TD1 : Rappels sur le modèle linéaire

Exercice 1 : Petits exemples

Exercice 1.1 : jeu de données iris

1.

```
data('iris')
dim(iris)
```

```
## [1] 150   5
```

```
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

2.

```
model <- lm(Petal.Length ~ ., data = iris) # Y = Petal.Length et X = variables restantes
model # estimateurs associés
```

```
##
## Call:
## lm(formula = Petal.Length ~ ., data = iris)
##
## Coefficients:
##       (Intercept)       Sepal.Length        Sepal.Width        Petal.Width
##           -1.1110             0.6080            -0.1805             0.6022
## Speciesversicolor   Speciesvirginica
##            1.4634             1.9742
```

```
summary(model) # sigificativité des estimateurs
```

```
##
## Call:
## lm(formula = Petal.Length ~ ., data = iris)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.78396 -0.15708  0.00193  0.14730  0.65418
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)       -1.11099    0.26987  -4.117 6.45e-05 ***
## Sepal.Length       0.60801    0.05024  12.101  < 2e-16 ***
## Sepal.Width       -0.18052    0.08036  -2.246   0.0262 *
## Petal.Width        0.60222    0.12144   4.959 1.97e-06 ***
## Speciesversicolor  1.46337    0.17345   8.437 3.14e-14 ***
## Speciesvirginica   1.97422    0.24480   8.065 2.60e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2627 on 144 degrees of freedom
## Multiple R-squared:  0.9786, Adjusted R-squared:  0.9778
## F-statistic:  1317 on 5 and 144 DF,  p-value: < 2.2e-16
```

Toutes les variables sont retenues comme significatives par le modèle (avec une p-value < 0.05). Cependant, la variable principale semble être la longueur de la sépale **Sepal.Length**. On remarque que la variable **Species** apparaît deux fois. Ceci est dû au fait qu'il s'agit d'une variable discrète à trois états. Il faut donc la modéliser à l'aide de deux coefficients dans le vecteur β : l'un des états (ici **setosa**) est choisi comme base et la variable **Species** est remplacée par deux indicatrices (**versicolor** et **virginica**). Les coefficients liés à ces variables indiquent alors la différence de valeur moyenne quand on passe de **setosa** à l'état correspondant. En général, on ne prend pas les variables qualitatives comme explicatives dans un modèle linéaire gaussien.

3.

```
iris$versicolor <- iris$Species == 'versicolor'
modellogit <- glm(versicolor ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
                  family = binomial, data = iris) # modèle de régression logistique
summary(modellogit)
```

```
##
## Call:
## glm(formula = versicolor ~ Sepal.Length + Sepal.Width + Petal.Length +
##     Petal.Width, family = binomial, data = iris)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1280  -0.7668  -0.3818   0.7866   2.1202
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    7.3785     2.4993   2.952 0.003155 **
## Sepal.Length  -0.2454     0.6496  -0.378 0.705634
## Sepal.Width   -2.7966     0.7835  -3.569 0.000358 ***
## Petal.Length   1.3136     0.6838   1.921 0.054713 .
## Petal.Width   -2.7783     1.1731  -2.368 0.017868 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 190.95  on 149  degrees of freedom
## Residual deviance: 145.07  on 145  degrees of freedom
## AIC: 155.07
##
## Number of Fisher Scoring iterations: 5
```

La variable **Sepal.Width** caractérise le mieux l'espcèce **versicolor**.

Exercice 1.2 : jeu de données airquality

1.

```
data("airquality")
dim(airquality)
```

```
## [1] 153   6
```

```
head(airquality)
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 5    NA      NA 14.3   56     5   5
## 6    28      NA 14.9   66     5   6
```

2.

```
modelair <- lm(Ozone ~ ., data = airquality) # Y = Ozone et X = variables restantes
modelair
```

```
##
## Call:
## lm(formula = Ozone ~ ., data = airquality)
##
## Coefficients:
## (Intercept)      Solar.R         Wind         Temp        Month          Day
##   -64.11632      0.05027     -3.31844      1.89579     -3.03996      0.27388
```

