Tutoriel : Régression linéaire avec R
Master 1 EDSB

2023-24

Plan

Ce document est un complément au cours *modèle linéaire* des M1 EDSB.

Références (en plus des slides du cours) :
J.J. Faraway (2002) Practical Regression and Anova using R.
Della Vedova, C. https://statistique-et-logiciel-r.com/
la-regression-lineaire-simple-avec-le-logiciel-r/
Cornillon, P.A., Hengartner, N., Matzner-Lober, E., Rouvière, L. (2019)
"Régression avec R", EDPsciences.

Outline

Exemples

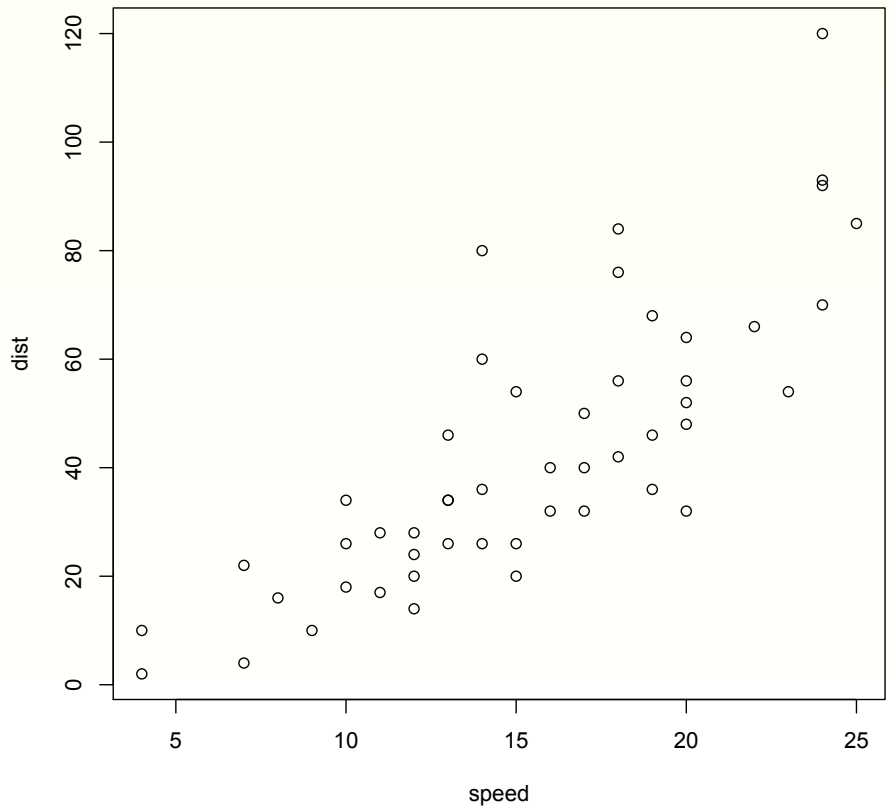
Régression linéaire simple - Exemples :

- Effet de la vitesse d'un véhicule sur la distance d'arrêt ?
- Effet de la surface d'un logement sur son loyer ?
- Effet du montant investi dans une campagne publicitaire sur les ventes ?
- Effet du temps de travail sur le salaire ?

Exemple : Distance d'arrêt d'un véhicule en fonction de sa vitesse

On peut représenter les données par un nuage de points pour se donner une idée sur l'éventuelle relation entre les 2 variables :

```
> plot(cars)
```

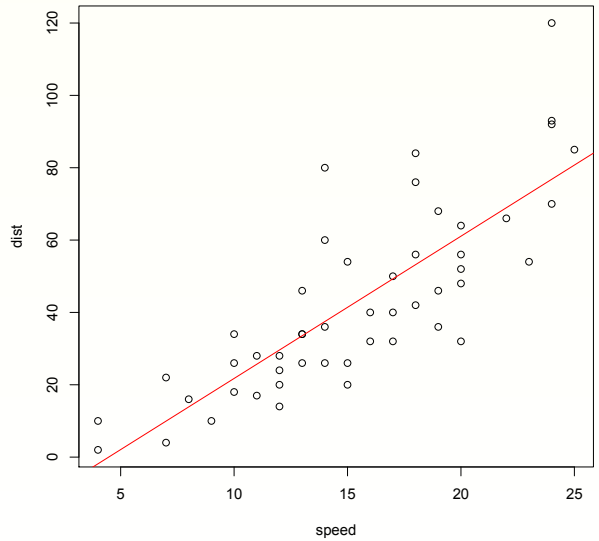


Mise en pratique R : le fonction lm()

```
> mod1<-lm(dist~speed,data=cars)
> coef(mod1)
(Intercept)       speed
 -17.579095    3.932409
```

- On a donc $\widehat{b} = -17.58$ et $\widehat{a} = 3.93$.
- On en déduit $\widehat{y}_i = -17.58 + 3.93x_i$. Traçons cette droite de régression sur le nuage de points :

```
> plot(cars)
> abline(mod1,col=2)
```



Mise en pratique R - Sommes des carrés et R^2.

