TP1_ST

Nathan_Edery

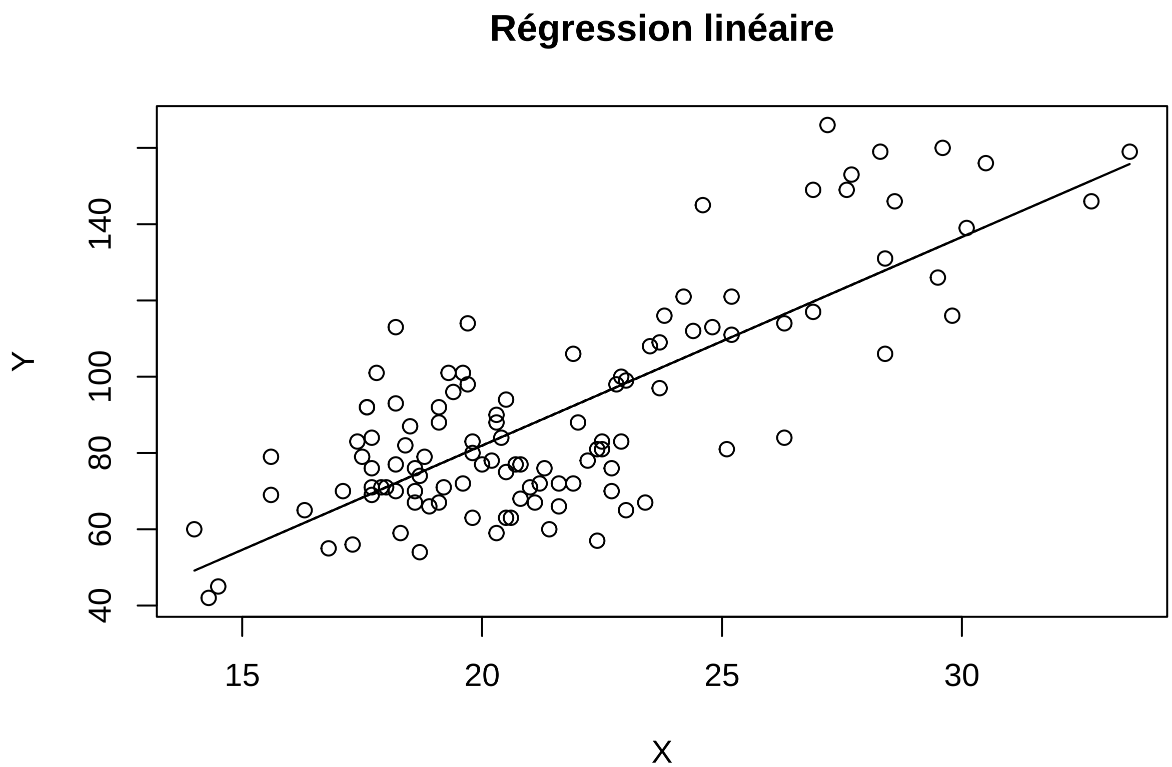
2024-01-26

TP1 - Série Temporelles

Exercice 1 :

1. Calculer une estimation des coefficients de pente et d'ordonnées à l'origine de la droite de régression en mettant en application les formules vues en cours. Tracer la droite de régression sur le nuage de points.

```
dataset=read.table('Dataset_ozone.txt', header=TRUE, sep=';', dec=',')

Y = dataset$maxO3
X = dataset$T12

Xn = mean(X)
Yn = mean(Y)

valup = 0
valdown = 0
for (i in 1:length(X)) {
  valup = valup + (Y[i]-Yn)*(X[i]-Xn)
  valdown = valdown + (X[i]-Xn)^2
}
beta1_chap = valup/valdown
beta0_chap = Yn-beta1_chap*Xn

reg = beta0_chap + beta1_chap*X
plot(X, Y)
lines(X, reg)
title("Régression linéaire")
```



