Linear regression

Cathy Maugis-Rabusseau

INSA Toulouse / IMT
GMM 116
cathy.maugis@insa-toulouse.fr

2023-2024

Outline

1. **Introduction**

2. Estimation

3. Tests, confidence intervals and prediction intervals

4. Selection of explanatory variables

5. Regularized regressions

6. Model validation

Example

- Data collected for 31 persons during aerobic sessions
- 7 variables:
  - age (a): age
  - weight (w): weight
  - oxy (oxy): oxygen consumption
  - runtime (run): time of effort
  - rstpulse (rst): heart rate measurement 1
  - runpulse (rp): heart rate measurement 2
  - maxpulse (maxp): heart rate measurement 3

```
head(fitness)
```

```
  age weight    oxy runtime rstpulse runpulse maxpulse
1  44  89.47 44.609   11.37       62      178      182
2  40  75.07 45.313   10.07       62      185      185
3  44  85.84 54.297    8.65       45      156      168
4  42  68.15 59.571    8.17       40      166      172
5  38  89.02 49.874    9.22       55      178      180
6  47  77.45 44.811   11.63       58      176      176
```

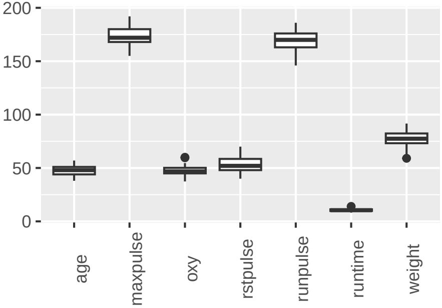
Example

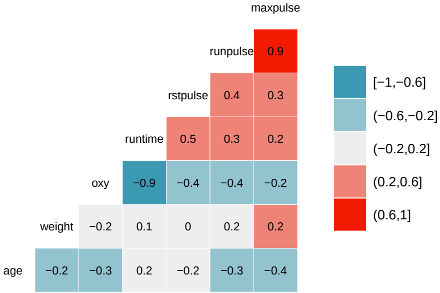
Goal: explain the consumption of oxygen (response variable Y=oxy) according to the other quantitative variables ($p = 6$).

```
summary(fitness)
```

```
      age            weight           oxy            runtime     
 Min.   :38.00   Min.   :59.08   Min.   :37.39   Min.   : 8.17  
 1st Qu.:44.00   1st Qu.:73.20   1st Qu.:44.96   1st Qu.: 9.78  
 Median :48.00   Median :77.45   Median :46.77   Median :10.47  
 Mean   :47.68   Mean   :77.44   Mean   :47.38   Mean   :10.59  
 3rd Qu.:51.00   3rd Qu.:82.33   3rd Qu.:50.13   3rd Qu.:11.27  
 Max.   :57.00   Max.   :91.63   Max.   :60.05   Max.   :14.03  
    rstpulse        runpulse        maxpulse    
 Min.   :40.00   Min.   :146.0   Min.   :155.0  
 1st Qu.:48.00   1st Qu.:163.0   1st Qu.:168.0  
 Median :52.00   Median :170.0   Median :172.0  
 Mean   :53.45   Mean   :169.6   Mean   :173.8  
 3rd Qu.:58.50   3rd Qu.:176.0   3rd Qu.:180.0  
 Max.   :70.00   Max.   :186.0   Max.   :192.0  
```

Example





	weight	oxy	runtime	rstpulse	runpulse	maxpulse
runpulse						0.9
rstpulse					0.4	0.3
runtime				0.5	0.3	0.2
oxy			−0.9	−0.4	−0.4	−0.2
weight		−0.2	0.1	0	0.2	0.2
age	−0.2	−0.3	0.2	−0.2	−0.3	−0.4

Linear regression

- **Regression**: model for establishing a link between a quantitative variable and one or more other quantitative variables

- **Simple regression**: explain one quantitative response variable according to one quantitative variable (e.g *oxy*$\sim$*runtime*)

- **Multiple regression**: explain one quantitative response variable according to several quantitative variables (e.g *oxy* according to all other quantitative variables)

- The regression requires the existence of a cause and effect relationship between the variables taken into account in the model.

Notation

- Let Y be a **quantitative response** variable
- Let p **quantitative explanatory** variables $z^{(1)}, \ldots, z^{(p)}$ (predictors)
- Data : the observation of a n-sample:

$$Y := \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix} \text{ and } \forall i = 1, \ldots, n,\ z_i = (z_i^{(1)}, \ldots, z_i^{(p)})$$

- In our example, $n = 31$, Y is the variable *oxy* and $p = 6$.