```
summary(modelair)
```

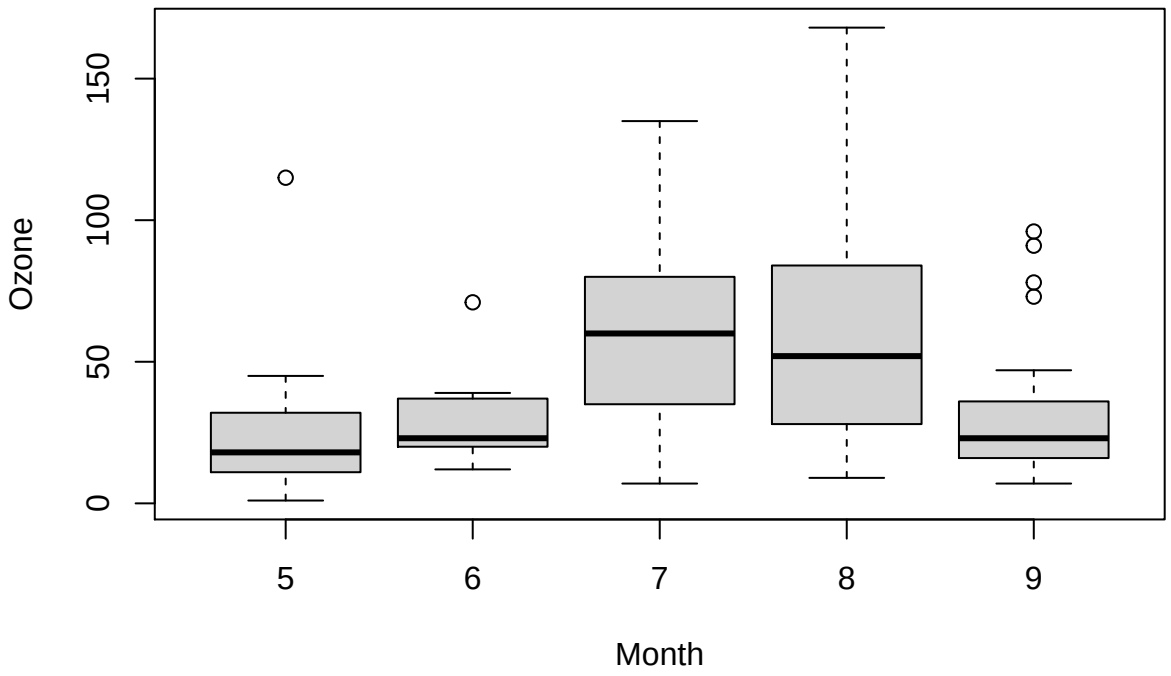
```
##
## Call:
## lm(formula = Ozone ~ ., data = airquality)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.014 -12.284  -3.302   8.454  95.348
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -64.11632   23.48249  -2.730  0.00742 **
## Solar.R       0.05027    0.02342   2.147  0.03411 *
## Wind         -3.31844    0.64451  -5.149 1.23e-06 ***
## Temp          1.89579    0.27389   6.922 3.66e-10 ***
## Month        -3.03996    1.51346  -2.009  0.04714 *
## Day           0.27388    0.22967   1.192  0.23576
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.86 on 105 degrees of freedom
```

```
##   (42 observations deleted due to missingness)
## Multiple R-squared:  0.6249, Adjusted R-squared:  0.6071
## F-statistic: 34.99 on 5 and 105 DF,  p-value: < 2.2e-16
```

Les coefficients les plus significatifs sont un coefficient positif associé à la **Temp** et un coefficient négatif associé au **Wind**, ce qui correspond bien à nos à priori (plus il fait chaud, plus le taux d'ozone est élevé, et inversement pour le force du vent).

3.

```
boxplot(Ozone ~ Month, data = airquality)
```




```
summer <- airquality[airquality$Month < 8, ]
modelair2 <- lm(Ozone ~ ., data = summer) # Y = Ozone et X = variables restantes
modelair2
```

```
##
## Call:
## lm(formula = Ozone ~ ., data = summer)
##
## Coefficients:
## (Intercept)      Solar.R         Wind         Temp        Month          Day
##   -60.87857      0.03958     -2.77841      1.81846     -3.11778      0.17306
```

```
summary(modelair2)
```

```
##
## Call:
## lm(formula = Ozone ~ ., data = summer)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.408 -16.375  -1.268  11.440  65.525
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -60.87857   28.37084  -2.146  0.03649 *
## Solar.R       0.03958    0.03125   1.266  0.21091
## Wind         -2.77841    0.83194  -3.340  0.00154 **
## Temp          1.81846    0.52299   3.477  0.00102 **
## Month        -3.11778    5.12936  -0.608  0.54590
## Day           0.17306    0.31867   0.543  0.58936
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.65 on 53 degrees of freedom
##   (33 observations deleted due to missingness)
## Multiple R-squared:  0.5984, Adjusted R-squared:  0.5605
## F-statistic: 15.79 on 5 and 53 DF,  p-value: 1.676e-09
```

Si on réduit les données aux mois de Mai, Juin et Juillet, les conclusions restent identiques que sur l'ensemble des données.

```
res <- lm(Ozone ~ Month, data = summer)
summary(res)
```

```
##
## Call:
## lm(formula = Ozone ~ Month, data = summer)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.357 -15.857  -3.857  12.143  93.143
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -66.893     22.222  -3.010  0.00384 **
## Month         17.750      3.661   4.849 9.41e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.4 on 59 degrees of freedom
##   (31 observations deleted due to missingness)
## Multiple R-squared:  0.285,  Adjusted R-squared:  0.2728
## F-statistic: 23.51 on 1 and 59 DF,  p-value: 9.414e-06
```

Si on explique le taux d'ozone par la variable **Month**, on obtient un coefficient très significatif et positif. Cette fluctuation par la présence ou non des variables **Wind** et **Temp** laisse penser que la variable **Month** est sans doute très corrélée à l'une ou l'autre de ces variables, ce qui rend l'interprétation des coefficients difficiles.

```
cor.test(summer$Wind, summer$Month)
```

```
##
##  Pearson's product-moment correlation
##
## data:  summer$Wind and summer$Month
## t = -3.0713, df = 90, p-value = 0.002818
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
```

```
##  -0.4823942 -0.1101398
## sample estimates:
##       cor
## -0.308009
```

```
cor.test(summer$Temp, summer$Month)
```

```
##
##  Pearson's product-moment correlation
##
## data:  summer$Temp and summer$Month
## t = 11.397, df = 90, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6691003 0.8410128
## sample estimates:
##       cor
## 0.7685878
```

Les variables semblent corrélées.

Exercice 2 : Problème de colinéarité

Exercice 2.1 : jeu de données Prostate

1.

