TP II: Régression linéaire simple

Etude exploratoire

1. Créer dans `R` un objet nommé `cars` à partir du fichier de données `Cars.txt`. En tirer un objet nommé `cars.quantitative.vars` ne contenant que les variables quantitatives de `cars`.

```
cars <- read.table("Cars.txt", header=T)
summary(cars)
```

```
##    Country              Car                 MPG            Weight
##  Length:38          Length:38          Min.   :15.50   Min.   :1.915
##  Class :character   Class :character   1st Qu.:18.75   1st Qu.:2.208
##  Mode  :character   Mode  :character   Median :26.65   Median :2.685
##                                        Mean   :25.64   Mean   :2.863
##                                        3rd Qu.:30.80   3rd Qu.:3.410
##                                        Max.   :50.20   Max.   :4.360
##   Drive_Ratio      Horsepower      Displacement     Cylinders
##  Min.   :2.260   Min.   :  1.00   Min.   : 85.0   Min.   :4.000
##  1st Qu.:2.695   1st Qu.: 75.75   1st Qu.:101.2   1st Qu.:4.000
##  Median :3.080   Median : 97.00   Median :143.0   Median :4.500
##  Mean   :3.093   Mean   : 97.68   Mean   :170.7   Mean   :5.395
##  3rd Qu.:3.625   3rd Qu.:118.75   3rd Qu.:218.8   3rd Qu.:6.000
##  Max.   :3.900   Max.   :150.00   Max.   :360.0   Max.   :8.000
```

On ne conserve que les variables continues:

```
cars.quantitative.vars <- cars[c(-1,-2)]
summary(cars.quantitative.vars)
```

```
##       MPG            Weight       Drive_Ratio      Horsepower
##  Min.   :15.50   Min.   :1.915   Min.   :2.260   Min.   :  1.00
##  1st Qu.:18.75   1st Qu.:2.208   1st Qu.:2.695   1st Qu.: 75.75
##  Median :26.65   Median :2.685   Median :3.080   Median : 97.00
##  Mean   :25.64   Mean   :2.863   Mean   :3.093   Mean   : 97.68
##  3rd Qu.:30.80   3rd Qu.:3.410   3rd Qu.:3.625   3rd Qu.:118.75
##  Max.   :50.20   Max.   :4.360   Max.   :3.900   Max.   :150.00
##   Displacement     Cylinders
##  Min.   : 85.0   Min.   :4.000
##  1st Qu.:101.2   1st Qu.:4.000
##  Median :143.0   Median :4.500
##  Mean   :170.7   Mean   :5.395
##  3rd Qu.:218.8   3rd Qu.:6.000
##  Max.   :360.0   Max.   :8.000
```

2. Faire une étude exploratoire des variables de `cars.quantitative.vars`. Calculer les corrélations entre les variables continues. Que peut-on en conclure?

```
cor(cars.quantitative.vars,cars.quantitative.vars)
```

```
##                     MPG     Weight Drive_Ratio Horsepower Displacement
## MPG           1.0000000 -0.5265409   0.2759633 -0.9088438   -0.7242370
```

```
## Weight       -0.5265409  1.0000000  -0.6878798  0.5204863    0.8266357
## Drive_Ratio   0.2759633 -0.6878798   1.0000000 -0.4264688   -0.7796902
## Horsepower   -0.9088438  0.5204863  -0.4264688  1.0000000    0.7874087
## Displacement -0.7242370  0.8266357  -0.7796902  0.7874087    1.0000000
## Cylinders    -0.5008739  0.9166777  -0.6921504  0.5420376    0.8548706
##               Cylinders
## MPG          -0.5008739
## Weight        0.9166777
## Drive_Ratio  -0.6921504
## Horsepower    0.5420376
## Displacement  0.8548706
## Cylinders     1.0000000
```

3. Créer un sous-objet de `cars.quantitative.vars` nommé `carsntyp` dans lequel le véhicule `Buick Estate Wagon` est supprimé. Refaire la question précédente. Que se passe t-il? Pourquoi a-t-on enlevé ce véhicule?