Simple linear regression model

For each individual i ($i = 1, \cdots, n$), we observe

- $Y_i =$ value of the response variable Y (e.g *oxy*),
- $z_i =$ value of the quantitative explanatory variable z (e.g *runtime*)

Simple linear regression model:

$$\left\{ \begin{array}{l} Y_i = \theta_0 + \theta_1 z_i + \varepsilon_i, \forall i = 1, \cdots, n, \\ \\ \varepsilon_1, \ldots, \varepsilon_n \text{ i.i.d } \mathcal{N}(0, \sigma^2) \end{array} \right.$$

Multiple linear regression model

For each individual i ($i = 1, \cdots, n$), we observe

- $Y_i =$ value of the quantitative response variable Y (e.g *oxy*),
- $z_i^{(1)}, \cdots, z_i^{(p)}$ values of the p quantitative explanatory variables

Multiple linear regression model:

$$\begin{cases} Y_i = \theta_0 + \theta_1 z_i^{(1)} + \cdots, + \theta_p z_i^{(p)} + \varepsilon_i, \quad \forall i = 1, \cdots, n \\ \\ \varepsilon_1, \ldots, \varepsilon_n \text{ i.i.d } \mathcal{N}(0, \sigma^2) \end{cases}$$

Outline

1 Introduction

2 **Estimation**

3 Tests, confidence intervals and prediction intervals

4 Selection of explanatory variables

5 Regularized regressions

6 Model validation

Outline

2 **Estimation**

- Least squares estimators
- Predicted values and residuals
- SST, SSE, SSR, R^2

Least squares estimator of θ

- Linear regression model:

$$\underbrace{\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix}}_{Y} = \underbrace{\begin{pmatrix} 1 & z_1^{(1)} & z_1^{(2)} & \dots & z_1^{(p)} \\ 1 & z_2^{(1)} & z_2^{(2)} & \dots & z_2^{(p)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & z_n^{(1)} & z_n^{(2)} & \dots & z_n^{(p)} \end{pmatrix}}_{X} \underbrace{\begin{pmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_p \end{pmatrix}}_{\theta} + \underbrace{\begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}}_{\varepsilon}$$

where $X \in \mathcal{M}_{n,p+1}(\mathbb{R})$ (here, $k = p + 1$).

- If the model is **regular** ($X'X$ invertible),

$$\widehat{\theta} = (X'X)^{-1}X'Y \sim \mathcal{N}_{p+1}(\theta, \sigma^2(X'X)^{-1})$$

Outline

2 **Estimation**

- Least squares estimators
- Predicted values and residuals
- SST, SSE, SSR, R^2

Predicted values and residuals

- **Predicted values** of Y: $\widehat{Y} = X\widehat{\theta}$

$$\widehat{Y_i} = (X\widehat{\theta})_i = \widehat{\theta}_0 + \sum_{j=1}^{p} \widehat{\theta}_j z_i^{(j)}$$

$=$ projection of Y onto the subspace $Im(X)$

- **Residuals**: $\widehat{\varepsilon} = Y - \widehat{Y}$ i.e $\forall i = 1, \ldots, n, \quad \widehat{\varepsilon}_i = Y_i - \widehat{Y}_i$

$=$ the orthogonal projection of Y onto the subspace $Im(X)^\perp$

- Estimator of the variance σ^2:

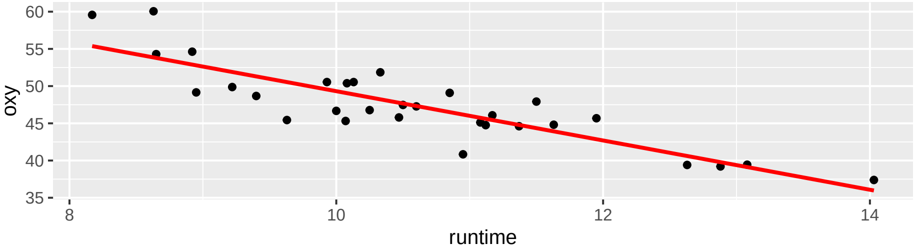
$$\widehat{\sigma^2} = \frac{\| Y - X\widehat{\theta}\|^2}{n - (p+1)} = \frac{1}{n-(p+1)} \sum_{i=1}^{n} (\widehat{\varepsilon}_i)^2 .$$

Simple linear regression

The least squares estimators of θ_0 and θ_1 are:

$$\begin{cases} \widehat{\theta_1} = \frac{cov(Y,z)}{var(z)} = \frac{\sum\limits_{i=1}^{n}(z_i - \overline{z})(Y_i - \overline{Y})}{\sum\limits_{i=1}^{n}(z_i - \overline{z})^2} \\ \widehat{\theta_0} = \overline{Y} - \widehat{\theta_1}\,\overline{z} \end{cases}$$

where $\overline{z} = \frac{1}{n}\sum_{i=1}^{n} z_i$ and $\overline{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$