```
library(lasso2)
```

```
## R Package to solve regression problems while imposing
##   an L1 constraint on the parameters. Based on S-plus Release 2.1
## Copyright (C) 1998, 1999
## Justin Lokhorst   <jlokhors@stats.adelaide.edu.au>
## Berwin A. Turlach <bturlach@stats.adelaide.edu.au>
## Bill Venables     <wvenable@stats.adelaide.edu.au>
##
## Copyright (C) 2002
## Martin Maechler <maechler@stat.math.ethz.ch>
```

```
data("Prostate")
dim(Prostate)
```

```
## [1] 97  9
```

```
head(Prostate)
```

```
##       lcavol  lweight age      lbph svi       lcp gleason pgg45       lpsa
## 1 -0.5798185 2.769459  50 -1.386294   0 -1.386294       6     0 -0.4307829
## 2 -0.9942523 3.319626  58 -1.386294   0 -1.386294       6     0 -0.1625189
## 3 -0.5108256 2.691243  74 -1.386294   0 -1.386294       7    20 -0.1625189
## 4 -1.2039728 3.282789  58 -1.386294   0 -1.386294       6     0 -0.1625189
## 5  0.7514161 3.432373  62 -1.386294   0 -1.386294       6     0  0.3715636
## 6 -1.0498221 3.228826  50 -1.386294   0 -1.386294       6     0  0.7654678
```

```
modellcavol <- lm(lcavol ~ ., data = Prostate)
summary(modellcavol)
```

```
##
```

```
## Call:
## lm(formula = lcavol ~ ., data = Prostate)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.88603 -0.47346 -0.03987  0.55719  1.86870
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.260101   1.259683  -1.794   0.0762 .
## lweight     -0.073166   0.174450  -0.419   0.6759
## age          0.022736   0.010964   2.074   0.0410 *
## lbph        -0.087449   0.058084  -1.506   0.1358
## svi         -0.153591   0.253932  -0.605   0.5468
## lcp          0.367300   0.081689   4.496 2.10e-05 ***
## gleason      0.190759   0.154283   1.236   0.2196
## pgg45       -0.007158   0.004326  -1.654   0.1016
## lpsa         0.572797   0.085790   6.677 2.11e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6998 on 88 degrees of freedom
## Multiple R-squared:  0.6769, Adjusted R-squared:  0.6475
## F-statistic: 23.04 on 8 and 88 DF,  p-value: < 2.2e-16
```

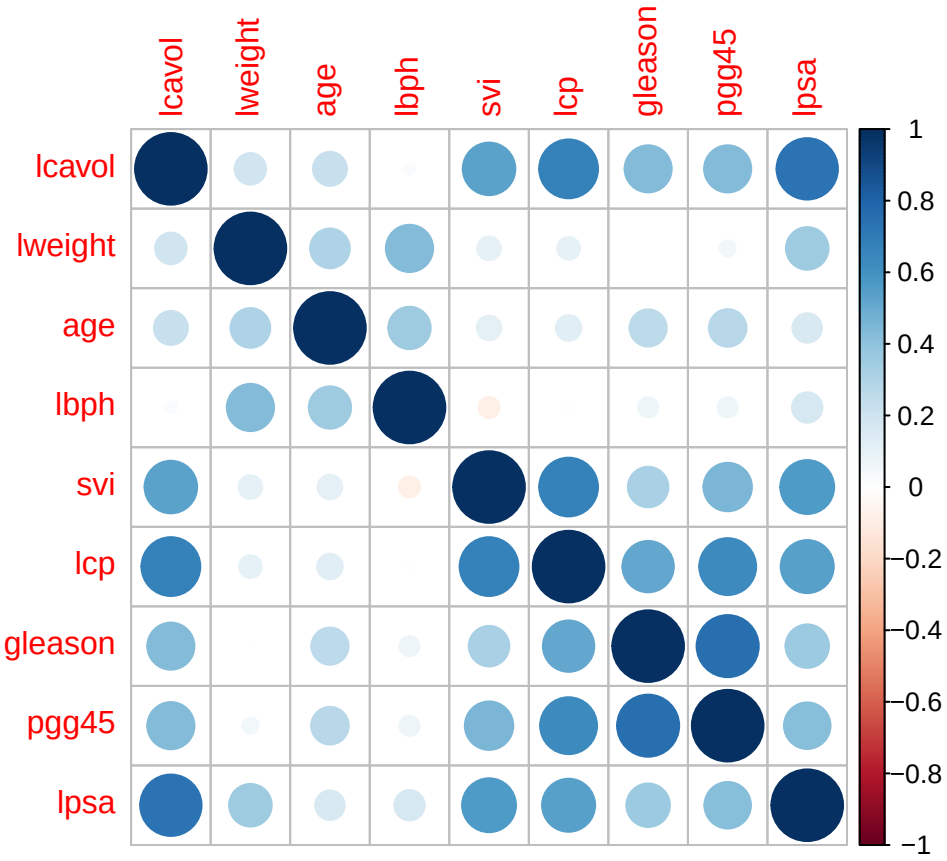
Trois variables (**age**, **lcp** et **lpsa**) semblent expliquer la variable **lcavol** (p-valeurs significatives).

2.

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
cov <- cor(Prostate)
corrplot(cov)
```



La variable **lpsa** est très corrélée à la variable réponse **lcavol**, ce qui peut expliquer pourquoi elle a une influence significative. Les autres variables, **pgg45**, **gleason**, **lcp** et **svi** semblent être corrélées. Cela peut poser un problème lorsque l'on interprète les variables significatives dans le modèle linéaire.