On inspecte le véhicule `Buick Estate Wagon`

```
cars[1,]
```

```
##   Country                Car  MPG Weight Drive_Ratio Horsepower Displacement
## 1    U.S. Buick_Estate_Wagon 50.2   4.36        2.73          1          100
##   Cylinders
## 1         8
```

La puissance est `Horsepower` est à 1. Cela suggère une erreur dans la prise de donnée. Retirons la du jeu de données:

```
carsntyp <- cars.quantitative.vars[-1,]
summary(carsntyp)
```

```
##       MPG            Weight       Drive_Ratio      Horsepower
##  Min.   :15.50   Min.   :1.915   Min.   :2.260   Min.   : 65.0
##  1st Qu.:18.60   1st Qu.:2.200   1st Qu.:2.690   1st Qu.: 78.0
##  Median :26.50   Median :2.670   Median :3.080   Median : 97.0
##  Mean   :24.97   Mean   :2.822   Mean   :3.103   Mean   :100.3
##  3rd Qu.:30.50   3rd Qu.:3.410   3rd Qu.:3.640   3rd Qu.:120.0
##  Max.   :37.30   Max.   :4.054   Max.   :3.900   Max.   :150.0
##   Displacement     Cylinders
##  Min.   : 85.0   Min.   :4.000
##  1st Qu.:105.0   1st Qu.:4.000
##  Median :146.0   Median :4.000
##  Mean   :172.6   Mean   :5.324
##  3rd Qu.:225.0   3rd Qu.:6.000
##  Max.   :360.0   Max.   :8.000
```

```
cor(carsntyp,carsntyp)
```

```
##                     MPG     Weight Drive_Ratio Horsepower Displacement
## MPG           1.0000000 -0.9080722   0.4047299 -0.8712662   -0.7780733
## Weight       -0.9080722  1.0000000  -0.6958352  0.9062768    0.9449247
## Drive_Ratio   0.4047299 -0.6958352   1.0000000 -0.5875217   -0.8092459
## Horsepower   -0.8712662  0.9062768  -0.5875217  1.0000000    0.8562489
## Displacement -0.7780733  0.9449247  -0.8092459  0.8562489    1.0000000
## Cylinders    -0.7966059  0.9116385  -0.6908519  0.8524076    0.9361232
##               Cylinders
## MPG          -0.7966059
```

```
## Weight        0.9116385
## Drive_Ratio  -0.6908519
## Horsepower    0.8524076
## Displacement  0.9361232
## Cylinders     1.0000000
```