Example (Simple regression)

```r
reg.simple<-lm(oxy~runtime,data=fitness)
summary(reg.simple)
```

```
Call:
lm(formula = oxy ~ runtime, data = fitness)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.3352 -1.8424 -0.0569  1.5342  6.2033 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  82.4218     3.8553  21.379  < 2e-16 ***
runtime      -3.3106     0.3612  -9.166 4.59e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.745 on 29 degrees of freedom
Multiple R-squared:  0.7434,	Adjusted R-squared:  0.7345 
F-statistic: 84.01 on 1 and 29 DF,  p-value: 4.585e-10
```

Example (Simple regression)

```python
import statsmodels.api as sm
import numpy as np
fitnesspy=r.fitness
x = np.array(fitnesspy.runtime).reshape((-1, 1))
x = sm.add_constant(x)
y = np.array(fitnesspy.oxy)
regsimplepy = sm.OLS(y, x)
resultsregsimple = regsimplepy.fit()
print(resultsregsimple.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.743
Model:                            OLS   Adj. R-squared:                  0.735
Method:                 Least Squares   F-statistic:                     84.01
Date:                Mar, 22 aoû 2023   Prob (F-statistic):           4.59e-10
Time:                        08:21:13   Log-Likelihood:                -74.254
No. Observations:                  31   AIC:                             152.5
Df Residuals:                      29   BIC:                             155.4
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         82.4218      3.855     21.379      0.000      74.537      90.307
x1            -3.3106      0.361     -9.166      0.000      -4.049      -2.572
==============================================================================
Omnibus:                        0.032   Durbin-Watson:                   1.924
Prob(Omnibus):                  0.984   Jarque-Bera (JB):                0.072
Skew:                           0.028   Prob(JB):                        0.964
Kurtosis:                       2.770   Cond. No.                         84.2
==============================================================================
```

Example (Multiple regression)

```r
reg.multi<-lm(oxy~.,data=fitness)
summary(reg.multi)
```

```
Call:
lm(formula = oxy ~ ., data = fitness)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.4026 -0.8991  0.0706  1.0496  5.3847 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 102.93448   12.40326   8.299 1.64e-08 ***
age          -0.22697    0.09984  -2.273  0.03224 *  
weight       -0.07418    0.05459  -1.359  0.18687    
runtime      -2.62865    0.38456  -6.835 4.54e-07 ***
rstpulse     -0.02153    0.06605  -0.326  0.74725    
runpulse     -0.36963    0.11985  -3.084  0.00508 ** 
maxpulse      0.30322    0.13650   2.221  0.03601 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.317 on 24 degrees of freedom
Multiple R-squared:  0.8487,    Adjusted R-squared:  0.8108 
F-statistic: 22.43 on 6 and 24 DF,  p-value: 9.715e-09
```

Example (Multiple regression)

```python
list_var=fitnesspy.columns.drop("oxy")
X=fitnesspy[list_var]
X = sm.add_constant(X)
y=np.array(fitnesspy.oxy)

regmultipy = sm.OLS(y, X)
resultsregmulti = regmultipy.fit()
```

Example (Multiple regression)

```
print(resultsregmulti.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.849
Model:                            OLS   Adj. R-squared:                  0.811
Method:                 Least Squares   F-statistic:                     22.43
Date:                Mar, 22 aoû 2023   Prob (F-statistic):           9.72e-09
Time:                        08:21:14   Log-Likelihood:                -66.068
No. Observations:                  31   AIC:                             146.1
Df Residuals:                      24   BIC:                             156.2
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        102.9345     12.403      8.299      0.000      77.335     128.534
age           -0.2270      0.100     -2.273      0.032      -0.433      -0.021
weight        -0.0742      0.055     -1.359      0.187      -0.187       0.038
runtime       -2.6287      0.385     -6.835      0.000      -3.422      -1.835
rstpulse      -0.0215      0.066     -0.326      0.747      -0.158       0.115
runpulse      -0.3696      0.120     -3.084      0.005      -0.617      -0.122
maxpulse       0.3032      0.136      2.221      0.036       0.022       0.585
==============================================================================
Omnibus:                        2.609   Durbin-Watson:                   1.711
Prob(Omnibus):                  0.271   Jarque-Bera (JB):                1.465
Skew:                          -0.069   Prob(JB):                        0.481
Kurtosis:                       4.056   Cond. No.                     7.91e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 7.91e+03. This might indicate that there are
```

Properties in simple linear regression

1. $\sum_{i=1}^n \widehat{\varepsilon_i} = 0$, $\sum_{i=1}^n \widehat{Y_i} = \sum_{i=1}^n Y_i$
2. The regression line passes through the coordinate point $(\overline{z}, \overline{Y})$.
3. The residual vector is not correlated with the explanatory variable: $cov(z, \widehat{\varepsilon}) = 0$.
4. The residual vector is not correlated with the vector of the fitted values: $cov(\widehat{Y}, \widehat{\varepsilon}) = 0$.
5. The variance of Y admits the following decomposition:

$$var(Y) = var(\widehat{Y}) + var(\widehat{\varepsilon}).$$

6. The square of the correlation coefficient between z and Y can be written as follows:

$$r^2(z, Y) = \frac{var(\widehat{Y})}{var(Y)} = 1 - \frac{var(\widehat{\varepsilon})}{var(Y)}.$$

We deduce that the empirical variance of Y can be decomposed into the *explained variance* ($var(\widehat{Y})$) and *the residual variance* ($var(\widehat{\varepsilon})$), and $r^2(z, Y)$ is the ratio of the explained variance and the total variance.

