|  |
| --- |
| [Workspace loaded from ~/.RData]  > ventes<-c(4, 6; 7; 9, 12, 14, 16, 20, 25, 35)  Error: unexpected ';' in "ventes<-c(4, 6;"  > ventes<-c(4, 6, 7, 9, 12, 14, 16, 20, 25, 35)  > ventes  [1] 4 6 7 9 12 14 16 20 25 35  > length(ventes)  [1] 10  > publicite<-c(2, 3, 4, 6, 8, 9, 12, 13, 15, 18)  > publicite  [1] 2 3 4 6 8 9 12 13 15 18  > # on cherche à trouver un modèle linéaire qui va lier y en fonction de x  > # on veut expliquer les ventes en fonction des frais de publicité  > plot(publicite, ventes)  > model1<-lm(ventes~publicite)  > model1  Call:  lm(formula = ventes ~ publicite)  Coefficients:  (Intercept) publicite  -0.8298 1.7366  > # ventes (y) = -0.8298 + 1.7366\*publicite  > # régression simple: on explique y en fonction d'une seule variable  > # on va estimer y par la quantité ajustée (droite de régression): fitted values: y estimée (ventes estimées), représenté par y^  > # y^ = -0.8298 + 1.7366\*publicite  > ventes\_estimees<--0.8298 + 1.7366\*publicite  > ventes\_estimees  [1] 2.6434 4.3800 6.1166 9.5898 13.0630  [6] 14.7996 20.0094 21.7460 25.2192 30.4290  > # Résidu = y (observée) - y(estimée)  > residu<-ventes-ventes\_estimees  > residu  [1] 1.3566 1.6200 0.8834 -0.5898 -1.0630  [6] -0.7996 -4.0094 -1.7460 -0.2192 4.5710  > sum(residu)  [1] 0.004  > # on va calculer maintenant la somme des résidus ^2: variance résiduelle  > sum(residu^2)  [1] 47.42825  > somme\_des\_residus\_carres<-sum(residu^2)  > # on va calculer la variance (totale) du modèle: variance de y (ventes)  > # variance (y) = sum(y^2) - n\*(mean(y)^2)  > # variance (y) = sum(y-mean(y))^2  > vt<-sum(y^2)-(length(y)\*mean(y)^2)  Error: object 'y' not found  > vt<-sum(ventes^2)-(length(ventes)\*mean(ventes)^2)  > vt  [1] 837.6  > vt\_camara<-sum(ventes-mean(ventes))^2  > vt\_camara  [1] 5.04871e-29  > vt\_camara<-sum((ventes-mean(ventes))^2)  > vt\_camara  [1] 837.6  > # variance expliquée = variance totale - variance residuelle  > VE<-vt-somme\_des\_residus\_carres  > VE  [1] 790.1718  > # coefficient de détermination R^2 : VE/VT = 1 - VR/VT  > coef\_determination<-VE/vt  > coef\_determination  [1] 0.943376  > 1 -(somme\_des\_residus\_carres/vt)  [1] 0.943376  > # si R^2 est proche de 1 on dit qu'on a un bon ajustement linéaire, s'il est proche de 0 mauvais ajustement linéaire.  > # R^2 n'est autre que le carré du coef de corrélation linéaire entre x et y .  > cor(publicite, ventes, method=c("pearson"))  [1] 0.9712755  > cor(publicite, ventes, method=c("pearson"))^2  [1] 0.943376  > summary(model1)  Call:  lm(formula = ventes ~ publicite)  Residuals:  Min 1Q Median 3Q Max  -4.0099 -0.9975 -0.4050 1.2382 4.5702  Coefficients:  Estimate Std. Error t value  (Intercept) -0.8298 1.5575 -0.533  publicite 1.7366 0.1504 11.545  Pr(>|t|)  (Intercept) 0.609  publicite 2.88e-06 \*\*\*  ---  Signif. codes:  0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 2.435 on 8 degrees of freedom  Multiple R-squared: 0.9434, Adjusted R-squared: 0.9363  F-statistic: 133.3 on 1 and 8 DF, p-value: 2.877e-06  > model1$residuals  1 2 3 4  1.3564885 1.6198473 0.8832061 -0.5900763  5 6 7 8  -1.0633588 -0.8000000 -4.0099237 -1.7465649  9 10  -0.2198473 4.5702290  > sum(model1$residuals)  [1] -1.665335e-16  > model1$fitted.values  1 2 3 4  2.643511 4.380153 6.116794 9.590076  5 6 7 8  13.063359 14.800000 20.009924 21.746565  9 10  25.219847 30.429771  > abline(model1)  > # abline permet de tracer la droite de régression y^ = a^ + b^ x |
|  |
| |  | | --- | | > | |