Tester si le modèle est significatif :

```
> anova(mod1)
Analysis of Variance Table

Response: dist
          Df Sum Sq Mean Sq F value    Pr(>F)
speed      1  21186 21185.5  89.567 1.490e-12 ***
Residuals 48  11354   236.5
---
Signif. codes:  0 Ô***Õ 0.001 Ô**Õ 0.01 Ô*Õ 0.05 Ô.Õ 0.1 Ô Õ 1
```

Coefficient de détermination :

```
> 21186/(21186+11354)
[1] 0.6510756

> (cor(cars$speed,cars$dist))^2
[1] 0.6510794
```

Ou encore via la commande *summary*.

Mise en pratique R - Tests associés au modèle

```
> summary(mod1)

Call:
lm(formula = dist ~ speed, data = cars)

Residuals:
    Min      1Q  Median      3Q     Max
-29.069  -9.525  -2.272   9.215  43.201

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -17.5791     6.7584  -2.601   0.0123 *
speed         3.9324     0.4155   9.464 1.49e-12 ***
---
Signif. codes:  0 Ô***Õ 0.001 Ô**Õ 0.01 Ô*Õ 0.05 Ô.Õ 0.1 Ô Õ 1

Residual standard error: 15.38 on 48 degrees of freedom
Multiple R-squared: 0.6511,Adjusted R-squared: 0.6438
F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.490e-12
```

Mise en pratique R - Tests associés au modèle

Commentaires :

- On retrouve la valeur du R^2 et celle de la stat de test F du modèle, avec la p-value associée. Ici, sous les postulats du modèle linéaire, on rejette H_0 et donc on conclut à un effet de la vitesse du véhicule sur la distance de freinage.
- Les t-values et les probas associées correspondent aux tests individuels de Student d'égalité à 0 des paramètres.
- Pour mémoire, *residual standard error* correspond à l'estimation de l'écart-type résiduel (l'estimation de la variance résiduelle est donnée par lé carré résiduel moyen du modèle : 236.5).

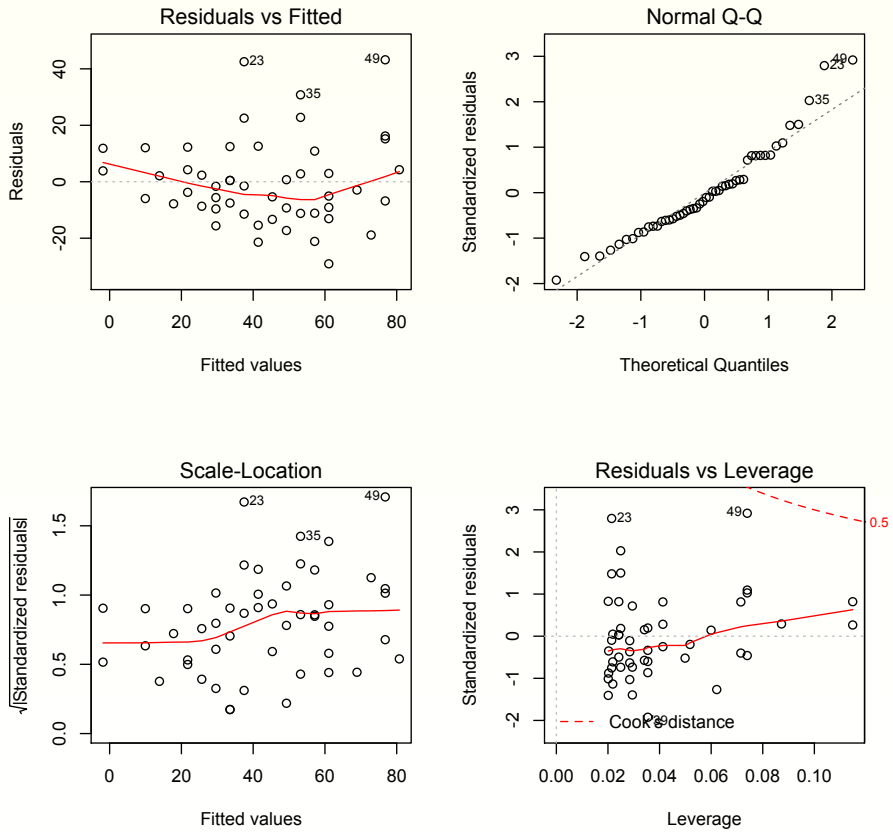
Il est possible de construire des intervalles de confiance pour les paramètres estimés :

```
> confint(mod1)
                 2.5 %    97.5 %
(Intercept) -31.167850 -3.990340
speed         3.096964  4.767853
```

Mise en pratique R - Validation du modèle : graphes associés

Validation : graphique (des approches tests seraient aussi envisageables mais attention les résidus estimés ne sont ni indépendants ni de variance égale)

```
> par(mfrow=c(2,2))
> plot(mod1) # via residus : mod1$residuals ou residuals(mod1)
```



Mise en pratique R - Graphes associés - Vérification graphique

- Le premier graphique trace les résidus en fonction des valeurs ajustées. On s'attend à un graphe sans forme particulière (*patatoide !*). La tendance moyenne est tracée en rouge et les résidus de trop forte valeur par rapport à l'hypothèse d'une distribution normale sont indiqués par leur numéro. On s'attend à une ligne rouge à peu près horizontale (pas de tendance dans les résidus) et pas plus de 5% des points marqués. La variabilité des points doit être à peu près constante (sans forme systématique, type entonnoir).

