Lecture 2 : Model estimation/validation through R example

M2-Modèles pour la régression

K. Meziani

Dauphine | PSL

Packages

```
library(MASS)
library(car)
library(carData)
library(knitr) ~> for joli tableau
library(ggplot2)
library(caret)
library(cowplot)
library(reshape2)
library(mlbench)
library(GGally)
library(corrplot)
library(questionr)
library(multcomp)
library(dplyr)
```

Study case

- ☛ The goal of this lecture is to implement the statistical tools reviewed thus far on a real dataset.
- ☛ Goals: - Estimate and validate a model. - Detect atypical points.

Section 1

1. The data

Car consumption dataset

Explain and predict gas consumption (in liters per 100 km) of different automobile models based on the following variables:

- `Type` = Type of the vehicle.
- `Consommation` = Fuel consumption in liters per 100 km.
- `Prix` = Vehicle price in Swiss francs.
- `Cylindree` = Cylinder capacity in cm3.
- `Puissance` = Power in kW.
- `Poids` = Weight in kg.

Response variable Y is `Consommation`. The covariates X_j correspond to the other 4 variables.

Dataset

Download dataset "conso.txt"

```
conso_voit = read.table("conso.txt", header=TRUE, sep="\t", dec=","
                         ,row.names=1)
conso_voit_complet = read.table("conso.txt", header=TRUE, sep="\t", dec=",")
```

row.names = 1 , première colonne non prise en compte

We have $n = 31$ observations of 5

```
dim(conso_voit_complet)
```

```
## [1] 31  6
```

Print the names of the variables

```
names(conso_voit_complet)
```

```
## [1] "Type"          "Prix"          "Cylindree"     "Puissance"     "Poi
## [6] "Consommation"
```

Display the head of the dataset in a table

```
kable(head(conso_voit_complet))
```

↳ pour joli tableau. si besoin knitr::kable. (kable existe dans plusieurs packages)

Type	Prix	Cylindree	Puissance	Poids	Consommation
Daihatsu Cuore	11600	846	32	650	5.7
Suzuki Swift 1.0 GLS	12490	993	39	790	5.8
Fiat Panda Mambo L	10450	899	29	730	6.1
VW Polo 1.4 60	17140	1390	44	955	6.5
Opel Corsa 1.2i Eco	14825	1195	33	895	6.8
Subaru Vivio 4WD	13730	658	32	740	6.8

Check the type of the variable

The command `str` allows to check whether the type of each variable is well specified. Here, there is no mistake.

```
str(conso_voit_complet)
```

il faudrait factoriser si on voulait s'intéresser au type

```
## 'data.frame':    31 obs. of  6 variables:
##  $ Type        : chr  "Daihatsu Cuore" "Suzuki Swift 1.0 GLS" "Fiat 
##  $ Prix        : int  11600 12490 10450 17140 14825 13730 19490 2850
##  $ Cylindree   : int  846 993 899 1390 1195 658 1331 5474 5987 2789 
##  $ Puissance   : int  32 39 29 44 33 32 55 325 300 209 ...
##  $ Poids       : int  650 790 730 955 895 740 1010 1690 2250 1485 ..
##  $ Consommation: num  5.7 5.8 6.1 6.5 6.8 6.8 7.1 21.3 18.7 14.5 ...
```

⚠ des facteurs 0,1 peuvent être considérés comme des int ou num.

Descriptive dataset analysis

Provides elementary statistics (average, quantiles, ...)

```
summary(conso_voit)
```