Influence d'un point atypique sur la modélisation

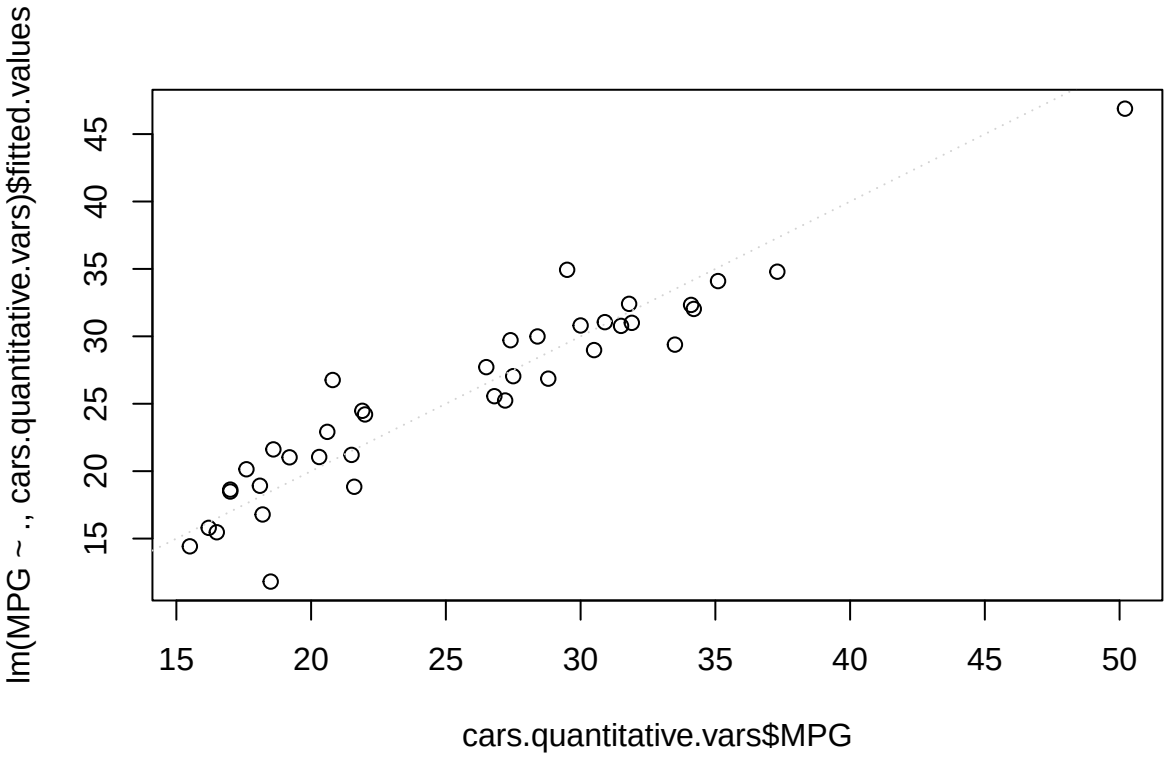
1. Estimer le modèle complet expliquant la variable `MPG` en fonction de toutes les autres variables de l'objet `cars.quantitative.vars`. Interpréter les résultats obtenus. Ce modèle vous semble-il satisfaisant? Quelles sont, à 5%, les variables significatives?

```
fit <- lm(MPG~., cars.quantitative.vars)
summary(fit)
```

```
##
## Call:
## lm(formula = MPG ~ ., data = cars.quantitative.vars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9602 -1.6212  0.3487  1.4918  6.6888
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  71.86000    6.01147  11.954 2.43e-13 ***
## Weight       -3.46883    1.60975  -2.155 0.038801 *
## Drive_Ratio  -6.24793    1.57509  -3.967 0.000385 ***
## Horsepower   -0.17498    0.03062  -5.714 2.49e-06 ***
## Displacement -0.04695    0.02012  -2.334 0.026037 *
## Cylinders     1.50956    0.78904   1.913 0.064711 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.682 on 32 degrees of freedom
## Multiple R-squared:  0.8926, Adjusted R-squared:  0.8758
## F-statistic: 53.17 on 5 and 32 DF,  p-value: 1.445e-14
```

On peut faire le tracer suivant pour comparer les valeurs observées et prédites:

```
plot(cars.quantitative.vars$MPG,lm(MPG~.,cars.quantitative.vars)$fitted.values)
abline(a=0, b=1, col = "lightgray", lty = 3)
```