- Le graphique 2 (QQ-plot) permet de vérifier graphiquement l'hypothèse de normalité des résidus : si les points sont à peu près alignés en se confondant avec la première bissectrice des axes, on peut dire que les résidus suivent une loi normale. Des écarts aux extrémités ne sont pas trop génants si le graphe est correctement aligné pour les valeurs centrales (loi unimodale symétrique).

- Le troisième graphe répète le premier à une autre échelle différente, permettant de mieux appréhender les fluctuations.

- Le graphique 4 (Cooks D) permet de repérer les points pour lesquels la régression linéaire pourrait être mal (ou pas) adaptée : ces points ont une forte influence sur le résultat d'estimation du vecteur des paramètres (par ex, parce qu'ils s'écartent trop de la droite de régression). Les points trop influents sont identifiés au-dela de la zone rouge.

Mise en pratique R

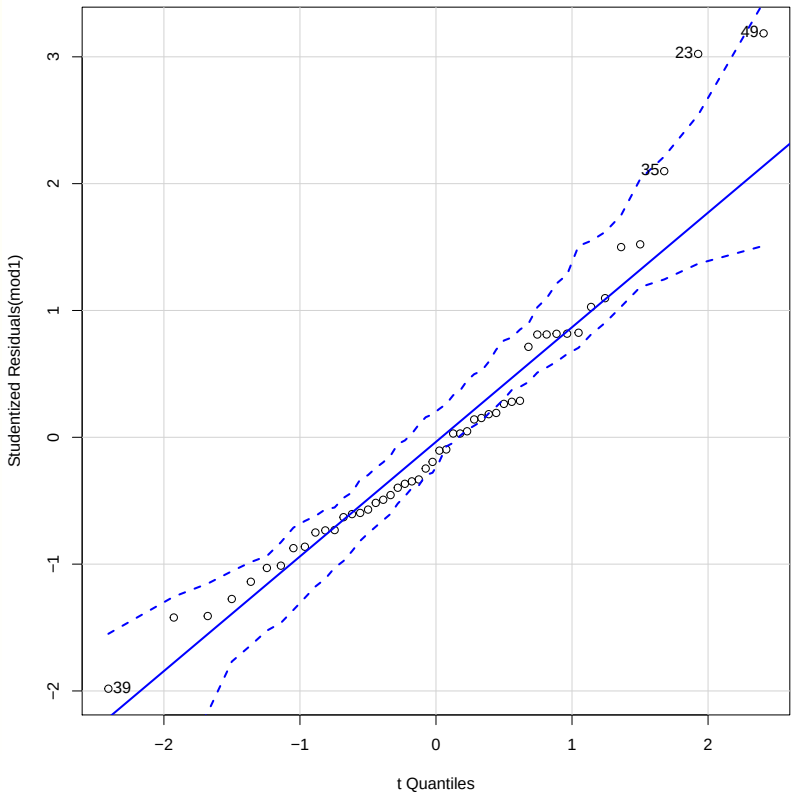
Si questions à propos d'individus peut-être trop influents, possibilité d'expliciter les effets sur les estimations . . . par exemple :

```
#effets  individus 2 et 3 ? les retirer pour estimer et comparer !

mod1_sans2_3 <- lm(dist~speed,data=cars[-c(2,3),])
compareCoefs(mod1,mod1_sans2_3)
car::compareCoefs(mod1,mod1_sans2_3) #necessite package car

Calls:
1: lm(formula = dist ~ speed, data = cars)
2: lm(formula = dist ~ speed, data = cars[-c(2, 3), ])

             Model 1 Model 2
(Intercept)  -17.58  -18.89
SE             6.76    7.57

speed          3.932   4.008
SE             0.416   0.457
```

Mise en pratique R

Qqplot résidus standardisés et identifier individus écarts les plus extrêmes

```
car::qqPlot(mod1) #identifier ceux qui "sortent"
car::qqPlot(mod1,id=list(n=4)) # identifier les 4 plus extremes
```



Mise en pratique R - Prédictions

- Les 4 commandes suivantes sont équivalentes et renvoient les n valeurs prédites $\widehat{y}_i$ selon le modèle estimé.

```
> fitted(mod1)
> mod1$fitted
> predict(mod1)
> coef(mod1)[1]+coef(mod1)[2]*cars$speed
```