Outline

2 Estimation

- Least squares estimators
- Predicted values and residuals
- SST, SSE, SSR, R^2

Coefficient of determination R^2

- R^2 is a measure for goodness-of-fit. It is the ratio of the explained variance and the total variance:

$$R^2 = r^2(z, Y) = \frac{var(\widehat{Y})}{var(Y)} \in [0, 1]$$

- Decomposition of the variability: $SST = SSE + SSR$
  - **Total sum of squares**: $SST = \sum_{i=1}^{n}(Y_i - \overline{Y})^2$,
  - **Explained sum of squares**: $SSE = \sum_{i=1}^{n}(\widehat{Y}_i - \overline{Y})^2$,
  - **Residual sum of squares**: $SSR = \sum_{i=1}^{n}(\widehat{\varepsilon}_i)^2$

$$\Longrightarrow R^2 = \frac{SSE}{SST} = 1 - \frac{SSR}{SST}$$

Example (SST, SSE, SSR)

```
anova(reg.simple)
```

```
Analysis of Variance Table

Response: oxy
          Df Sum Sq Mean Sq F value    Pr(>F)    
runtime    1 632.90  632.90  84.008 4.585e-10 ***
Residuals 29 218.48    7.53                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# SSE
var(reg.simple$fitted.values)*(n-1)
```

```
[1] 632.9001
```

```
# SSR
var(reg.simple$residuals)*(n-1)
```

```
[1] 218.4814
```

```
# SST
var(fitness$oxy)*(n-1)
```

```
[1] 851.3815
```

Example (SST,SSE,SSR)

```
print('SSR:', resultsregsimple.ssr)
```

```
SSR: 218.48144498782733
```

```
print('SSE:', resultsregsimple.ess)
```

```
SSE: 632.9000998508823
```

```
print('SST:', resultsregsimple.centered_tss)
```

```
SST: 851.3815448387096
```

Coefficient of determination R^2

- In the case of a multiple regression of Y by $z^{(1)}, \cdots, z^{(p)}$, the *multiple correlation coefficient* is defined as the empirical correlation coefficient between Y and $\widehat{Y}$:

$$r(Y, z^{(1)}, \cdots, z^{(p)}) = r(Y, \widehat{Y}).$$

- The coefficient of determination $R^2 = r^2(Y, z^{(1)}, \cdots, z^{(p)})$.

- When an explanatory variable is added in a model, the sum of the squares of the residuals decreases or at least remains stable. This implies the "mechanical" increase of the R^2 without improving the model.

Example R^2

- With R :

```r
print(paste('Coefficient of determination: ',round(summary(reg.simple)$r.squared,3),sep=""))
```

```
[1] "Coefficient of determination: 0.743"
```

```r
print(paste('Coefficient of determination: ',round(summary(reg.multi)$r.squared,3),sep=""))
```

```
[1] "Coefficient of determination: 0.849"
```

- With Python :

```python
print('coefficient of determination:',round(np.float(resultsregsimple.rsquared),3))
```

```
coefficient of determination: 0.743
```

```python
print('coefficient of determination:', round(np.float(resultsregmulti.rsquared),3))
```

```
coefficient of determination: 0.849
```

Outline

1. Introduction

2. Estimation

3. **Tests, confidence intervals and prediction intervals**

4. Selection of explanatory variables

5. Regularized regressions

6. Model validation

Outline

3 **Tests, confidence intervals and prediction intervals**

- Student's test and Fisher's test
- Confidence intervals
- Prediction interval

Test for the nullity of θ_j

- To test the effect of an explanatory variable $z^{(j)}$:

$$\mathcal{H}_0^{(j)} : \theta_j = 0 \text{ against } \mathcal{H}_1^{(j)} : \theta_j \neq 0$$

- Student's test procedure:
  - $\hat{\theta}_j \sim \mathcal{N}(\theta_j, \sigma^2[(X'X)^{-1}]_{j+1,j+1})$
  - $(n-(p+1))\hat{\sigma}^2 \sim \sigma^2\chi^2(n-(p+1))$
  - $\hat{\theta}_j$ and $\hat{\sigma}^2$ independent

$$\Longrightarrow \qquad T_j = \frac{\widehat{\theta}_j}{\sqrt{\widehat{\sigma}^2[(X'X)^{-1}]_{j+1,j+1}}} \underset{\mathcal{H}_0}{\sim} \mathcal{T}(n-(p+1))$$

- Rejection zone:

$$\mathcal{R}_\alpha = \left\{ |T_j| \geq t_{(n-(p+1)),1-\alpha/2} \right\}$$

where $t_{(n-(p+1)),1-\alpha/2}$ is the $(1-\alpha/2)$ quantile of $\mathcal{T}(n-(p+1))$.