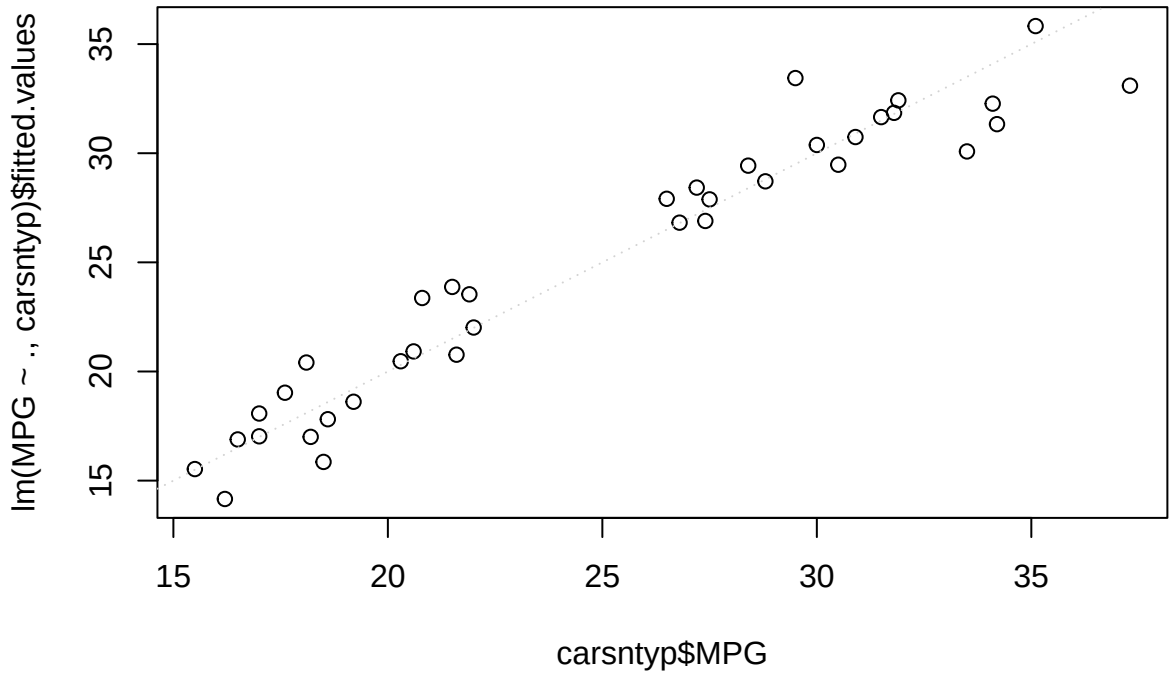
2. Estimer le modèle complet expliquant la variable `MPG` en fonction de toutes les autres variables de l'objet `carsntyp`. Interpréter les résultats obtenus. Ce modèle vous semble-il satisfaisant? Quelles sont, à 5%, les variables significatives?

```
fit.carsntyp <- lm(MPG~.,carsntyp)
summary(fit.carsntyp)
```

```
##
## Call:
## lm(formula = MPG ~ ., data = carsntyp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9430 -1.0312 -0.0493  0.7874  4.2056
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   73.50042    4.06369  18.087  < 2e-16 ***
## Weight       -11.81499    1.71780  -6.878 1.04e-07 ***
## Drive_Ratio   -4.35077    1.10477  -3.938 0.000434 ***
## Horsepower    -0.04709    0.02903  -1.622 0.114904
## Displacement   0.02043    0.01731   1.180 0.246853
## Cylinders     -0.09069    0.59029  -0.154 0.878888
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.809 on 31 degrees of freedom
## Multiple R-squared:  0.9334, Adjusted R-squared:  0.9226
## F-statistic: 86.85 on 5 and 31 DF,  p-value: < 2.2e-16
```

On peut faire le tracer suivant pour comparer les valeurs observées et prédites:

```
plot(carsntyp$MPG,lm(MPG~.,carsntyp)$fitted.values)
abline(a=0, b=1, col = "lightgray", lty = 3)
```



Sélection de variables

1. Voici une procédure de choix de modèle à la main par élimination. On partira du modèle complet:
   - Identifier la variable explicative pour laquelle le test de Student est le moins significatif (i.e. plus grande p-value).
   - La retirer du modèle et relancer l'estimation.

   On itérera ces 2 étapes jusqu'à ce que tous les coefficients soient significatifs à 5%. Attention! La variable constante est généralement conservée dans tous les modèles.

```
fit <- lm(MPG~., cars.quantitative.vars)
summary(fit)
```

```
##
## Call:
## lm(formula = MPG ~ ., data = cars.quantitative.vars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9602 -1.6212  0.3487  1.4918  6.6888
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  71.86000    6.01147  11.954 2.43e-13 ***
## Weight       -3.46883    1.60975  -2.155 0.038801 *
## Drive_Ratio  -6.24793    1.57509  -3.967 0.000385 ***
## Horsepower   -0.17498    0.03062  -5.714 2.49e-06 ***
## Displacement -0.04695    0.02012  -2.334 0.026037 *
## Cylinders     1.50956    0.78904   1.913 0.064711 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 2.682 on 32 degrees of freedom
## Multiple R-squared:  0.8926, Adjusted R-squared:  0.8758
## F-statistic: 53.17 on 5 and 32 DF,  p-value: 1.445e-14
```