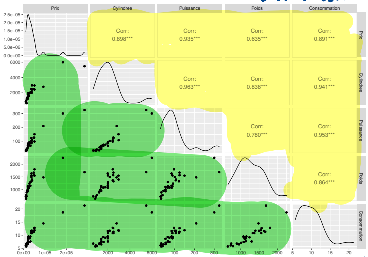
```
##       Prix           Cylindree       Puissance           Poids     
##  Min.   : 10450   Min.   : 658   Min.   : 29.0   Min.   : 650  
##  1st Qu.: 19820   1st Qu.:1390   1st Qu.: 55.0   1st Qu.:1042  
##  Median : 28750   Median :1984   Median : 85.0   Median :1155  
##  Mean   : 43756   Mean   :2094   Mean   : 97.1   Mean   :1256  
##  3rd Qu.: 39395   3rd Qu.:2456   3rd Qu.:106.5   3rd Qu.:1525  
##  Max.   :285000   Max.   :5987   Max.   :325.0   Max.   :2250  
##   Consommation  
##  Min.   : 5.700  
##  1st Qu.: 7.250  
##  Median : 9.300  
##  Mean   : 9.955  
##  3rd Qu.:11.650  
##  Max.   :21.300  
```

Visualize the correlation with `library(GGally)`

library(GGally)

```
ggpairs(conso_voit)
```

corrélation



Plot des covariables en fonctions des autres

densité estimée

Interpretation:

- Possible linear relation between `Consommation` and the others covariates.
- Diagonal plot: estimated density of each variable.
- Above the diagonal, correlations between variables are computed.

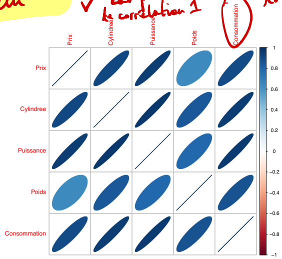
Visualize the correlation with `library(corrplot)`

```
r=round(cor(conso_voit),2); corrplot(r,method="ellipse")
```

On voit tout de suite si la matrice design est de rang plein

✓ cor pas de corrélation 1

→ corrélée avec tout le monde



↳ ici elle aura sans doute des petites VP car bcp de corrélation

Interpretation:

- "circle" : uncorrelated ⟶ "line": linear linear corrélation
- "red" : correlation "-" ⟶ "blue" : correlation "+"
- "clear" : weak correlation ⟶ "dark" : strong correlation

FORME

corrélation ↗ cercle augmente

corrélation +

−

COULEUR

foncé → très cor

clair → peu cor

bleu → cor +

rouge → cor −

Descriptive dataset analysis conclusions

- The `Consommation` is well-correlated with the 4 others variables.
- The covariates `Cylindree` and `Puissance` are strongly correlated.

Section 2

2. The gaussian regression linear model

Linear regression model with `lm`

$$Y = \beta_0 \mathbf{1}_n + \beta_1 X_1 + \cdots + \beta_1 X_4 + \mathbf{E} = X\beta + \varepsilon,$$

where $\beta := (\beta_0, \beta_1, \cdots, \beta_4)^\top$, $\varepsilon \sim \mathcal{N}(\mathbf{0}_n, \sigma^2 \boldsymbol{I}_n)$ (**[P1]–[P4]** statisfied).

- By default **R** adds an intercept (a column of 1). Here, the design matrix X is a matrix of size $n \times p$ with $p = 5$.
- The order in which variables are entered gives the indice j of the regressor X_j.

```
reg = lm(Consommation~Prix+Cylindree+Puissance+Poids,data=conso_voit)
```

- An easy way to declare the model is with "." and **R** takes into account all the variables.

Si j'écris .2, il prend aussi toutes les formes quadratiques.

```
reg = lm(Consommation~.,data=conso_voit)
```

Visualize the results using the function `summary`

```
summary(reg)
```

```
##
## Call:
## lm(formula = Consommation ~ ., data = conso_voit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5677 -0.6704  0.1183  0.5283  1.4361
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.456e+00  6.268e-01   3.919 0.000578 ***
## Prix         2.042e-05  8.731e-06   2.339 0.027297 *
## Cylindree   -5.006e-04  5.748e-04  -0.871 0.391797
## Puissance    2.499e-02  9.992e-03   2.501 0.018993 *
## Poids        4.161e-03  8.788e-04   4.734 6.77e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8172 on 26 degrees of freedom
## Multiple R-squared:  0.9546, Adjusted R-squared:  0.9476
## F-statistic: 136.5 on 4 and 26 DF,  p-value: < 2.2e-16
```

$\in [-2;2]$ ok

➤ `Call` : A reminder of the formula used.