3.

```
library(car)
```

```
## Loading required package: carData
```

```
VIF <- vif(modellcavol)
VIF
```

```
##  lweight      age     lbph      svi      lcp  gleason    pgg45     lpsa
## 1.471496 1.306232 1.392115 2.166568 2.557640 2.433439 2.918987 1.922553
```

VIF $< 10 \Rightarrow$ les variables ne sont pas colinéaires.

Exercice 2.2 : passage à la grande dimension

1.

On se donne p-1 variables très corrélées à X^1 et p-1 très corrélées à X^{p+1}.

```
library(MASS)

CreateData <- function(p, n, rho){# p : nb variables, n : taille de l'échantillon

  # créer la matrice de covariance
```

```
  sigma1 <- matrix(rho,p,p)
  diag(sigma1) <- 1
  sigma2 <- rbind(cbind(sigma1, matrix(0,p,p)), cbind(matrix(0,p,p), sigma1)) # concaténation 2p+1

  # créer la matrice X
  X <- mvrnorm(n = n, mu = rep(0, nrow(sigma2)), sigma2)

  # Y = X^1 + X^{p+1} + epsilon
  data <- cbind(X, X[,1] + X[,p+1] + rnorm(n, 0.5))
  colnames(data) <- paste('X', 1:dim(data)[2], sep = "")
  colnames(data)[2*p+1] <- 'Y'
  return(data)
}
```

2.

```
rho = 0.1
```

n >> p:

```
n = 60
p = 5

donnees <- data.frame(CreateData(p, n, rho))
modelgd <- lm(Y ~ ., data = donnees)
summary(modelgd)
```

```
##
## Call:
## lm(formula = Y ~ ., data = donnees)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.79724 -0.55484  0.03974  0.50870  1.86577
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.70096    0.14964   4.684 2.26e-05 ***
## X1           0.96648    0.13783   7.012 6.35e-09 ***
## X2          -0.08555    0.15674  -0.546    0.588
## X3           0.15722    0.14849   1.059    0.295
## X4          -0.19527    0.16089  -1.214    0.231
## X5          -0.10383    0.16182  -0.642    0.524
## X6           0.75702    0.16886   4.483 4.43e-05 ***
## X7          -0.05262    0.13485  -0.390    0.698
## X8          -0.07924    0.15937  -0.497    0.621
## X9          -0.04086    0.14556  -0.281    0.780
## X10         -0.01868    0.13288  -0.141    0.889
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.033 on 49 degrees of freedom
## Multiple R-squared:  0.6715, Adjusted R-squared:  0.6045
## F-statistic: 10.02 on 10 and 49 DF,  p-value: 6.931e-09
```

Les variables sélectionnées sont les plus significatives, cad celles qui ont été utilisées pour construire Y.

n > 2p:

```
n = 15
p = 5

donnees <- data.frame(CreateData(p, n, rho))
modelgd <- lm(Y ~ ., data = donnees)
summary(modelgd)
```

```
##
## Call:
## lm(formula = Y ~ ., data = donnees)
##
## Residuals:
##        1        2        3        4        5        6        7        8
##  0.36480 -0.01438  0.15887 -0.62290 -1.13772 -0.09360 -0.15185 -0.64917
##        9       10       11       12       13       14       15
##  0.04825  0.10248  1.20398  0.57869 -0.41483  0.20867  0.41870
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.28325    0.91227  -0.310   0.7717
## X1           1.13974    0.57713   1.975   0.1195
## X2          -0.16009    0.38335  -0.418   0.6977
## X3          -0.44299    1.22614  -0.361   0.7362
## X4          -0.20001    0.47486  -0.421   0.6953
## X5           0.61939    0.97376   0.636   0.5593
## X6           1.15486    0.49109   2.352   0.0784 .
## X7          -0.16938    1.15351  -0.147   0.8904
## X8           0.14592    0.72156   0.202   0.8496
## X9           0.18090    0.88372   0.205   0.8478
## X10         -0.04335    0.51736  -0.084   0.9373
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.059 on 4 degrees of freedom
## Multiple R-squared:   0.91,  Adjusted R-squared:  0.6849
## F-statistic: 4.042 on 10 and 4 DF,  p-value: 0.09523
```

L'algorithme converge moins souvent qu'avant.

n ≤ 2p:

```
n = 5
p = 7

donnees <- data.frame(CreateData(p, n, rho))
modelgd <- lm(Y ~ ., data = donnees)
summary(modelgd)
```

```
##
## Call:
## lm(formula = Y ~ ., data = donnees)
##
## Residuals:
```

```
## ALL 5 residuals are 0: no residual degrees of freedom!
##
## Coefficients: (10 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -6.324         NA      NA       NA
## X1            -2.272         NA      NA       NA
## X2             6.179         NA      NA       NA
## X3            -4.808         NA      NA       NA
## X4             2.787         NA      NA       NA
## X5                NA         NA      NA       NA
## X6                NA         NA      NA       NA
## X7                NA         NA      NA       NA
## X8                NA         NA      NA       NA
## X9                NA         NA      NA       NA
## X10               NA         NA      NA       NA
## X11               NA         NA      NA       NA
## X12               NA         NA      NA       NA
## X13               NA         NA      NA       NA
## X14               NA         NA      NA       NA
##
## Residual standard error: NaN on 0 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:    NaN
## F-statistic:   NaN on 4 and 0 DF,  p-value: NA
```

L'algorithme ne peut converger.