La variable la moins explicative est `Cylinders`. On la retire donc du modèle et on recommence

```
fit <- lm(MPG~. - Cylinders, cars.quantitative.vars)
summary(fit)
```

```
##
## Call:
## lm(formula = MPG ~ . - Cylinders, data = cars.quantitative.vars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.1324 -1.4565 -0.3325  1.7487  6.3710
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  71.51976    6.24634  11.450 4.91e-13 ***
## Weight       -1.41462    1.24674  -1.135  0.26470
## Drive_Ratio  -5.83490    1.62189  -3.598  0.00104 **
## Horsepower   -0.18988    0.03079  -6.167 5.90e-07 ***
## Displacement -0.03067    0.01895  -1.618  0.11508
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.788 on 33 degrees of freedom
## Multiple R-squared:  0.8803, Adjusted R-squared:  0.8658
## F-statistic: 60.66 on 4 and 33 DF,  p-value: 9.561e-15
```

On se rend compte alors que `Weight` et `displacement` ne sont plus significatives. On recommence, on enlève alors, la variable avec la plus grande p-value à savoir `Weight`

```
fit <- lm(MPG~. - Cylinders - Weight , cars.quantitative.vars)
summary(fit)
```

```
##
## Call:
## lm(formula = MPG ~ . - Cylinders - Weight, data = cars.quantitative.vars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.8592 -1.6897 -0.2622  1.9698  6.7216
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  69.01406    5.86759  11.762 1.56e-13 ***
## Drive_Ratio  -5.97017    1.62433  -3.675 0.000812 ***
## Horsepower   -0.17715    0.02879  -6.153 5.46e-07 ***
## Displacement -0.04455    0.01453  -3.066 0.004236 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.8 on 34 degrees of freedom
```

```
## Multiple R-squared:  0.8756, Adjusted R-squared:  0.8646
## F-statistic: 79.78 on 3 and 34 DF,  p-value: 1.824e-15
```

On s'arrête alors, toutes les variables étant significatives à 5%.

2. La procédure `step` permet également de chercher le meilleur modèle de régression. Tester les différentes options et comparer avec le modèle obtenu "à la main".

```
model <- step(lm(MPG~.,cars.quantitative.vars))
```

```
## Start:  AIC=80.44
## MPG ~ Weight + Drive_Ratio + Horsepower + Displacement + Cylinders
##
##                Df Sum of Sq    RSS     AIC
## <none>                      230.14  80.442
## - Cylinders     1    26.324 256.47  82.558
## - Weight        1    33.396 263.54  83.591
## - Displacement  1    39.178 269.32  84.416
## - Drive_Ratio   1   113.164 343.31  93.639
## - Horsepower    1   234.794 464.94 105.164
```

```
summary(model)
```

```
##
## Call:
## lm(formula = MPG ~ Weight + Drive_Ratio + Horsepower + Displacement +
##     Cylinders, data = cars.quantitative.vars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9602 -1.6212  0.3487  1.4918  6.6888
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  71.86000    6.01147  11.954 2.43e-13 ***
## Weight       -3.46883    1.60975  -2.155 0.038801 *
## Drive_Ratio  -6.24793    1.57509  -3.967 0.000385 ***
## Horsepower   -0.17498    0.03062  -5.714 2.49e-06 ***
## Displacement -0.04695    0.02012  -2.334 0.026037 *
## Cylinders     1.50956    0.78904   1.913 0.064711 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.682 on 32 degrees of freedom
## Multiple R-squared:  0.8926, Adjusted R-squared:  0.8758
## F-statistic: 53.17 on 5 and 32 DF,  p-value: 1.445e-14
```