Estimation of β

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.456e+00  6.268e-01   3.919 0.000578 ***
Prix         2.042e-05  8.731e-06   2.339 0.027297 *
Cylindree   -5.006e-04  5.748e-04  -0.871 0.391797
Puissance    2.499e-02  9.992e-03   2.501 0.018993 *
Poids        4.161e-03  8.788e-04   4.734 6.77e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- **`Estimate`** : The value of $\widehat{\beta}_j$ the least square estimator $\widehat{\beta}$ (which is the maximum likelihood estimator under **[P1]–[P4]**).
- **`Std. Error`** : The value of $\widehat{\sigma}_j = \sqrt{\widehat{\mathbb{V}\mathrm{ar}}_\beta(\widehat{\beta}_j)}$, estimator of the standard deviation of $\widehat{\beta}_j$.

Test of the regressors

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.456e+00  6.268e-01   3.919 0.000578 ***
Prix         2.042e-05  8.731e-06   2.339 0.027297 *
Cylindree   -5.006e-04  5.748e-04  -0.871 0.391797
Puissance    2.499e-02  9.992e-03   2.501 0.018993 *
Poids        4.161e-03  8.788e-04   4.734 6.77e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

si < 0,05 on rejette $H_0 : \beta_j = 0$
sinon, on ne peut pas rejeter H_0

- **`t value`** : Here we test $\mathcal{H}_0 : \beta_j = 0$ vs $H_1 : \beta_j \neq 0$. The `t value` is the value of the Student test statistic T, such that under $\mathcal{H}_0$

$$T = \frac{\widehat{\beta}_j}{\widehat{\sigma}\sqrt{(X^T X)^{-1}_{jj}}} \sim t_{n-p}$$

- **`Pr(>|t|)`** : The *p-value* of the previous Student tests. $n - r$ ddl.
- **Attention:** Nonreject $\mathcal{H}_0 \neq$ accept $\mathcal{H}_0$.

Estimation of σ^2

```
Residual standard error: 0.8172 on 26 degrees of freedom
```

- `Residual standard error` : The value of $\widehat{\sigma}$.
- `0.8172` : The value of $\widehat{\sigma}$ and Here

$$\widehat{\sigma}^2 = 0.8172^2$$

- `on 26 degrees of freedom` : dthe number of degrees of freedom : $(n-p)$ (here $31-5=26$) of the chi2 law follow by $(n-r)\widehat{\sigma}^2/\sigma^2$.

Test of the model :Global Fisher test

```
F-statistic: 136.5 on 4 and 26 DF,  p-value: < 2.2e-16
```

- `F-statistic: 136.5` : the value of the Fisher's global test statistic $F = 136.5$ $s.t.$ under $\mathcal{H}_0$

$$F = \frac{(\text{RSS}_0 - \text{RSS})/(p-1)}{\text{RSS}/(n-p)} = \frac{\|P_X Y - \overline{Y}\mathbf{1}\|^2/(p-1)}{\widehat{\sigma}^2} \sim F_{(p-1,n-p)}.$$

$$\text{where } \mathcal{H}_0 : Y_i = \beta_0 + \varepsilon_i \quad vs\ \mathcal{H}_1 : Y_i = \beta_0 + \sum_{j=1}^{4} \beta_j X_{ij} + \varepsilon_i$$

- `on 4 and 26 DF` : associated degrees of freedom $(p-1, n-p) = (4, 26)$.
- `p-value: < 2.2e-16` : so we reject H_0, meaningful test.

Model reduce to the intercept is $Y = \beta_0 \mathbf{1}_n + \varepsilon$

```r
reg0=lm(Consommation~1,data=conso_voit)
```

Coefficients R^2 and R_a^2

$$\frac{\text{variabilité expliquée par le modèle}}{\text{variabilité totale}}$$

```
Multiple R-squared:  0.9546,    Adjusted R-squared:  0.9476
```

- `Multiple R-squared:  0.9546` : The value of R^2.
- `Adjusted R-squared:  0.9476` : The value of R_a^2.