3.

```
rho = 0.9
```

n >> p:

```
n = 60
p = 5


donnees <- data.frame(CreateData(p, n, rho))
modelgd <- lm(Y ~ ., data = donnees)
summary(modelgd)
```

```
##
## Call:
## lm(formula = Y ~ ., data = donnees)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4686 -0.8046  0.1133  0.5867  2.7700
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.47482    0.14573   3.258 0.002040 **
## X1           1.24762    0.42707   2.921 0.005257 **
## X2          -0.46922    0.60163  -0.780 0.439187
## X3           0.14594    0.46456   0.314 0.754749
## X4           0.43745    0.43944   0.995 0.324393
## X5          -0.37247    0.36559  -1.019 0.313291
```

```
## X6           1.14821    0.32283   3.557 0.000844 ***
## X7           0.21346    0.36666   0.582 0.563122
## X8          -0.46954    0.40138  -1.170 0.247743
## X9          -0.07504    0.43022  -0.174 0.862257
## X10          0.44014    0.42063   1.046 0.300525
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.011 on 49 degrees of freedom
## Multiple R-squared:  0.7517, Adjusted R-squared:  0.7011
## F-statistic: 14.84 on 10 and 49 DF,  p-value: 1.113e-11
```

Le modèle converge.

n > 2p:

```
n = 15
p = 5

donnees <- data.frame(CreateData(p, n, rho))
modelgd <- lm(Y ~ ., data = donnees)
summary(modelgd)
```

```
##
## Call:
## lm(formula = Y ~ ., data = donnees)
##
## Residuals:
##        1        2        3        4        5        6        7        8
## -0.65954  0.65499  0.45269 -0.42976 -0.68270  0.03892  1.04034  0.53775
##        9       10       11       12       13       14       15
## -0.50342  0.39413 -0.11139  0.15623 -0.25042 -0.11859 -0.51923
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.06286    0.39697  -0.158    0.882
## X1          -0.31733    1.57441  -0.202    0.850
## X2           0.14021    1.19940   0.117    0.913
## X3           1.19856    1.45691   0.823    0.457
## X4           0.33712    1.41285   0.239    0.823
## X5           0.41909    1.26762   0.331    0.758
## X6          -0.60901    1.54146  -0.395    0.713
## X7           0.58654    1.54752   0.379    0.724
## X8          -0.48235    1.05387  -0.458    0.671
## X9           1.20744    1.66226   0.726    0.508
## X10          0.07499    1.46239   0.051    0.962
##
## Residual standard error: 0.9855 on 4 degrees of freedom
## Multiple R-squared:  0.8223, Adjusted R-squared:  0.3781
## F-statistic: 1.851 on 10 and 4 DF,  p-value: 0.29
```

On retrouve parfois les bonnes variables après plusieurs essais, mais pas systématiquement, et on sélectionne parfois une des mauvaises variables en raison de la forte colinéarité.

n ≤ 2p:

```r
n = 5
p = 10

donnees <- data.frame(CreateData(p, n, rho))
modelgd <- lm(Y ~ ., data = donnees)
summary(modelgd)
```

```
##
## Call:
## lm(formula = Y ~ ., data = donnees)
##
## Residuals:
## ALL 5 residuals are 0: no residual degrees of freedom!
##
## Coefficients: (16 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.972         NA      NA       NA
## X1             3.864         NA      NA       NA
## X2            -0.973         NA      NA       NA
## X3             3.139         NA      NA       NA
## X4            -5.087         NA      NA       NA
## X5                NA         NA      NA       NA
## X6                NA         NA      NA       NA
## X7                NA         NA      NA       NA
## X8                NA         NA      NA       NA
## X9                NA         NA      NA       NA
## X10               NA         NA      NA       NA
## X11               NA         NA      NA       NA
## X12               NA         NA      NA       NA
## X13               NA         NA      NA       NA
## X14               NA         NA      NA       NA
## X15               NA         NA      NA       NA
## X16               NA         NA      NA       NA
## X17               NA         NA      NA       NA
## X18               NA         NA      NA       NA
## X19               NA         NA      NA       NA
## X20               NA         NA      NA       NA
##
## Residual standard error: NaN on 0 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:    NaN
## F-statistic:   NaN on 4 and 0 DF,  p-value: NA
```

L'algorithme ne peut converger.