Example (multiple regression)

```
summary(reg.multi)
```

```
Call:
lm(formula = oxy ~ ., data = fitness)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.4026 -0.8991  0.0706  1.0496  5.3847 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 102.93448   12.40326   8.299 1.64e-08 ***
age          -0.22697    0.09984  -2.273  0.03224 *  
weight       -0.07418    0.05459  -1.359  0.18687    
runtime      -2.62865    0.38456  -6.835 4.54e-07 ***
rstpulse     -0.02153    0.06605  -0.326  0.74725    
runpulse     -0.36963    0.11985  -3.084  0.00508 ** 
maxpulse      0.30322    0.13650   2.221  0.03601 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.317 on 24 degrees of freedom
Multiple R-squared:  0.8487,    Adjusted R-squared:  0.8108 
F-statistic: 22.43 on 6 and 24 DF,  p-value: 9.715e-09
```

Example (multiple regression)

```
print(resultsregmulti.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.849
Model:                            OLS   Adj. R-squared:                  0.811
Method:                 Least Squares   F-statistic:                     22.43
Date:                Mar, 22 aoû 2023   Prob (F-statistic):           9.72e-09
Time:                        08:21:15   Log-Likelihood:                -66.068
No. Observations:                  31   AIC:                             146.1
Df Residuals:                      24   BIC:                             156.2
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        102.9345     12.403      8.299      0.000      77.335     128.534
age           -0.2270      0.100     -2.273      0.032      -0.433      -0.021
weight        -0.0742      0.055     -1.359      0.187      -0.187       0.038
runtime       -2.6287      0.385     -6.835      0.000      -3.422      -1.835
rstpulse      -0.0215      0.066     -0.326      0.747      -0.158       0.115
runpulse      -0.3696      0.120     -3.084      0.005      -0.617      -0.122
maxpulse       0.3032      0.136      2.221      0.036       0.022       0.585
==============================================================================
Omnibus:                        2.609   Durbin-Watson:                   1.711
Prob(Omnibus):                  0.271   Jarque-Bera (JB):                1.465
Skew:                          -0.069   Prob(JB):                        0.481
Kurtosis:                       4.056   Cond. No.                     7.91e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 7.91e+03. This might indicate that there are
```

Test for the nullity of several parameters

- To test the effect of q explanatory variables (with $q \leq p$) on the response variable:

$$\mathcal{H}_0 : \theta_1 = \theta_2 = \cdots = \theta_q = 0, \text{ against } \mathcal{H}_1 : \exists j \in \{1, \ldots, q\}; \theta_j \neq 0$$

- Fisher test of sub-model:

$$\begin{array}{lll} (M1) & Y_i = \theta_0 + \theta_1 z_i^{(1)} + \cdots + \theta_p z_i^{(p)} + \varepsilon_i & \text{under } \mathcal{H}_1 \\ \text{versus} & & \\ (M0) & Y_i = \theta_0 + \theta_{q+1} z_i^{(q+1)} + \cdots + \theta_p z_i^{(p)} + \varepsilon_i & \text{under } \mathcal{H}_0. \end{array}$$

- Fisher's test statistics:

$$F = \frac{(SSR_0 - SSR_1)/q}{SSR_1/(n-(p+1))} \underset{\mathcal{H}_0}{\sim} \mathcal{F}(q, n-(p+1))$$

- Rejection zone: $\mathcal{R}_\alpha = \{F \geq f_{q,n-p-1,1-\alpha}\}$ where $f_{q,n-(p+1),1-\alpha}$ is the $(1-\alpha)$ quantile of $\mathcal{F}(q, n-(p+1))$.

Example

In our multiple linear regression example, we want to test the sub-model composed only of the variables *age*, *runtime*,*runpulse* and *maxpulse*.

```r
regfin<-lm(oxy~age + runtime+runpulse+maxpulse,data=fitness)
res=anova(regfin,reg.multi)
print(res)
```

```
Analysis of Variance Table

Model 1: oxy ~ age + runtime + runpulse + maxpulse
Model 2: oxy ~ age + weight + runtime + rstpulse + runpulse + maxpulse
  Res.Df    RSS Df Sum of Sq    F Pr(>F)
1     26 138.93
2     24 128.84  2    10.092 0.94 0.4045
```

```r
paste('F statistics: ',round(res$F[2],3))
```

```
[1] "F statistics:  0.94"
```

```r
paste('pvalue: ',round(res$`Pr(>F)`[2],3))
```

```
[1] "pvalue:  0.405"
```