RSS , TSS et MSS

```
anova(reg0,reg)
```

(test de Fischer emboité)

2 modèles emboités, sous H_0 petit modèle, sous H_1 grand modèle.

```
## Analysis of Variance Table
##
## Model 1: Consommation ~ 1
## Model 2: Consommation ~ Prix + Cylindree + Puissance + Poids
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1     30 382.14
## 2     26  17.36  4    364.77 136.54 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Here: RSS $= 17.36$, TSS $= 364.77$ and MSS $= 382.14$ *s.t.*

$$\text{TSS} = \text{RSS} + \text{MSS}$$

Estimateds residuals

```
Residuals:
    Min      1Q  Median      3Q     Max
-1.5677 -0.6704  0.1183  0.5283  1.4361
```

- `Residuals` : A summary descriptive analysis of residues $\widehat{\varepsilon}_i$.

Section 3

3. Model validation

Model validation

We assume that X is full rank and the following postulates

[P1] Errors are centered/(the model is linear) : $\forall i = 1, \cdots, n \quad \mathbb{E}_\beta[\varepsilon_i] = 0$.

[P2] Errors have homoscedastic variance :
$\forall i = 1, \cdots, n \quad \mathbb{V}\mathrm{ar}_\beta[\varepsilon_i] = \sigma^2 > 0$.

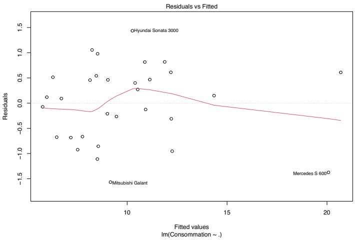
[P3] Errors are uncorrelated: $\forall i \neq j \quad \mathbb{C}\mathrm{ov}(\varepsilon_i, \varepsilon_j)$.

[P4] Errors are gaussian : $\forall i = 1, \cdots, n \quad \varepsilon_i \sim \mathcal{N}(0, \sigma^2)$.

- The rank hypothesis is easily verifiable. $\longrightarrow$ corrplot
- Checking the postulates requires an analysis of the residues.

[P1] Errors are centered

```r
plot(reg,1)
```



plutôt plate → faible échelle
→ peu d'observations

✓

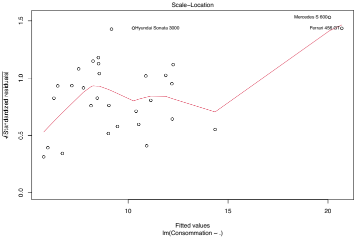
Interpretation:

- **[P1]** can be assesed by inspecting the *Residuals vs Fitted*-plot.
- Residues appear (reasonably) uniformly distributed around 0 (red line is approximately horizontal at 0).

[P2] homoscedasticity

```
plot(reg,3)
```

plutôt plate et proche de 1.
Rappel : peu d'observations..
→ faisons un test



Interpretation:

- **[P2]** can be checked by examining the *Scale-location*-plot. The postulate is validated if we see a horizontal line with equally spread points. .
- Here it seems difficult to validate this postulate based on visual inspection. So, let us make a Breush-Pagan test ($\mathcal{H}_O$: homoscedasticity) to assess it.

[P2] homoscedasticity

```
ncvTest(reg)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.7560996, Df = 1, p = 0.38455
```

Ho non rejetée
→ homoscedasticity .

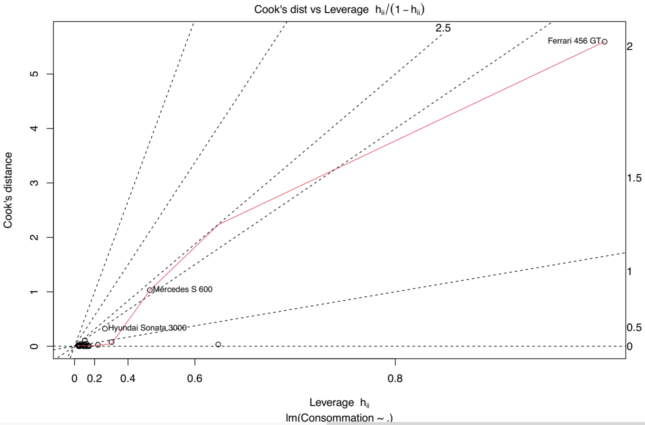
Interpretation:

- Breush-Pagan test : $\mathcal{H}_O$: homoscedasticity
- The command for the Breush-Pagan test is `ncvTest`. The homoscedasticity is rejected if the *p-value* is less than 0.05. Here, *p-value*$=0.38455 > 0.05$, the postulate is validated.

Slide pas censé être là

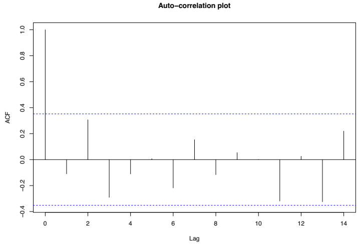
Nouveau P3