- Si on veut la valeur prédite $\widehat{y}$ pour une valeur donnée de x, par exemple $x = 5.5$, on a 2 solutions :

```
# formule explicite avec les estimations trouvées

> coef(mod1)[1]+coef(mod1)[2]*5.5
(Intercept)
   4.049153
```

Mise en pratique R - Prédictions et confiance

```
# fonction  predict()
> predict(mod1,list(speed=5.5))
       1
4.049153
```

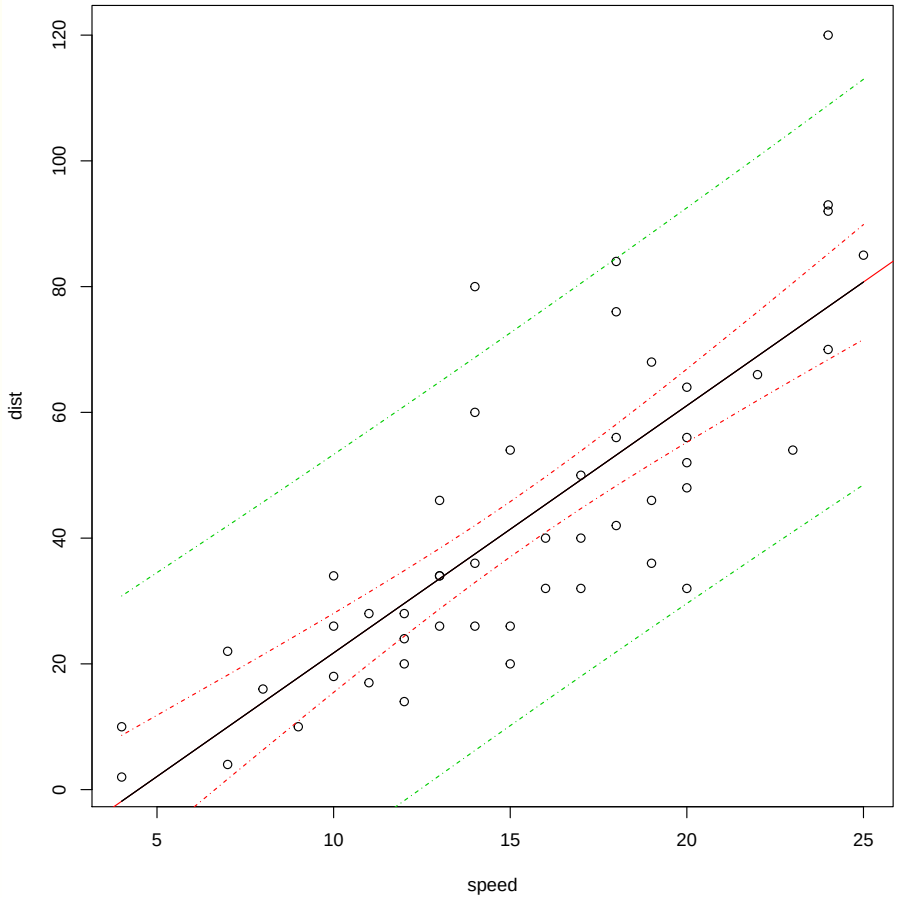
On peut aussi sortir les valeurs de l'intervalle de confiance de l'estimation de l'espérance de y ou l'intervalle de prédiction de y à x fixé :

```
> predict(mod1,list(speed=5.5),interval="confidence")
       fit       lwr      upr
1 4.049153 -5.306705 13.40501

> predict(mod1,list(speed=5.5),interval="prediction")
       fit       lwr      upr
1 4.049153 -28.25793 36.35623
```

Représentation graphique

```
plot(cars)
abline(mod1)
#abline(mod1, col=2) # rouge

IP<- predict(mod1,interval="prediction",level = .95)
IC<- predict(mod1,interval="confidence",level= .95)

matlines(cars$speed, IC, lty = c(1, 4, 4), col = c(1, 2, 2))
matlines(cars$speed, IP, lty = c(1, 4, 4), col = c(1, 3, 3))
```