Example

In our example in multiple linear regression, we want to test the sub-model composed only of the variables *age*, *runtime*,*runpulse* and *maxpulse*.

```
from statsmodels.formula.api import ols
from statsmodels.stats.anova import anova_lm
resregfin = ols('oxy~age + runtime+runpulse+maxpulse', data=fitnesspy).fit()
anovaResults = anova_lm(resregfin,resultsregmulti)
print(anovaResults)
```

```
   df_resid         ssr  df_diff   ss_diff         F   Pr(>F)
0      26.0  138.930018      0.0       NaN       NaN      NaN
1      24.0  128.837938      2.0  10.09208  0.939979  0.40455
```

```
print('F statistics:',round(np.float(anovaResults.F[1]),3))
```

```
F statistics: 0.94
```

```
print(' pvalue:',round(float(anovaResults['Pr(>F)'][1]),3))
```

```
 pvalue: 0.405
```

Test of nullity for all the parameters

- This test consists of comparing the current model to the "null model" (no explanatory variable present in the model to explain Y)

$$\mathcal{H}_0 : \theta_1 = \cdots = \theta_p = 0.$$

- Under $\mathcal{H}_0$, the "null model" is:

$$Y_i = \theta_0 + \varepsilon_i \text{ with } \widehat{\theta_0} = \overline{Y}$$

  and $SSR_0 = SST$.

- Fisher's test statistics:

$$F = \frac{SSE_1/p}{SSR_1/n-(p+1)} = \frac{R^2}{1-R^2} \times \frac{n-p-1}{p} \underset{\mathcal{H}_0}{\sim} \mathcal{F}(p, n-p-1)$$

  where $SST = SSE_1 + SSR_1$ and $R^2 = SSE_1/SST$.

- Rejection zone: $\mathcal{R}_\alpha = \{F \geq f_{p,n-p-1,1-\alpha}\}$.

Example

- With R :

```
regblanc<-lm(oxy~1,data=fitness)
anova(regblanc,reg.multi)
```

```
Analysis of Variance Table

Model 1: oxy ~ 1
Model 2: oxy ~ age + weight + runtime + rstpulse + runpulse + maxpulse
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     30 851.38
2     24 128.84  6    722.54 22.433 9.715e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- With Python :

```
resregnull = ols('oxy~1', data=fitnesspy).fit()
anovaResultsnull = anova_lm(resregnull,resultsregmulti)
print(anovaResultsnull)
```

```
   df_resid         ssr  df_diff     ss_diff          F        Pr(>F)
0      30.0  851.381545      0.0         NaN        NaN           NaN
1      24.0  128.837938      6.0  722.543607  22.432635  9.715305e-09
```

Outline

3. **Tests, confidence intervals and prediction intervals**

    - Student's test and Fisher's test
    - Confidence intervals
    - Prediction interval

Confidence interval of θ_j

Using that

- $\hat{\theta}_j \sim \mathcal{N}(\theta_j, \sigma^2[(X'X)^{-1}]_{j+1,j+1})$
- $(n-(p+1))\hat{\sigma}^2 \sim \sigma^2\chi(n-(p+1))$
- $\hat{\theta}_j$ and $\hat{\sigma}^2$ are independent

we have that

$$\frac{\widehat{\theta}_j - \theta_j}{\sqrt{\widehat{\sigma}^2[(X'X)^{-1}]_{j+1,j+1}}} \sim \mathcal{T}(n-(p+1)).$$

Then, the $1-\alpha$ confidence interval of θ_j is given by:

$$IC_{1-\alpha}(\theta_j) = \left[\ \widehat{\theta}_j \pm t_{n-(p+1),1-\alpha/2} \times \sqrt{\widehat{\sigma}^2[(X'X)^{-1}]_{j+1,j+1}}\ \right].$$

Example

```r
confint(reg.simple,level=0.9)
```

```
                  5 %      95 %
(Intercept) 75.871122 88.972424
runtime     -3.924271 -2.696839
```

```r
confint(reg.multi)
```

```
                  2.5 %       97.5 %
(Intercept) 77.33541293 128.53354604
age         -0.43302821  -0.02091938
weight      -0.18685216   0.03849733
runtime     -3.42235018  -1.83495545
rstpulse    -0.15786297   0.11479569
runpulse    -0.61699207  -0.12226345
maxpulse     0.02150491   0.58492935
```

Example