```
plot(reg,6)
```



[P3] Errors are uncorrelated

```r
acf(residuals(reg),main="Auto-correlation plot")
```



Interpretation:

- Auto-correlation of the residues can be represented with the command `acf()`. In our example, except the first one, none should exceeds dashed thresholds to validate the postulate..
- Thus uncorrelation is satisfied.

On peut aussi faire un test

[P3] Errors are uncorrelated

```
set.seed(2020)
durbinWatsonTest(reg)
```

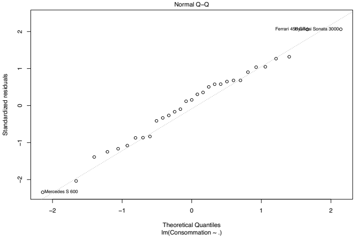
```
##  lag Autocorrelation D-W Statistic p-value
##    1      -0.1096954      2.180495   0.756
##  Alternative hypothesis: rho != 0
```

Interpretation:

- Durbin-Watson test : $\mathcal{H}_O$: uncorrelation
- Here, the *p-value*$= 0.756 > 0.05$ thus we can't reject H_0, the postulate is validated.

[P4] Errors are gaussian

```
plot(reg,2)
```



Interpretation:

- The points appear reasonably aligned along the reference line even the sample size $n = 31$ is small
- Thus, the postulate is validated.

[P4]: Errors are gaussian

```
shapiro.test(residuals((reg)))
```

ou Kolmogorov - Smirnoff

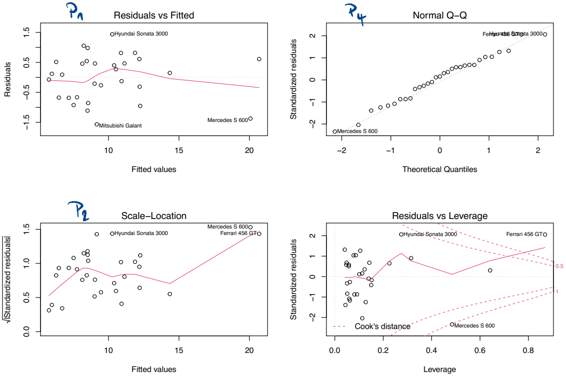
```
##
##  Shapiro-Wilk normality test
##
## data:  residuals((reg))
## W = 0.9709, p-value = 0.5442
```

Interpretation:

- Shapiro-Wilk test : $\mathcal{H}_O$: gaussien
- Here, the *p-value*$=0.5442 > 0.05$ thus we can't reject $\mathcal{H}_0$, the postulate is validated.

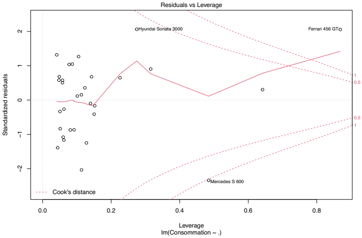
Model validation fonction `plot()`

```
par(mfrow=c(2,2))
plot(reg)
```



Cook's distance plot

```
plot(reg,which=5)
```



Interpretation:

- It appears in this plot, that 2 observations (`Ferrari 456 GT` and `Mercedes S 600`) have a Cook's distance larger than 1. They are *outliers* : *regression outliers* or *leverage points* or both.
- Be studied in section 5.

Section 4

4. Confidence interval

Confidence interval for the parameters β_j