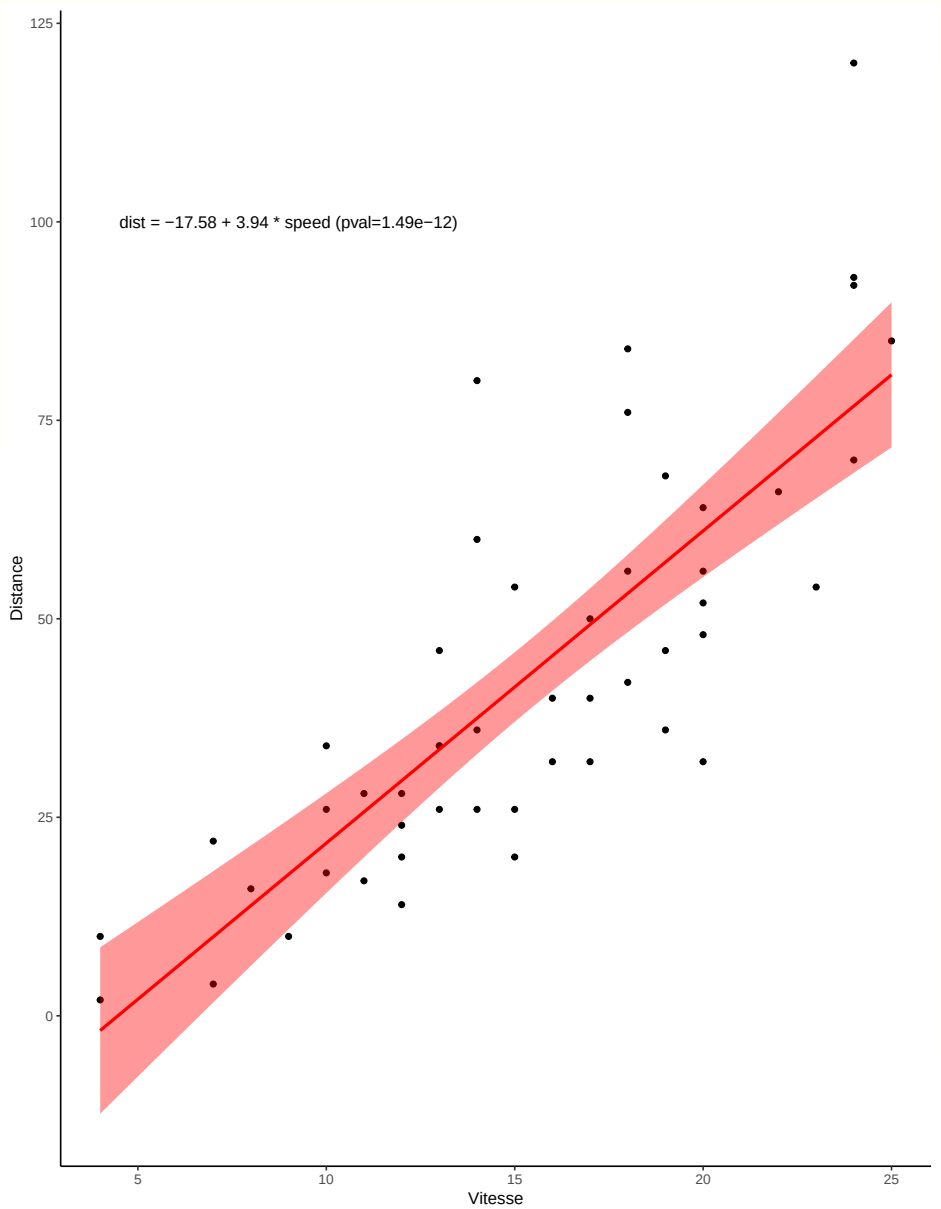
Représentation graphique



Représentation graphique : utilisation package ggplot2

```
library(ggplot2)


# IC fait automatiquement avec geom_smooth()

ggplot(cars, aes(y=dist, x=speed))+
  geom_point()+
  geom_smooth(colour="red", method="lm", fill="red") +
  ylab("Distance")+
  xlab("Vitesse") +
  theme_classic()+
  annotate("text", x = 9, y = 100, label = "dist = -17.58 + 3.94 * spee
```