```
resultsregsimple.conf_int(0.1)
```

```
array([[75.87112183, 88.97242353],
       [-3.9242713 , -2.69683942]])
```

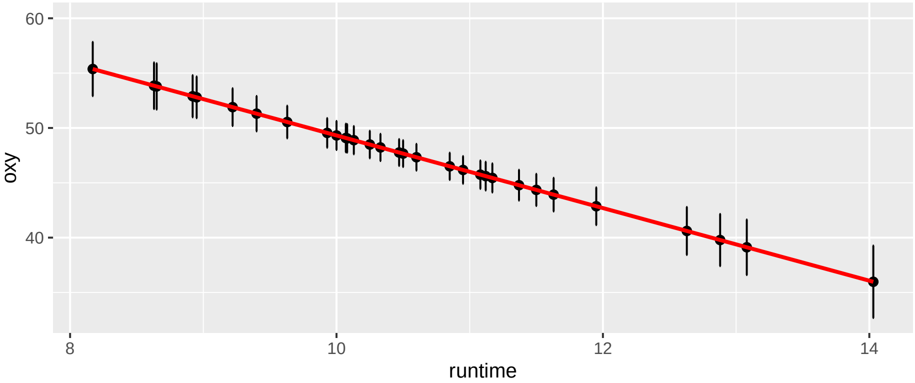
```
resultsregmulti.conf_int(0.05)
```

```
                  0           1
const     77.335413  128.533546
age       -0.433028   -0.020919
weight    -0.186852    0.038497
runtime   -3.422350   -1.834955
rstpulse  -0.157863    0.114796
runpulse  -0.616992   -0.122263
maxpulse   0.021505    0.584929
```

Confidence interval of $(X\theta)_i$

Using the construction made in Chapter 3, the confidence interval of $(X\theta)_i$ at the confidence level $1-\alpha$ is therefore given by:

$$IC_{1-\alpha}((X\theta)_i) = \left[\widehat{Y_i} \pm t_{n-(p+1),1-\alpha/2} \times \sqrt{\widehat{\sigma}^2[X(X'X)^{-1}X']_{ii}}\right].$$



Confidence interval of $X_0\theta$

- Let $z_0^{(1)}, \cdots, z_0^{(p)}$ be new values of the predictors.
  Let $X_0 = (1,\, z_0^{(1)},\, \cdots\, ,\, z_0^{(p)}) \in \mathcal{M}_{1,(p+1)}(\mathbb{R})$ be a new point.
- The mean response is

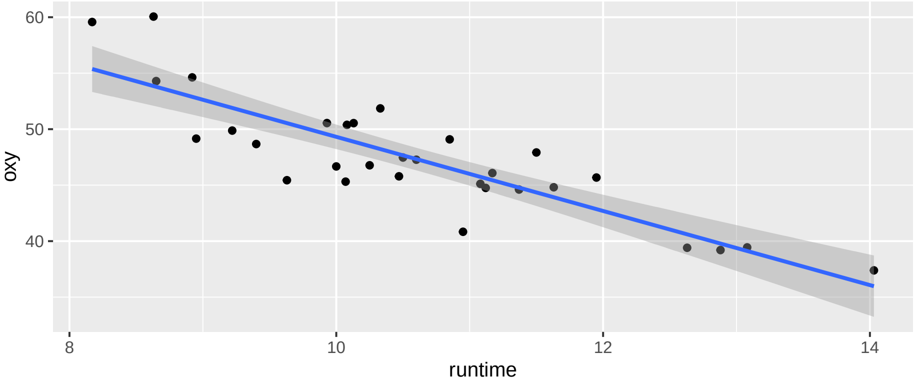
$$X_0\theta = \theta_0 + \sum_{j=1}^{p} \theta_j z_0^{(j)}.$$

- Using the construction made in Chapter 3, the $(1-\alpha)$ confidence interval of $X_0\theta$ is

$$IC_{1-\alpha}(X_0\theta) = \left[\ X_0\widehat{\theta} \pm t_{n-(p+1),1-\alpha/2} \times \sqrt{\widehat{\sigma^2} X_0(X'X)^{-1}X_0'}\ \right].$$

Confidence interval of $X_0\theta$

