

```

m = read.table("...../11623991.txt", header=TRUE)
# Sans "header=TRUE", les noms des variables sont "V1 V2 V3 V4 V5"
# La 1ere ligne des donnees sont des caracteres
### correction
# Q1
mod1 = lm(log(Y)~log(X), data = m); (su1 = summary(mod1))
N = m$X^2
mod2 = lm(Y~N, data = m); (su2 = summary(mod2))
mod3 = lm(Y~X, data = m); (su3 = summary(mod3))
par(mfrow = c(2,2))
plot(mod1)
par(mfrow = c(2,2))
plot(mod2)
par(mfrow = c(2,2))
plot(mod3)
### comparaison des graphiques des esperances des residus
plot(mod1, which = 1); plot(mod2, which = 1); plot(mod3, which = 1)
# conclusion : La courbure de mod3 est la plus forte
# Donc on elimine mod3
### comparaison des graphiques des variances des residus
par(mfrow = c(2,2))
plot(mod1, which = 3); plot(mod2, which = 3); plot(mod3, which = 3)
# conclusion : Les formes se ressemblent pour les 3 modeles.
### comparaison des graphiques des normalite des residus
par(mfrow = c(2,2))
plot(mod1, which = 2); plot(mod2, which = 2); plot(mod3, which = 2)
# conclusion : Les formes se ressemblent pour les 3 modeles.
X = m$X; Y = m$Y
R1 = 1-sum((Y-exp(mod1$fitted.values))^2)/sum((Y-mean(Y))^2) # pseudo-R2
# R2 de mod1
R2 = su2$r.squared # R2 de mod2
R3 = su3$r.squared # R2 de mod3
RR = c(su1$r.squared, su2$r.squared, su3$r.squared) # R2 des trois modeles
# 0.8431803 0.9525279 0.9253806
# mod2 > mod3 > mod1
# Il faut recalculer R2 du mod1. Parce que les SCT ne sont pas les memes
RRp = c(R1, R2, R3) # R2 comparable
# 0.8963818 0.9525279 0.9253806
odR = order(c(R1, R2, R3)) # R2 comparable ordonne croissant
# 1 3 2
# conclusion : odR[3] est le meilleur modele. C.a.d. mod2 est le meilleur
# Q2
ab1 = c(mod1$coefficients, exp(mod1$coefficients[1])) # (intercept, a_hat,
b_hat)
ab2 = c(mod2$coefficients, 0) # (intercept, a_hat, 0)
ab3 = c(mod3$coefficients, 0) # (intercept, a_hat, 0)
ah = c(ab1[2], ab2[2], ab3[2]) # a_hat des trois modeles
c(ab1[3], ab2[1], ab3[1])# b_hat des trois modeles
mod1;mod2;mod3
# Q3
summary(mod2)
#Coefficients:
#Estimate Std. Error t value Pr(>|t|)
#(Intercept) 10.46297 1.75692 5.955 6.54e-06 ***
# N 0.52105 0.02538 20.527 2.23e-15 ***
# a_chapeau = 0.52105, a_0 = 1
# Dans les sorties de "summary(mod2)", p-value "Pr(>|t|)<2.23e-15" est pour
# tester entre "a = 0" contre "a != 0"

```

```

# La question 3 demande de tester entre "a = 1" contre "a < 1"
t_chapeau = (0.52105-1)/0.02538
# Le degre de liberte est n-2 pour le modele simple (23-2)
pt(t_chapeau, 21) # 5.973466e-15 < 0.05 On rejette H0. La conclusion est a
< 1
# Q4
newdata=data.frame(X=16)
pre1 = predict(mod1,newdata,interval="prediction", data = m)
con1 = predict(mod1,newdata,interval="confidence", data = m)
newdata2=data.frame(N=16*16)
pre2 = predict(mod2,newdata2,interval="prediction")
con2 = predict(mod2,newdata2,interval="confidence")
pre3 = predict(mod3,newdata,interval="prediction")
con3 = predict(mod3,newdata,interval="confidence")
pre1;con1
pre2;con2
pre3;con3
# pre : intervalle de prediction ( intervalle de confiance pour Y)
# con : intervalle de confiance (intervalle de confiance pour E(Y))
# Pour mod1 les intervalles obtenus sont pour log(Y). Il faut appliquer exp
# sur les sorties pour obtenir les intervalles de Y

```