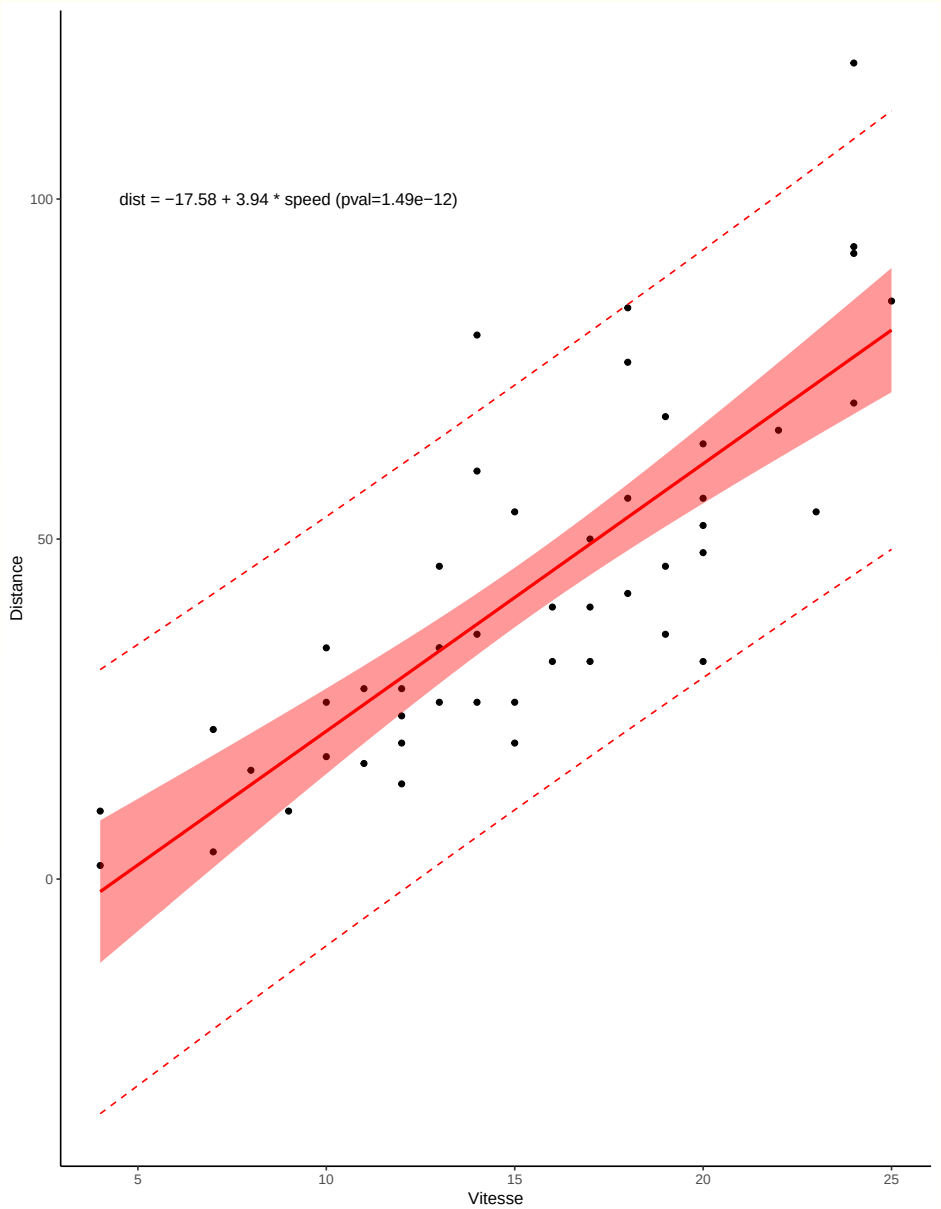
Représentation graphique : utilisation package ggplot2



Représentation graphique : utilisation package ggplot2

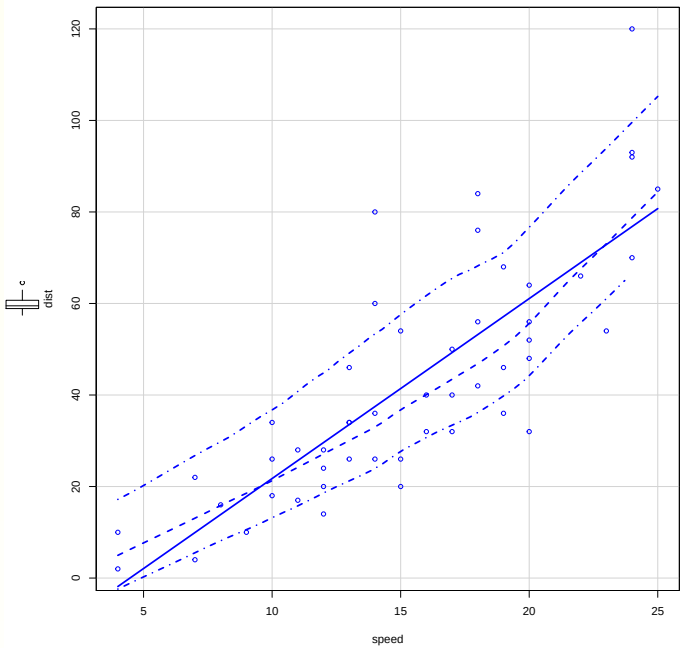
```
# rajouter intervalle predictions via geom_line()

cars2<-cbind(cars,predict(mod1,interval="prediction"))

ggplot(cars2, aes(y=dist, x=speed))+
  geom_point()+
  geom_smooth(colour="red", method="lm", fill="red") +
  geom_line(aes(y=lwr), color = "red", linetype = "dashed")+
  geom_line(aes(y=upr), color = "red", linetype = "dashed")+
  ylab("Distance")+
  xlab("Vitesse") +
  theme_classic()+
  annotate("text", x = 9, y = 100, label = "dist = -17.58 + 3.94 * spee
```

Représentation graphique : utilisation package ggplot2



Représentation graphique avec approche non-paramétrique

```
> car::scatterplot(dist~speed, data=cars)# approche NP
```



La droite pleine est la droite de régression ; la ligne centrale en pointillé est la courbe de régression locale de type lowess (approche NP) qui indique la tendance globale entre les deux variables. Les deux lignes extérieures donnent l'IC à 95% de la courbe lowess $\rightsquigarrow$ *permet d'apprécier les comportements sous et hors modèle imposé . . .*

Outline

Régression linéaire multiple : un exemple

On considère le jeu de données *accidents* dans lequel le nombre d'accidents par mois dans une banlieue de Montréal est donné. Il y a 3 ans de données ($3\times$ 12) et des variables climatiques suivantes :

- *tmax* : température max du mois ;
- *tmin* : température min du mois ;
- *tmoy* : température moyenne du mois ;
- *tcon* : température moyenne de congélation ;
- *soleil* : durée moyenne ensoleillement du mois ;
- *plui* : précipitation moyenne du mois ;
- *neig* : précipitation moyenne neige du mois ;
- *jour* : nombre de jour du mois.