On doit alors spécifier un argument à la fonction step, à savoir direction. Les arguments sont `both` (ascendant et descendant), `backward` (descendant) et enfin, `forward` (ascendant). On recommence alors :

```
extractAIC(lm(MPG~1 + Weight + Drive_Ratio + Horsepower + Displacement + Cylinders,carsntyp))
```

```
## [1]  6.00000 49.32266
```

```
extractAIC(lm(MPG~.-Cylinders,carsntyp))
```

```
## [1]  5.00000 47.35082
```

```
step(lm(MPG~.,carsntyp), direction= "backward" )
```

```
## Start:  AIC=49.32
## MPG ~ Weight + Drive_Ratio + Horsepower + Displacement + Cylinders
##
##                Df Sum of Sq    RSS    AIC
## - Cylinders     1     0.077 101.53 47.351
## - Displacement  1     4.559 106.02 48.949
## <none>                      101.46 49.323
## - Horsepower    1     8.612 110.07 50.337
## - Drive_Ratio   1    50.758 152.22 62.332
## - Weight        1   154.826 256.28 81.609
##
## Step:  AIC=47.35
## MPG ~ Weight + Drive_Ratio + Horsepower + Displacement
##
##                Df Sum of Sq    RSS    AIC
## <none>                      101.53 47.351
## - Displacement  1     5.769 107.30 47.396
## - Horsepower    1     9.021 110.56 48.500
## - Drive_Ratio   1    54.787 156.32 61.317
## - Weight        1   155.457 256.99 79.711
```

```
##
## Call:
## lm(formula = MPG ~ Weight + Drive_Ratio + Horsepower + Displacement,
##     data = carsntyp)
##
## Coefficients:
##  (Intercept)        Weight   Drive_Ratio    Horsepower  Displacement
##     73.49966     -11.82693      -4.39104      -0.04772       0.01892
```

```
step(lm(MPG~.,carsntyp), MPG~1, direction= "backward" )
```

```
## Start:  AIC=49.32
## MPG ~ Weight + Drive_Ratio + Horsepower + Displacement + Cylinders
##
##                Df Sum of Sq    RSS    AIC
## - Cylinders     1     0.077 101.53 47.351
## - Displacement  1     4.559 106.02 48.949
## <none>                      101.46 49.323
## - Horsepower    1     8.612 110.07 50.337
## - Drive_Ratio   1    50.758 152.22 62.332
## - Weight        1   154.826 256.28 81.609
##
## Step:  AIC=47.35
## MPG ~ Weight + Drive_Ratio + Horsepower + Displacement
##
##                Df Sum of Sq    RSS    AIC
## <none>                      101.53 47.351
## - Displacement  1     5.769 107.30 47.396
## - Horsepower    1     9.021 110.56 48.500
## - Drive_Ratio   1    54.787 156.32 61.317
## - Weight        1   155.457 256.99 79.711
```

```
##
## Call:
## lm(formula = MPG ~ Weight + Drive_Ratio + Horsepower + Displacement,
##     data = carsntyp)
##
## Coefficients:
##  (Intercept)        Weight   Drive_Ratio    Horsepower  Displacement
##     73.49966     -11.82693      -4.39104      -0.04772       0.01892
```

```
extractAIC(lm(MPG~1,cars.quantitative.vars))
```

```
## [1]   1.0000 155.2177
```

```
step(lm(MPG~1,cars.quantitative.vars),MPG~Weight + Drive_Ratio + Horsepower + Displacement + Cylinders
```

```
## Start:  AIC=155.22
## MPG ~ 1
##
##                Df Sum of Sq     RSS     AIC
## + Horsepower    1   1769.52  372.76  90.768
## + Displacement  1   1123.67 1018.62 128.967
## + Weight        1    593.94 1548.35 144.880
## + Cylinders     1    537.45 1604.84 146.241
## + Drive_Ratio   1    163.15 1979.14 154.208
## <none>                      2142.29 155.218
##
## Step:  AIC=90.77
## MPG ~ Horsepower
##
##                Df Sum of Sq    RSS    AIC
## + Drive_Ratio   1    32.630 340.13 89.287
## <none>                      372.76 90.768
## + Weight        1     8.410 364.35 91.901
## + Displacement  1     0.417 372.35 92.725
## + Cylinders     1     0.206 372.56 92.747
##
## Step:  AIC=89.29
## MPG ~ Horsepower + Drive_Ratio
##
##                Df Sum of Sq    RSS    AIC
## + Displacement  1    73.663 266.47 82.012
## + Weight        1    63.312 276.82 83.460
## + Cylinders     1    24.290 315.84 88.471
## <none>                      340.13 89.287
##
## Step:  AIC=82.01
## MPG ~ Horsepower + Drive_Ratio + Displacement
##
##             Df Sum of Sq    RSS    AIC
## <none>                    266.47 82.012
## + Weight     1   10.0055 256.47 82.558
## + Cylinders  1    2.9334 263.54 83.591
```

```
##
## Call:
## lm(formula = MPG ~ Horsepower + Drive_Ratio + Displacement, data = cars.quantitative.vars)
```

```
##
## Coefficients:
##  (Intercept)    Horsepower   Drive_Ratio  Displacement
##     69.01406      -0.17715      -5.97017      -0.04455
```