Exercice 3 : Sélection de variables

1.

```r
chen <- read.table("chenilles.txt",header=TRUE)
attach(chen)
```

2.

```
modelchen <- lm(NbNids ~ ., data = chen)
summary(modelchen)
```

```
##
## Call:
## lm(formula = NbNids ~ ., data = chen)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.03941 -0.26272 -0.02351  0.21953  1.35140
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.561849   2.096950   4.083 0.000493 ***
## Altitude    -0.002956   0.001038  -2.847 0.009374 **
## Pente       -0.034821   0.014510  -2.400 0.025311 *
## NbPins       0.035385   0.066586   0.531 0.600454
## Hauteur     -0.501564   0.378701  -1.324 0.198955
## Diametre     0.108739   0.069495   1.565 0.131925
## Densite     -0.032715   1.044915  -0.031 0.975305
## Orient      -0.203959   0.669598  -0.305 0.763535
## HautMax      0.028180   0.157007   0.179 0.859201
## NbStrat     -0.862409   0.572133  -1.507 0.145945
## Melange     -0.448124   0.513764  -0.872 0.392499
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5493 on 22 degrees of freedom
## Multiple R-squared:  0.6809, Adjusted R-squared:  0.5359
## F-statistic: 4.695 on 10 and 22 DF,  p-value: 0.001203
```

Les variables **Altitude** et **Pente** sont les plus importantes.

3.

```
reschen <- lm(NbNids ~ Altitude + Pente + NbPins + Hauteur +
                       Diametre + Orient + NbStrat + Melange, data = chen)
summary(reschen)
```

```
##
## Call:
## lm(formula = NbNids ~ Altitude + Pente + NbPins + Hauteur + Diametre +
##     Orient + NbStrat + Melange, data = chen)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.04501 -0.25651 -0.00999  0.21317  1.34503
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.5361069  1.8216591   4.686 9.23e-05 ***
## Altitude    -0.0029815  0.0009775  -3.050  0.00551 **
## Pente       -0.0348705  0.0138695  -2.514  0.01904 *
```

```
## NbPins       0.0341444  0.0247372   1.380  0.18022
## Hauteur     -0.4646446  0.2813275  -1.652  0.11164
## Diametre     0.1072147  0.0629929   1.702  0.10167
## Orient      -0.2137894  0.5867429  -0.364  0.71878
## NbStrat     -0.8072234  0.4128418  -1.955  0.06229 .
## Melange     -0.4513521  0.4231607  -1.067  0.29676
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5263 on 24 degrees of freedom
## Multiple R-squared:  0.6804, Adjusted R-squared:  0.5739
## F-statistic: 6.387 on 8 and 24 DF,  p-value: 0.0001815
```

On constate que le R^2 (Adjusted R-squared) est plus grand pour le modèle réduit que pour le modèle complet, il semble donc meilleur, ce que confirme le test de Fisher suivant :

```
Ftest <- anova(reschen, modelchen)
Ftest
```

```
## Analysis of Variance Table
##
## Model 1: NbNids ~ Altitude + Pente + NbPins + Hauteur + Diametre + Orient +
##     NbStrat + Melange
## Model 2: NbNids ~ Altitude + Pente + NbPins + Hauteur + Diametre + Densite +
##     Orient + HautMax + NbStrat + Melange
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     24 6.6474
## 2     22 6.6369  2  0.010476 0.0174 0.9828
```

Ici, la p-valeur vaut 0.98, le test est donc non significatif : on ne rejette pas H_0, on garde le modèle réduit. En fait, pour trouver le meilleur modèle, il faudrait faire p! tests (ce qu'on ne fera évidemment pas).

4.

```
summary(modelchen)
```

```
##
## Call:
## lm(formula = NbNids ~ ., data = chen)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.03941 -0.26272 -0.02351  0.21953  1.35140
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.561849   2.096950   4.083 0.000493 ***
## Altitude    -0.002956   0.001038  -2.847 0.009374 **
## Pente       -0.034821   0.014510  -2.400 0.025311 *
## NbPins       0.035385   0.066586   0.531 0.600454
## Hauteur     -0.501564   0.378701  -1.324 0.198955
## Diametre     0.108739   0.069495   1.565 0.131925
## Densite     -0.032715   1.044915  -0.031 0.975305
## Orient      -0.203959   0.669598  -0.305 0.763535
## HautMax      0.028180   0.157007   0.179 0.859201
## NbStrat     -0.862409   0.572133  -1.507 0.145945
```

```
## Melange     -0.448124   0.513764  -0.872 0.392499
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5493 on 22 degrees of freedom
## Multiple R-squared:  0.6809, Adjusted R-squared:  0.5359
## F-statistic: 4.695 on 10 and 22 DF,  p-value: 0.001203
```

On enlève du modèle la variable qui a la p-valeur la plus grande : **Densite**.

```
m10 <- lm(NbNids ~ . - Densite, data = chen)
summary(m10)
```

```
##
## Call:
## lm(formula = NbNids ~ . - Densite, data = chen)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.03854 -0.25981 -0.02242  0.21970  1.35325
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.534150   1.859443   4.590 0.000129 ***
## Altitude    -0.002953   0.001009  -2.925 0.007616 **
## Pente       -0.034792   0.014163  -2.457 0.021999 *
## NbPins       0.033467   0.025506   1.312 0.202425
## Hauteur     -0.496385   0.333195  -1.490 0.149870
## Diametre     0.108048   0.064451   1.676 0.107196
## Orient      -0.212437   0.598947  -0.355 0.726057
## HautMax      0.026082   0.138865   0.188 0.852664
## NbStrat     -0.868023   0.531381  -1.634 0.115975
## Melange     -0.440130   0.436043  -1.009 0.323298
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5372 on 23 degrees of freedom
## Multiple R-squared:  0.6809, Adjusted R-squared:  0.556
## F-statistic: 5.453 on 9 and 23 DF,  p-value: 0.0004826
```

```
Ftest <- anova(modelchen, m10)
Ftest
```

```
## Analysis of Variance Table
##
## Model 1: NbNids ~ Altitude + Pente + NbPins + Hauteur + Diametre + Densite +
##     Orient + HautMax + NbStrat + Melange
## Model 2: NbNids ~ (Altitude + Pente + NbPins + Hauteur + Diametre + Densite +
##     Orient + HautMax + NbStrat + Melange) - Densite
##   Res.Df    RSS Df   Sum of Sq     F Pr(>F)
## 1     22 6.6369
## 2     23 6.6372 -1 -0.00029572 0.001 0.9753
```