Le but est de chercher à expliquer le nombre d'accidents mensuels selon les variables climatiques à disposition.

⚠**attention, ce qui suit n'est qu'un exemple d'utilisation des fonctions** R **pour la régression multiple . . .** au final le modèle obtenu n'est pas forcément le meilleur modèle linéaire car on ne va pas se poser par exemple la question d'un éventuel effet *année* ou *nbre de jours* (si pris en qualitatifs) !

Graphe des paires de variables :

```
pairs(data)

# ### modélisation
# approche "a la main"
# part de toutes les var et on enleve de proche en proche les NS

mod_acc_1<- lm(acc~.,data=data)
# idem mod_acc_1=lm(data$acc~data$tmax+data$tmin+data$tmoy
#           +data$tcon+data$soleil+data$plui+data$neig+data$jour)

summary(mod_acc_1)
```

```
lm(formula = acc ~ ., data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-21.5035  -7.7130  -0.0703   6.8611  28.3164

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 70.96651   87.89237   0.807   0.4265
tmax         2.93284    8.76705   0.335   0.7406
tmin        -2.16064   10.77910  -0.200   0.8426
tmoy        -5.84881   18.09550  -0.323   0.7490
tcon         4.80028    2.36833   2.027   0.0527 .
soleil      -0.03285    0.06736  -0.488   0.6297
plui         0.04848    0.07896   0.614   0.5443
neig         0.04642    0.17336   0.268   0.7909
jour         0.06666    3.00649   0.022   0.9825
---
Residual standard error: 12.62 on 27 degrees of freedom
Multiple R-squared:  0.4914,Adjusted R-squared:  0.3407
F-statistic: 3.261 on 8 and 27 DF,  p-value: 0.00992
```

On enlève $jour$ et on recommence . . . l'idée est de supprimer de proche en proche (i.e. une par une) les variables les moins significatives jusqu'à obtenir un modèle n'ayant plus que des variables significatives !

```
> mod_acc_1=lm(acc~tmax+tmin+tmoy+tcon+soleil+plui+neig,data=data)
> summary(mod_acc_1)
.
.
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 72.89062   13.70334   5.319 1.16e-05 ***
tmax         2.92966    8.60800   0.340   0.7361
tmin        -2.19416   10.48037  -0.209   0.8357
tmoy        -5.82333   17.73372  -0.328   0.7451
tcon         4.81302    2.25621   2.133   0.0418 *
soleil      -0.03267    0.06566  -0.498   0.6227
plui         0.04876    0.07658   0.637   0.5295
neig         0.04725    0.16629   0.284   0.7784
---
```

$\hookrightarrow$ enlever $tmin$ . . .

```
mod_acc_1=lm(data$acc~data$tmax+data$tmoy+data$tcon+data$sol+data$plui+
summary(mod_acc_1)

# enlever neig

etc ...
> mod_acc_1=lm(data$acc~data$tmax+data$tmoy+data$tcon)
> summary(mod_acc_1)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   78.561      9.210   8.530 9.52e-10 ***
data$tmax      4.730      4.024   1.175  0.24847
data$tmoy    -10.681      5.432  -1.966  0.05797 .
data$tcon      5.598      1.666   3.361  0.00202 **
--
Residual standard error: 11.83 on 32 degrees of freedom
Multiple R-squared:  0.4697,Adjusted R-squared:   0.42
F-statistic: 9.448 on 3 and 32 DF,  p-value: 0.0001278
```

```
mod_acc_1=lm(data$acc~data$tmoy+data$tcon)
summary(mod_acc_1)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   86.072      6.671  12.902 1.92e-14 ***
data$tmoy     -4.405      1.004  -4.386 0.000111 ***
data$tcon      4.111      1.090   3.772 0.000639 ***
---
Residual standard error: 11.9 on 33 degrees of freedom
Multiple R-squared:  0.4468,Adjusted R-squared:  0.4133
F-statistic: 13.33 on 2 and 33 DF,  p-value: 5.721e-05
```

Le modèle avec les variables explicatives $tmoy$ et $tcon$ est donc candidat pour la modélisation recherchée ...