3. Ici, le nombre de variables explicatives n'étant pas trop important, on peut également faire une recherche exhaustive parmi tous les modèles possibles. Tester cela en choisissant différents critères de sélection.

```
## Function to get statistics for one fit
get_stats_fit <- function(fit) {
  sumfit <- summary(fit)
  res <- data.frame(AIC = extractAIC(fit)[2],
                    BIC = BIC(fit),
                    R2 = sumfit$r.squared)
  return(res)
}

## Function to get statistics for one set of predictors
get_stats_pred <- function(preds, y, data) {
  fit <- lm(reformulate(preds, y), data = data)
  res <- get_stats_fit(fit)
  res$preds <- paste(preds, collapse = "+")
  return(res)
}

## Function to get all statistics
get_all_stats <- function(data_sub) {
  all_preds <- colnames(data_sub)[-1]
  y <- colnames(data_sub)[1]
  all_combs <- c(combn(all_preds, 1, get_stats_pred, simplify = FALSE, y = y, data = data_sub),
                 combn(all_preds, 2, get_stats_pred, simplify = FALSE, y = y, data = data_sub),
                 combn(all_preds, 3, get_stats_pred, simplify = FALSE, y = y, data = data_sub),
                 combn(all_preds, 4, get_stats_pred, simplify = FALSE, y = y, data = data_sub),
                 combn(all_preds, 5, get_stats_pred, simplify = FALSE, y = y, data = data_sub))
  res <- do.call(rbind, all_combs)
  return(res)
}
```

```
## Fit all possible models
model_perf <- get_all_stats(carsntyp)
model_perf
```

```
##          AIC      BIC        R2
## 1   77.13494 188.9691 0.8245951
## 2  134.92019 246.7544 0.1638063
## 3   88.87375 200.7080 0.7591048
## 4  107.13383 218.9680 0.6053980
## 5  104.29101 216.1252 0.6345810
## 6   47.88846 161.3336 0.9246158
## 7   76.27248 189.7176 0.8376535
## 8   63.73214 177.1773 0.8843226
## 9   77.89666 191.3418 0.8303682
## 10  88.07728 201.5224 0.7766407
## 11  91.95148 205.3966 0.7519849
## 12 101.93790 215.3830 0.6751404
## 13  90.27761 203.7227 0.7629550
```

```
## 14  89.20251 202.6476 0.7697436
## 15 105.42402 218.8691 0.6430440
## 16  47.39554 162.4516 0.9295276
## 17  48.50016 163.5562 0.9273919
## 18  49.76675 164.8228 0.9248634
## 19  61.31684 176.3729 0.8973349
## 20  76.18811 191.2441 0.8465463
## 21  64.04453 179.1006 0.8894803
## 22  79.71063 194.7667 0.8312190
## 23  84.13695 199.1930 0.8097703
## 24  93.11639 208.1724 0.7575199
## 25  91.09470 206.1507 0.7704136
## 26  47.35082 164.0178 0.9333164
## 27  48.94916 165.6161 0.9303727
## 28  50.33704 167.0040 0.9277114
## 29  62.33212 178.9991 0.9000312
## 30  81.60853 198.2755 0.8316841
## 31  49.32266 167.6005 0.9333672
##                                                    preds
## 1                                                 Weight
## 2                                             Drive_Ratio
## 3                                              Horsepower
## 4                                            Displacement
## 5                                               Cylinders
## 6                                       Weight+Drive_Ratio
## 7                                        Weight+Horsepower
## 8                                      Weight+Displacement
## 9                                         Weight+Cylinders
## 10                                 Drive_Ratio+Horsepower
## 11                               Drive_Ratio+Displacement
## 12                                  Drive_Ratio+Cylinders
## 13                                Horsepower+Displacement
## 14                                   Horsepower+Cylinders
## 15                                 Displacement+Cylinders
## 16                           Weight+Drive_Ratio+Horsepower
## 17                         Weight+Drive_Ratio+Displacement
## 18                            Weight+Drive_Ratio+Cylinders
## 19                          Weight+Horsepower+Displacement
## 20                             Weight+Horsepower+Cylinders
## 21                           Weight+Displacement+Cylinders
## 22                     Drive_Ratio+Horsepower+Displacement
## 23                        Drive_Ratio+Horsepower+Cylinders
## 24                      Drive_Ratio+Displacement+Cylinders
## 25                       Horsepower+Displacement+Cylinders
## 26             Weight+Drive_Ratio+Horsepower+Displacement
## 27                Weight+Drive_Ratio+Horsepower+Cylinders
## 28              Weight+Drive_Ratio+Displacement+Cylinders
## 29               Weight+Horsepower+Displacement+Cylinders
## 30          Drive_Ratio+Horsepower+Displacement+Cylinders
## 31 Weight+Drive_Ratio+Horsepower+Displacement+Cylinders
```

```
model_perf[c(which.min(model_perf$AIC), which.min(model_perf$BIC), which.max(model_perf$R2)),]
```

```
##         AIC      BIC        R2
## 26 47.35082 164.0178 0.9333164
```

```
## 6  47.88846 161.3336 0.9246158
## 31 49.32266 167.6005 0.9333672
##                                                    preds
## 26           Weight+Drive_Ratio+Horsepower+Displacement
## 6                                    Weight+Drive_Ratio
## 31 Weight+Drive_Ratio+Horsepower+Displacement+Cylinders
```