```
cbind(confint(reg),coef(reg))
```

```
##                     2.5 %       97.5 %
## (Intercept)  1.167851e+00 3.744737e+00  2.456294e+00
## Prix         2.474392e-06 3.836669e-05  2.042054e-05
## Cylindree   -1.682157e-03 6.809703e-04 -5.005933e-04
## Puissance    4.455929e-03 4.553302e-02  2.499448e-02
## Poids        2.354210e-03 5.966955e-03  4.160583e-03
```

- **`confint`**: to display confidence intervals for the parameters β_j of the model.
- They are based on a Student's law, if the postulate **[P4]** is satisfied. If not, these intervals are biased.

Confidence Interval for $x_i^T \beta$

```
ICconf = predict(reg, interval = "confidence", level = 0.95)
head(ICconf)
```

```
##                          fit      lwr      upr
## Daihatsu Cuore      5.773872 5.145857 6.401888
## Suzuki Swift 1.0 GLS 6.475902 5.966890 6.984914
## Fiat Panda Mambo L  5.981720 5.416875 6.546566
## VW Polo 1.4 60      7.183591 6.705853 7.661329
## Opel Corsa 1.2i Eco 6.709359 6.174759 7.243959
## Subaru Vivio 4WD    6.285932 5.651261 6.920603
```

Confidence Interval for Y_i

```
ICpred= predict(reg, interval = "prediction", level = 0.95)
head(ICpred)
```

```
##                          fit      lwr      upr
## Daihatsu Cuore      5.773872 3.980461 7.567284
## Suzuki Swift 1.0 GLS 6.475902 4.720620 8.231184
## Fiat Panda Mambo L  5.981720 4.209442 7.753999
## VW Polo 1.4 60      7.183591 5.437121 8.930060
## Opel Corsa 1.2i Eco 6.709359 4.946486 8.472231
## Subaru Vivio 4WD    6.285932 4.490179 8.081685
```

Confidence Interval for `Consommation`

Set a simple linear model **Consommation** $=\beta_0+\beta_1$ **Poids**

```
regP=lm(Consommation~Poids,data=conso_voit)
newgrille=data.frame(Poids=seq(min(conso_voit$Poids)+1,
                               max(conso_voit$Poids)-1),2)
predicgrille=predict.lm(regP,newgrille,
                        interval='confidence',level=0.95)
head(predicgrille)
```

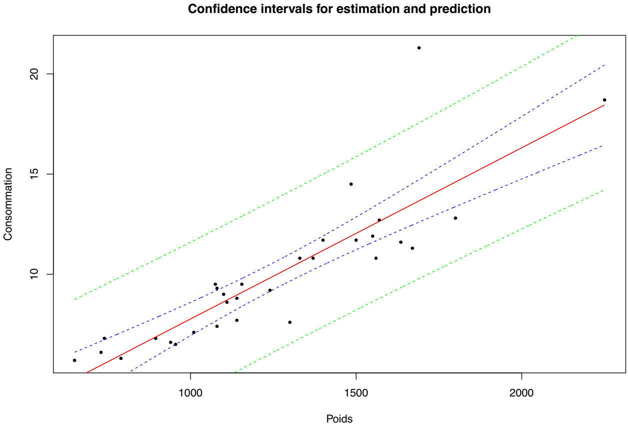
```
##        fit      lwr      upr
## 1 4.783165 3.455416 6.110914
## 2 4.791711 3.465595 6.117828
## 3 4.800257 3.475773 6.124742
## 4 4.808804 3.485950 6.131658
## 5 4.817350 3.496126 6.138574
## 6 4.825897 3.506302 6.145491
```

Confidence Interval for `Consommation`