Remarque : on peut vérifier que la comparaison de deux modèles emboités redonne bien la p-value associée au paramètre testé ...

```
mod_acc_2=lm(data$acc~data$tmax+data$tmoy+data$tcon)
anova(mod_acc_1,mod_acc_2)

Analysis of Variance Table

Model 1: data$acc ~ data$tmoy + data$tcon
Model 2: data$acc ~ data$tmax + data$tmoy + data$tcon
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     33 4675.6
2     32 4482.1  1    193.54 1.3818 0.2485

# retrouve bien meme p-value de comparaison que celle associée à tmax
# dans modèle à 3 var : tmax, tmoy, tcon
```

Sélection de modèles : approches automatiques pas à pas

```
mod_min<-lm(acc~1,data=data)

mod_complet<-lm(data$acc~data$tmax+data$tmin+data$tmoy+data$tcon
                +data$sol+data$plui+data$neig+data$jour)

modstepwise=step(mod_complet,direction="both", test='F')

modstepwiseA=step(mod_complet,direction="both")# par defaut AIC

modback=step(mod_complet,scope=list(lower=mod_min),direction='backward'
test='F')
summary(modback)  # retrouve ca qu'on a fait a la main : tmoy tcon

modfor=step(mod_min,scope=list(upper=mod_complet),direction='forward',t
summary(modfor)  #ne garde que sol
```

$\hookrightarrow$ *peut donner des modèles différents, candidats pour une modélisation !* . . .
qu'il faut ensuite regarder pour choisir celui qui semble le mieux adapté (donc voir le comportement des résidus, sélectionner sur critère particulier à décider)

La librairie "Leaps" possède une fonction qui **recherche de façon exhaustive les meilleurs sous-ensemble de régresseurs parmi ceux possibles** . . . on peut utiliser les critères Cp, R^2 ou $adjR^2$; la fonction renvoie les *nbest* meilleurs sous-modèles des différentes tailles . . . (même type que fonction *regsubsets())*

```
library(leaps)
> X<-leaps( x=data[,2:8], y=data[,1], int=TRUE,
                  names=names(data)[2:8], method='Cp', nbest=2)
> X
$which
   tmax  tmin  tmoy  tcon soleil  plui  neig
1 FALSE FALSE FALSE FALSE   TRUE FALSE FALSE
1  TRUE FALSE FALSE FALSE  FALSE FALSE FALSE
2 FALSE  TRUE FALSE  TRUE  FALSE FALSE FALSE
2 FALSE FALSE  TRUE  TRUE  FALSE FALSE FALSE
.
6  TRUE FALSE  TRUE  TRUE   TRUE  TRUE  TRUE
6  TRUE  TRUE  TRUE  TRUE   TRUE  TRUE FALSE
7  TRUE  TRUE  TRUE  TRUE   TRUE  TRUE  TRUE

> plot(X$size,X$Cp)
> abline(0,1) # visualiser les Cp des modeles retenus (proches diago ?)
X
```

Etude de l'influence des individus etn recherche de corrélation entre variables pour un modèle donné : *modstepwise*

```
# obtenir les leverage de $Y_i$ et les changements dans les résultats
# d'estimation lorsque le i eme individu est enleve
> lm.influence(modstepwise)

# mesures generales influence individu
> influence.measures(modstepwise)
## dfb = DFBETAS ; dffit = DFFITS ; cov.r = covariance ratio ;
##cook.d = distance de Cook ; hat = dig(H) ;
## inf = repere influence data (individu(s) qui pourrai(en)t poser pbs)

# obtenir résumes des individus potentiellement à pbs
> summary(influence.measures(modstepwise))

> modif<-influence.measures(modstepwise)
> which(apply(modif$is.inf,1,any)) # sortir les "influence data"
```

```
# graphes résumant des mesures d'influence individu
> influenceIndexPlot(modstepwise,
                     vars=c("Cook", "Studentized","Bonf", "hat") )
# Bonferroni test ajusté de valeurs aberrantes
#pour residus standardisés

# test residus outlier :
> outlierTest(modstepwise) # ici resultat NS

## si on veut voir l'influence d'individus sur l'ajustement,
## on peut estimer avec et sans eux et comparer
## ici par ex retirer les individus 6 et 26
> modbis <- lm(data$acc~data$tmoy+data$tcon, data=data[-c(6,26),])

compareCoefs(modstepwise,modbis)

# recupérer le "variance influence factor" pour juger de la colinéarité
# des variables explicatives
> vif(modstepwise)
```

Etude des résidus : *même approche que régression simple . . .*

```
> par(mfrow=c(2,2))
> plot(modstepwise)
```

Prédiction : *mêmes approches que régression simple . . .*

```
> modfin<-lm(acc~tmoy+tcon,data=data) # le modele retenu (donc validé !
> predict(modfin,data.frame(tmoy=2,tcon=-1), interval = "conf")

      fit      lwr      upr
1 73.1519 65.32827 80.97552

##valeurs des covar pour prévisions
> data_prev <- data.frame(tmoy=c(2,4),tcon=c(-1,-1))

> predict(modfin,data_prev, interval = "conf")

       fit      lwr      upr
1 73.15190 65.32827 80.97552
2 64.34184 59.52731 69.15637

> predict(modfin,data_prev, interval = "predict")

       fit      lwr      upr
1 73.15190 47.70239 98.60140
2 64.34184 39.65079 89.03289
```