Prédiction

A partir du modèle complet, puis du meilleur modèle obtenu selon le critère BIC, prédire le `MPG` des véhicules dont les caractéristiques sont les suivantes :

Le meilleur modèle selon le critère BIC

```
## Best model according to BIC
fit_BIC <- lm(MPG ~ Weight + Drive_Ratio, carsntyp)
summary(fit_BIC)
```

```
##
## Call:
## lm(formula = MPG ~ Weight + Drive_Ratio, data = carsntyp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8631 -1.3746  0.2042  0.6539  4.1528
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  75.2709     4.0123  18.760  < 2e-16 ***
## Weight      -11.7794     0.6359 -18.524  < 2e-16 ***
## Drive_Ratio  -5.4947     0.8181  -6.717 1.03e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.837 on 34 degrees of freedom
## Multiple R-squared:  0.9246, Adjusted R-squared:  0.9202
## F-statistic: 208.5 on 2 and 34 DF,  p-value: < 2.2e-16
```

```
## predict values on new data
predict(fit_BIC, data.frame(Country = c("US", "France", "Germany", "Japan" ),
                            Car = c("Pontiac", "CitroenC3", "AudiA3", "ToyotaCorona"),
                            Weight = c(3.654, 2.99, 3.22, 4.001),
                            Drive_Ratio = c(3.064, 3.101, 2.885, 3.965) ))
```

```
##         1         2         3         4
## 15.393237 23.011454 21.489048  6.355057
```