```
plot(conso_voit$Poids,conso_voit$Consommation,
     ylab="Consommation",xlab="Poids",
     pch=20,cex=0.8,type="p",
     main="Confidence intervals for estimation and prediction")

matlines(newgrille$Poids,predicgrille,lty=c(1,2,2),
         col=c("red","blue","blue"))
predicgrille=predict.lm(regP,newgrille,
                        interval='prediction',level=0.95)
matlines(newgrille$Poids,predicgrille,lty=c(1,2,2),
         col=c("red","green","green"))
```

plot Confidence Interval for `Consommation`



Section 5

5. Outliers and leverage points

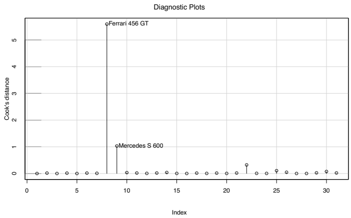
Outliers and leverage points

The library **`car`** offers an easy way to detect graphically atypical observations and to assess about their nature by tests. The command **`influenceIndexPlot`** is an important one.

Cook's distance plot

```
influenceIndexPlot(reg,vars="Cook")
```

Si je ne met pas ça, j'ai tous les plots

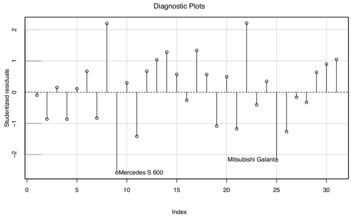


The Cook's distance plot highlight the influence of each observation on the estimation of the model (on β). As seen on the previous chapter, we compare the Cook's distance with 1. Here, two observations have a Cook's distance larger than 1 :

$\boxed{\texttt{Ferrari 456 GT and Mercedes S 600}}$

Studentized plot

```
influenceIndexPlot(reg,vars="Studentized")
```



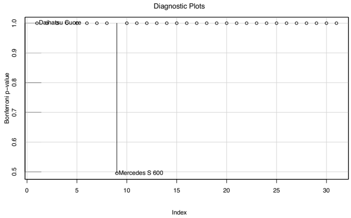
The studentized residuals plot can also be used to detect outliers. Values higher than 2 are flagged as outliers. Here, two observations seems doubtful :

$\boxed{\texttt{Mitsubishi Galant and Mercedes S 600}}$

Bonferroni plot

```
influenceIndexPlot(reg,vars="Bonf")
```

tout outlier (à retirer)
si p-value < 0,05



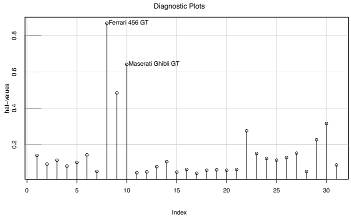
Bonferroni *p-value* plot. Is considered an outlier any observation with a *p-value* less than 0.05. Here, the plot detetects one observation:

$\boxed{\texttt{Mercedes S 600}}$

Hat plot

```
influenceIndexPlot(reg,vars="hat")
```

> 0.5 : fort impact sur sa propre estimation

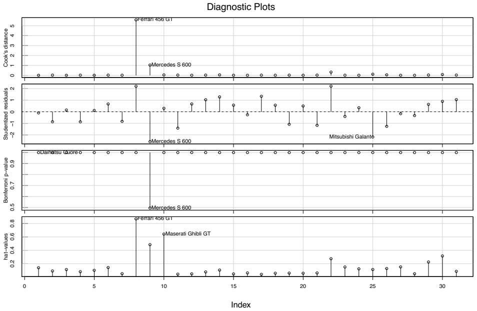


It represents the leverage (h_{ii}) of each observation on its own estimate. An observation is considered to be a *high leverage point* when this value is higher than 0.05. Here, the doubtful observations are :

$\boxed{\texttt{Ferrari 456 GT and Maserati Ghibli GT}}$

All the plots to better vision

```
influenceIndexPlot(reg)
```



pas grave [-2;2]

outliers

impact on own estimation

Here, the doubtful observations are :

$\boxed{\texttt{Mercedes S 600 and Ferrari 456 GT}}$

Access to Bonferroni's with the `outlierTest` command

```
outlierTest(reg)
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##                  rstudent unadjusted p-value Bonferroni p
## Mercedes S 600 -2.584781            0.01597      0.49506
```

- The adjusted *p-value* by the Bonferonni method is equal to 0.49506 and is very far from the threshold of 0.05. The `Mercedes S 600` observation can not be considered as `outlier`.
- To assess if the observations `Ferrari 456 GT` and `Mercedes S 600` really an big impact on the estimation of our model (β), we can compare the results of the estimation of β with and without these observations. Use the command `comparCoefs`.

To going futher

Let us keep an observation ($i = 24$) to evaluate the quality of our model.