Test non significatif, on préfère **m10**. On enlève du modèle **m10** la variable qui a la p-valeur la plus grande : **HautMax**.

```
m9 <- lm(NbNids ~ . -Densite -HautMax, data = chen)
summary(m9)
```

```
##
## Call:
## lm(formula = NbNids ~ . - Densite - HautMax, data = chen)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.04501 -0.25651 -0.00999  0.21317  1.34503
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.5361069  1.8216591   4.686 9.23e-05 ***
## Altitude    -0.0029815  0.0009775  -3.050  0.00551 **
## Pente       -0.0348705  0.0138695  -2.514  0.01904 *
## NbPins       0.0341444  0.0247372   1.380  0.18022
## Hauteur     -0.4646446  0.2813275  -1.652  0.11164
## Diametre     0.1072147  0.0629929   1.702  0.10167
## Orient      -0.2137894  0.5867429  -0.364  0.71878
## NbStrat     -0.8072234  0.4128418  -1.955  0.06229 .
## Melange     -0.4513521  0.4231607  -1.067  0.29676
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5263 on 24 degrees of freedom
## Multiple R-squared:  0.6804, Adjusted R-squared:  0.5739
## F-statistic: 6.387 on 8 and 24 DF,  p-value: 0.0001815
```

```
Ftest <- anova(m10, m9)
Ftest
```

```
## Analysis of Variance Table
##
## Model 1: NbNids ~ (Altitude + Pente + NbPins + Hauteur + Diametre + Densite +
##     Orient + HautMax + NbStrat + Melange) - Densite
## Model 2: NbNids ~ (Altitude + Pente + NbPins + Hauteur + Diametre + Densite +
##     Orient + HautMax + NbStrat + Melange) - Densite - HautMax
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     23 6.6372
## 2     24 6.6474 -1  -0.01018 0.0353 0.8527
```

Le test est non significatif, on préfère **m9**. On continue ainsi de suite jusqu'à **m5**.

```
m6 <- lm(NbNids ~ . -Densite -HautMax -Orient -Melange -NbPins, data = chen)
m5 <- lm(NbNids ~ . -Densite -HautMax -Orient -Melange -NbPins -NbStrat, data = chen)
summary(m5)
```

```
##
## Call:
## lm(formula = NbNids ~ . - Densite - HautMax - Orient - Melange -
##     NbPins - NbStrat, data = chen)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.98022 -0.35940 -0.08678  0.35270  1.20749
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.3120222  1.0245235   6.161 1.19e-06 ***
## Altitude    -0.0026538  0.0007899  -3.360  0.00227 **
## Pente       -0.0410588  0.0133439  -3.077  0.00464 **
## Hauteur     -0.7472203  0.2201267  -3.395  0.00207 **
## Diametre     0.1644422  0.0524864   3.133  0.00403 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5442 on 28 degrees of freedom
## Multiple R-squared:  0.6014, Adjusted R-squared:  0.5444
## F-statistic: 10.56 on 4 and 28 DF,  p-value: 2.409e-05
```

Tout est significatif, on s'arrête.

```
Ftest <- anova(m6, m5)
Ftest
```

```
## Analysis of Variance Table
##
## Model 1: NbNids ~ (Altitude + Pente + NbPins + Hauteur + Diametre + Densite +
##     Orient + HautMax + NbStrat + Melange) - Densite - HautMax -
##     Orient - Melange - NbPins
## Model 2: NbNids ~ (Altitude + Pente + NbPins + Hauteur + Diametre + Densite +
##     Orient + HautMax + NbStrat + Melange) - Densite - HautMax -
##     Orient - Melange - NbPins - NbStrat
##   Res.Df    RSS Df Sum of Sq     F  Pr(>F)
## 1     27 7.4381
## 2     28 8.2912 -1  -0.85317 3.097 0.08976 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Si on continue une fois de plus, le test deviendra significatif. Le modèle final retenu est le modèle **m5** pour lequel il ne reste que les variables **Altitude**, **Pente**, **Hauteur** et **Diametre**.

5.