2. Utiliser la fonction lm pour retrouver les mêmes résultats.

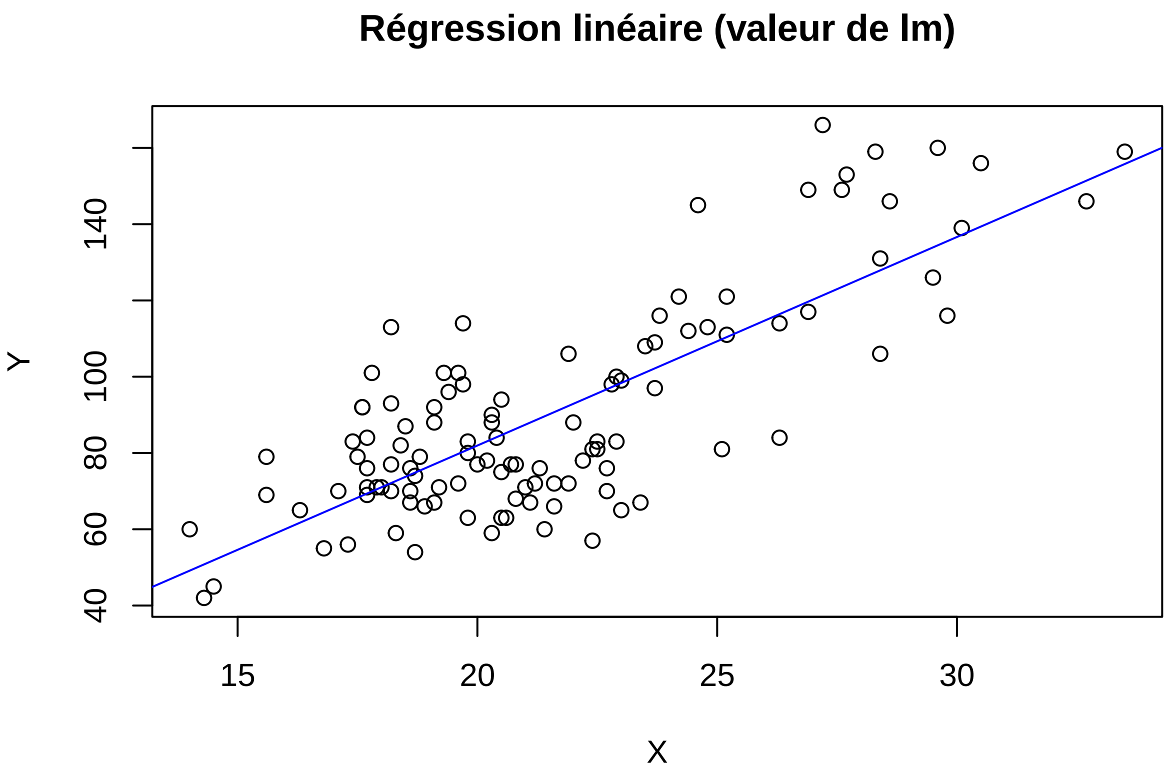
```
modele_reg = lm(Y ~ X)

beta0_lm = coef(modele_reg)[1]
beta1_lm = coef(modele_reg)[2]

print(paste("Coefficient d'ordonnée à l'origine (beta0) estimé par lm:", beta0_lm))
```

```
## [1] "Coefficient d'ordonnée à l'origine (beta0) estimé par lm: -27.4196358511813"
```

```
print(paste("Coefficient de pente (beta1) estimé par lm:", beta1_lm))
```

```
## [1] "Coefficient de pente (beta1) estimé par lm: 5.46868486741282"
```

```
plot(X, Y)
abline(modele_reg, col="blue")
title("Régression linéaire (valeur de lm)")
```



3. Evaluer la qualité du modèle en calculant vous-même l'indicateur. Puis retrouver ce résultat grâce à la commande lm.

```
valdown2=0
tmp1=0
tmp2=0
for (j in 1:length(X)) {
  tmp1 = tmp1 + (X[j]-Xn)^2
  tmp2 = tmp2 + (Y[j]-Yn)^2
}
valdown2 = tmp1*tmp2


R2 = valup^2/valdown2

R2_lm = summary(lm(Y~X))$r.squared

print(paste("Valeur de la qualité du modèle calculer avec la formule vu en cours : ", R2))
```

```
## [1] "Valeur de la qualité du modèle calculer avec la formule vu en cours :  0.615067401885113"
```