```
voit=conso_voit[-c(24),]; voit_c=conso_voit_complet[-c(24),]
Ndata=conso_voit[24,]
Ndata
```

```
##                      Prix Cylindree Puissance Poids Consommation
## Mazda Hachtback V 36200      2497       122  1330         10.8
```

j'enlève un individu, mais je le conserve quand même qq part

Creation of a new dataset

Take the data `Ndata` and the observations `Ferrari 456 GT` and `Mercedes S 600` to create a new data set.

```
Fer=voit[c(which(voit_c$Type=="Ferrari 456 GT")),]
Mer=voit[c(which(voit_c$Type=="Mercedes S 600")),]
NEW=rbind.data.frame(Ndata,Fer,Mer)
kable(NEW)
```

	Prix	Cylindree	Puissance	Poids	Consommation
Mazda Hachtback V	36200	2497	122	1330	10.8
Ferrari 456 GT	285000	5474	325	1690	21.3
Mercedes S 600	183900	5987	300	2250	18.7

Declaration of 4 models

bon sans Ferrari sans Mercedes

Complete model (**modFb**), without one (**modFsF**), without the other (**modFsM**) and without both (**modFsFM**).

```
modFb=lm(Consommation~ Puissance + Poids+ Prix,data=voit)
modFsF = lm(Consommation~ Puissance + Poids+ Prix,
            data=voit[-c(which(voit_c$Type=="Ferrari 456 GT")),])
modFsM= lm(Consommation~ Puissance + Poids+ Prix,
            data=voit[-c(which(voit_c$Type=="Mercedes S 600")),])
modFsFM= lm(Consommation~ Puissance + Poids+ Prix,
            data=voit[-c(which(voit_c$Type=="Ferrari 456 GT"),
                     which(voit_c$Type=="Mercedes S 600")),])
```

Prediction with the 4 differents models

Prediction of NEW

What **Consommation** predict the 4 models for the 3 observations defined in **NEW**?

```
Result=cbind.data.frame(NEW$Consommation,predict(modFb,newdata=NEW),
        predict(modFsF,newdata=NEW),predict(modFsM,newdata=NEW),
        predict(modFsFM,newdata=NEW))
names(Result)=c("True","modFb","modFsF","modFsM","modFsMF")
kable(Result)
```

	True	modFb	modFsF	modFsM	modFsMF
Mazda Hachtback V	10.8	10.56498	10.98804	10.78030	10.75619
Ferrari 456 GT	21.3	20.64765	16.36461	21.31119	21.77824
Mercedes S 600	18.7	20.32782	19.01070	21.18636	21.38130

Square Root of the Mean Square Error (RMSE) on NEW

root mean squared error

$$RMSE = \sqrt{\frac{1}{3}\sum_{i=1}^{3}\left(Y_i - \widehat{Y}_i\right)^2}$$

```
RMSE=cbind.data.frame('RMSE : ',
sqrt(mean((NEW$Consommation-predict(modFb,newdata=NEW))^2)),
sqrt(mean((NEW$Consommation-predict(modFsF,newdata=NEW))^2)),
sqrt(mean((NEW$Consommation-predict(modFsM,newdata=NEW))^2)),
sqrt(mean((NEW$Consommation-predict(modFsFM,newdata=NEW))^2)))
names(RMSE)=c("","modFb","modFsF","modFsM","modFsFM")
```

Square Root of the Mean Square Error (RMSE) on NEW

```
kable(RMSE)
```

	modFb	modFsF	modFsM	modFsFM
RMSE :	1.021537	2.857152	1.435561	1.572686

The 2 observations have little influence on the coefficients of the model parameters, as well as on their standard error, since these values do not vary much, even if they have a Cook's distance larger than 1.