```
step(modelchen, direction = "backward")
```

```
## Start:  AIC=-30.93
## NbNids ~ Altitude + Pente + NbPins + Hauteur + Diametre + Densite +
##     Orient + HautMax + NbStrat + Melange
##
##            Df Sum of Sq    RSS     AIC
## - Densite   1   0.00030 6.6372 -32.926
## - HautMax   1   0.00972 6.6466 -32.879
## - Orient    1   0.02799 6.6649 -32.788
## - NbPins    1   0.08520 6.7221 -32.506
## - Melange   1   0.22952 6.8664 -31.805
## <none>                  6.6369 -30.927
## - Hauteur   1   0.52918 7.1661 -30.396
## - NbStrat   1   0.68545 7.3224 -29.684
## - Diametre  1   0.73859 7.3755 -29.445
## - Pente     1   1.73726 8.3742 -25.255
```

```
## - Altitude  1   2.44545 9.0824 -22.576
##
## Step:  AIC=-32.93
## NbNids ~ Altitude + Pente + NbPins + Hauteur + Diametre + Orient +
##     HautMax + NbStrat + Melange
##
##            Df Sum of Sq    RSS     AIC
## - HautMax   1   0.01018 6.6474 -34.875
## - Orient    1   0.03630 6.6735 -34.746
## - Melange   1   0.29401 6.9312 -33.496
## <none>                  6.6372 -32.926
## - NbPins    1   0.49683 7.1340 -32.544
## - Hauteur   1   0.64047 7.2777 -31.886
## - NbStrat   1   0.77003 7.4073 -31.304
## - Diametre  1   0.81101 7.4482 -31.122
## - Pente     1   1.74141 8.3786 -27.237
## - Altitude  1   2.46888 9.1061 -24.490
##
## Step:  AIC=-34.88
## NbNids ~ Altitude + Pente + NbPins + Hauteur + Diametre + Orient +
##     NbStrat + Melange
##
##            Df Sum of Sq    RSS     AIC
## - Orient    1   0.03677 6.6842 -36.693
## - Melange   1   0.31511 6.9625 -35.347
## <none>                  6.6474 -34.875
## - NbPins    1   0.52769 7.1751 -34.354
## - Hauteur   1   0.75554 7.4029 -33.323
## - Diametre  1   0.80235 7.4498 -33.115
## - NbStrat   1   1.05891 7.7063 -31.997
## - Pente     1   1.75080 8.3982 -29.160
## - Altitude  1   2.57672 9.2241 -26.065
##
## Step:  AIC=-36.69
## NbNids ~ Altitude + Pente + NbPins + Hauteur + Diametre + NbStrat +
##     Melange
##
##            Df Sum of Sq    RSS     AIC
## - Melange   1   0.40168 7.0859 -36.767
## <none>                  6.6842 -36.693
## - NbPins    1   0.50781 7.1920 -36.277
## - NbStrat   1   1.02215 7.7063 -33.997
## - Hauteur   1   1.03389 7.7181 -33.947
## - Diametre  1   1.12553 7.8097 -33.558
## - Pente     1   1.71868 8.4029 -31.142
## - Altitude  1   2.98918 9.6734 -26.495
##
## Step:  AIC=-36.77
## NbNids ~ Altitude + Pente + NbPins + Hauteur + Diametre + NbStrat
##
##            Df Sum of Sq    RSS     AIC
## - NbPins    1   0.35221 7.4381 -37.167
## <none>                  7.0859 -36.767
## - Hauteur   1   0.85056 7.9364 -35.027
```

```
## - Diametre  1   0.99324 8.0791 -34.439
## - NbStrat   1   0.99727 8.0831 -34.422
## - Pente     1   1.82065 8.9065 -31.221
## - Altitude  1   2.62466 9.7105 -28.369
##
## Step:  AIC=-37.17
## NbNids ~ Altitude + Pente + Hauteur + Diametre + NbStrat
##
##            Df Sum of Sq    RSS     AIC
## <none>                  7.4381 -37.167
## - NbStrat   1   0.85317 8.2912 -35.583
## - Hauteur   1   1.21834 8.6564 -34.161
## - Diametre  1   1.37527 8.8133 -33.568
## - Pente     1   1.72426 9.1623 -32.286
## - Altitude  1   2.32266 9.7607 -30.199
```

```
##
## Call:
## lm(formula = NbNids ~ Altitude + Pente + Hauteur + Diametre +
##     NbStrat, data = chen)
##
## Coefficients:
## (Intercept)     Altitude        Pente      Hauteur     Diametre      NbStrat
##    5.998179    -0.002292    -0.033809    -0.521596     0.124145    -0.384935
```

On garde le modèle qui a la AIC-valeur la plus petite : -37.17.

```
m10 <- lm(NbNids ~ Altitude + Pente + Hauteur + Diametre + NbStrat, data = chen)
step(m10)
```

```
## Start:  AIC=-37.17
## NbNids ~ Altitude + Pente + Hauteur + Diametre + NbStrat
##
##            Df Sum of Sq    RSS     AIC
## <none>                  7.4381 -37.167
## - NbStrat   1   0.85317 8.2912 -35.583
## - Hauteur   1   1.21834 8.6564 -34.161
## - Diametre  1   1.37527 8.8133 -33.568
## - Pente     1   1.72426 9.1623 -32.286
## - Altitude  1   2.32266 9.7607 -30.199
```

```
##
## Call:
## lm(formula = NbNids ~ Altitude + Pente + Hauteur + Diametre +
##     NbStrat, data = chen)
##
## Coefficients:
## (Intercept)     Altitude        Pente      Hauteur     Diametre      NbStrat
##    5.998179    -0.002292    -0.033809    -0.521596     0.124145    -0.384935
```

Le modèle choisi est **m6** pour lequel la variable **NbStrat** a été ajoutée). Ce n'est donc pas le même modèle que plus haut.