance des coefficients de regression
S_beta0 = sqrt(MSE * (1/n + x_bar^2 / S_xx))
S_beta1 = sqrt(MSE / S_xx)

cat('S(beta0_hat) =', S_beta0, '\n')
cat('S(beta1_hat) =', S_beta1, '\n')
```

In [ ]:
```
# Coefficient de détermination
R = sqrt(SSR / SST)
cat('R^2 =',R^2,'\n')
```

2.2.3 2. Table d'analyse de la variance

In [ ]:
```
cat('Table d\'analyse de la variance : \n')
cat('VAR \t| SS \t\t| df \t| MS \t\t| F0 \t\t| p-value \n')
cat('-----------------------------------------------------------------------------\n')
```

```
cat('Reg \t|', SSR, '\t|', 1, '\t|', MSR, '\t|', MSR / MSE, '\t|', pf(MSR/MSE,1,n-2,low
cat('Res \t|', SSE, '\t|', n-2, '\t|', MSE, '\t| NA \t\t| NA \n')
cat('Tot \t|', SST, '\t|', n-1, '\t| NA \t\t| NA \t\t| NA \n')
```

2.3 c) Intervalle de confiance pour la pente de la droite de régression

```
In [ ]: # Intervalle de confiance à 95% pour beta1
        alpha = 0.05

        l = beta1_hat - qt(alpha/2,n-2,lower.tail=FALSE) * S_beta1
        u = beta1_hat + qt(alpha/2,n-2,lower.tail=FALSE) * S_beta1
        cat(l,u)
```

2.4 d) Test de signification

2.4.1 1. Test de Student

On teste $H_0 : \beta_1 = \beta_{1,0}$ contre $H_1 : \beta_1 \neq \beta_{1,0}$ avec ici : $\beta_{1,0} = 0$

```
In [ ]: # Paramètre du test
        beta1_0 = 0
        alpha = 0.05

        # Statistique de test
        t_0 = (beta1_hat - beta1_0) / (sqrt(MSE / S_xx)) #suit une loi t(n-2)

        # Région critique
        U = qt(alpha/2,n-2,lower.tail = FALSE)
        cat('t0 =',t_0, '\nU = ', U,'\n')

        # Conclusion ?

        # Calcul de la p-value
        p = 2* pt(t_0,n-2,lower.tail = FALSE)
        cat('p-value =', p,'\n')
```

2.4.2 2. Test de Fisher

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

```
In [ ]: # Paramètre du test
        alpha = 0.05

        # Statistique de test
        f_0 = MSR / MSE #suit une loi F(1,n-2)

        # Région critique
        U = qf(alpha,1,n-2,lower.tail = FALSE)
        cat('f0 =',f_0, '\n U = ', U,'\n')
```