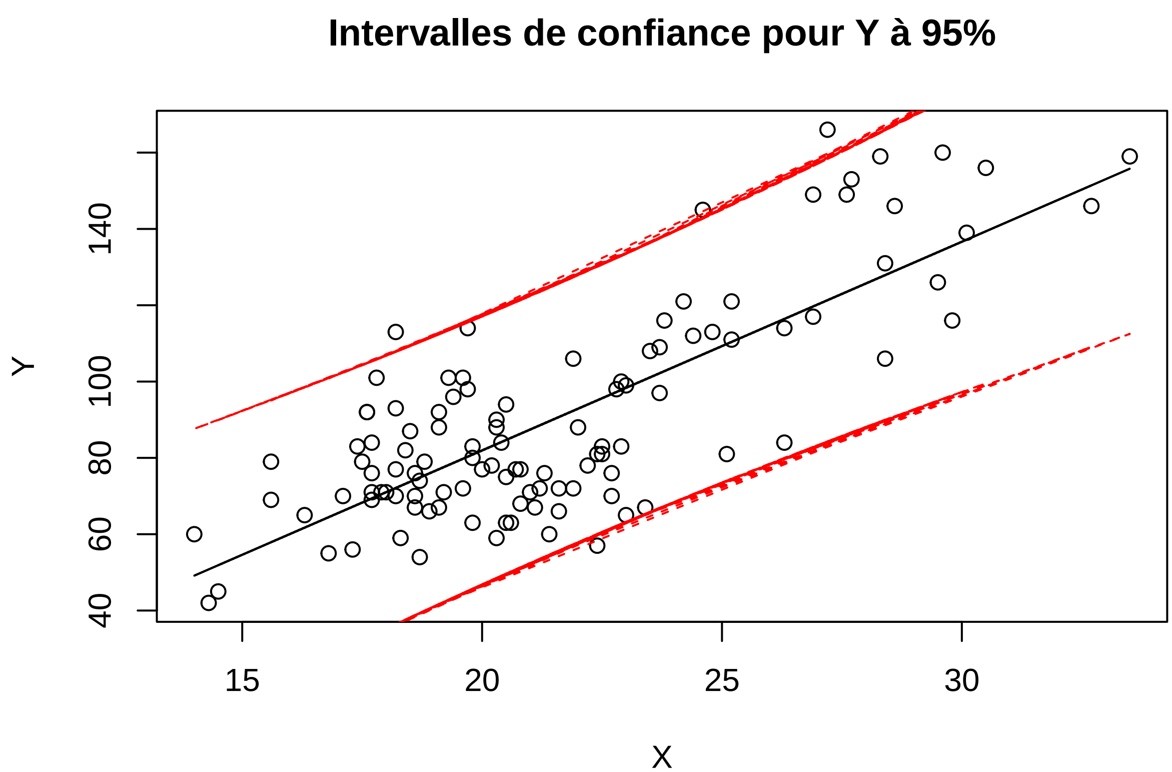
```
print(paste("Valeur de la qualité du modèle calculer avec lm : ", R2_lm))
```

```
## [1] "Valeur de la qualité du modèle calculer avec lm :  0.615067401885113"
```

4. Sur la représentation graphique précédente, tracer les intervalles de confiances pour y. Interprétez.

```
n = length(X)

tmp3 = 0
for (i in 1:n) {
  tmp3 = tmp3 + (Y[i]-reg[i])^2
```

```
}

sigma2 = (1/(n-2))*tmp3

alpha=0.05
t = qt((1-alpha/2), n-2)

f = t*(sigma2^0.5)*(1+1/n+((X-Xn)^2)/var(X)^2)^0.5

borne_sup = beta0_chap+beta1_chap*X + f
borne_inf = beta0_chap+beta1_chap*X - f

plot(X, Y)
lines(X, reg)
lines(X, borne_sup, col="red", lty=2)
lines(X, borne_inf, col="red", lty=2)
title("Intervalles de confiance pour Y à 95%")
```



5. Identifier les différentes sorties de la fonction lm

```
modele_reg = lm(Y ~ X, data = dataset)
summary(modele_reg)
```

```
##
## Call:
## lm(formula = Y ~ X, data = dataset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -38.079 -12.735   0.257  11.003  44.671
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -27.4196     9.0335  -3.035    0.003 **
## X             5.4687     0.4125  13.258   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.57 on 110 degrees of freedom
## Multiple R-squared:  0.6151, Adjusted R-squared:  0.6116
## F-statistic: 175.8 on 1 and 110 DF,  p-value: < 2.2e-16
```

la fonction lm() est utilisée pour ajuster un modèle de régression linéaire aux données. Elle calcule les coefficients de la droite de régression (pente et ordonnée à l'origine) qui minimisent la distance entre les points observés et la droite prédite.

6. Pour tracer l'intervalle de confiance, cela pré-suppose certaines hypothèses sur le modèle. Lesquelles? Comment les vérifier en pratique et le faire?

La traçabilité de l'intervalle de confiance présuppose certaines conditions du modèle de régression, notamment la linéarité de la relation entre la variable dépendante et les variables indépendantes, l'indépendance des résidus, l'homoscédasticité, et la normalité des résidus. Pour vérifier ces conditions, des diagnostics de régression peuvent être utilisés, tels que l'inspection visuelle des graphiques de résidus et l'utilisation de tests statistiques appropriés comme le test de Shapiro-Wilk pour évaluer la normalité des résidus.

```r
residus = modele_reg$residuals
shapiro.test(residus)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residus
## W = 0.99235, p-value = 0.792
```