```r
ggplot(fitness, aes(x=runtime, y=oxy))+
    geom_point() +
    geom_smooth(method=lm, se=TRUE)+
    xlab("runtime")+
    ylab("oxy")
```



Outline

3 **Tests, confidence intervals and prediction intervals**

- Student's test and Fisher's test
- Confidence intervals
- Prediction interval

Prediction interval

We want to predict in which interval the response associated to new values of the predictors $(z_0^{(1)}, \cdots, z_0^{(p)})$. We therefore want to construct a prediction interval for the response Y_0 associated to a new point $X_0 = (1, z_0^{(1)}, z_0^{(2)}, \cdots, z_0^{(p)})$

$$Y_0 = X_0\theta + \varepsilon_0,$$

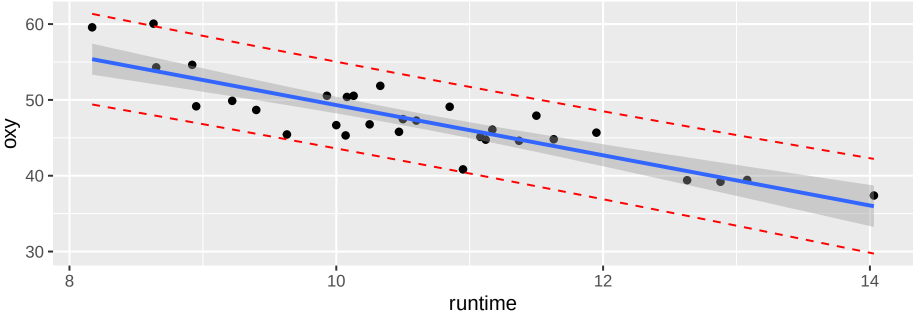
where ε_0 is independent of ε_i, $1 \leq i \leq n$ and where $\varepsilon_0 \sim \mathcal{N}(0, \sigma^2)$.

Using the construction made in Chapter 3,

$$IC_{1-\alpha}(Y_0) = \left[\ X_0\widehat{\theta} \pm t_{n-(p+1),1-\alpha/2}\widehat{\sigma}\sqrt{1 + X_0(X'X)^{-1}X_0'}\ \right].$$

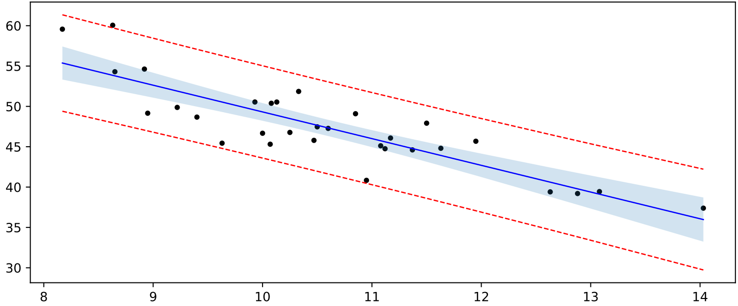
Prediction interval

```
temp_var <- predict(reg.simple, interval="prediction")
new_df <- cbind(fitness, temp_var)
ggplot(new_df, aes(x=runtime, y=oxy))+
    geom_point() +
    geom_line(aes(y=lwr), color = "red", linetype = "dashed")+
    geom_line(aes(y=upr), color = "red", linetype = "dashed")+
    geom_smooth(method=lm, se=TRUE)+
    xlab("runtime")+
    ylab("oxy")
```



Prediction interval

```python
from statsmodels.stats.outliers_influence import summary_table
st, data, ss2 = summary_table(resultsregsimple, alpha=0.05)
fittedvalues = data[:, 2]
predict_mean_se  = data[:, 3]
predict_mean_ci_low, predict_mean_ci_upp = data[:, 4:6].T
predict_ci_low, predict_ci_upp = data[:, 6:8].T
```



Outline

1. Introduction

2. Estimation

3. Tests, confidence intervals and prediction intervals

4. **Selection of explanatory variables**

5. Regularized regressions

6. Model validation

Outline

4 **Selection of explanatory variables**

- General framework
- Some criteria for model selection
- Variable selection algorithms

Introduction

- We focus on the study of the matrix X i.e. on the explanatory variables
- How to choose the model that best fits the data and eliminate some variables that are not very significant?
- Presentation of approaches allowing to refine (to select) a model among a model collection, i.e determine which are the most "significant" variables.
- For the sake of simplicity, we consider the framework of multiple linear regression. The tools presented can of course be used in a more general context (often without additional work).

General framework

- Sample of size n representing observations on a quantitative response variable Y and of p quantitative explanatory variables $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(p)}$.

- A collection of models $\mathcal{M}$ formally representing a family of subsets of $\{1, \ldots, p\}$. This choice is made a priori and may not be exhaustive. For example, we can consider
  - exhaustive collection: $\mathcal{M} = \mathcal{P}(\{1, \ldots, p\})$ i.e. the family of all the subsets of $\{1, \ldots, p\}$,
  - growing collection: $\mathcal{M} = (\{1, \ldots, m\})_{m=1,\ldots,p}$

- For $m \in \mathcal{M}$, we denote
  - $|m|$ the cardinal of m
  - $X_{(m)}$ the matrix composed of the vectors $\mathbf{x}^{(j)}$ for $j \in m$ (assumed to be regular, $\mathrm{rg}(X_{(m)}) = |m| + 1$)

General framework

- **Assumptions on the true model** : we assume that it exists $m^\star \in \mathcal{M}$, unknown, such that

$$Y = \mu^\star + \varepsilon^\star = X_{(m^\star)}\theta_{(m^\star)} + \varepsilon^\star, \text{ with } \varepsilon^\star \sim \mathcal{N}(0_n, {\sigma^\star}^2 I_n),$$

$\theta_{(m^\star)} \in \mathbb{R}^{|m^\star|+1}$ having all its non-zero coordinates.

- **Models to analyse** : To model the experiment and try to identify the true model we use the following family of models, which corresponds to $\mathcal{M}$:

$$Y = \mu_{(m)} + \varepsilon = X_{(m)}\theta_{(m)} + \varepsilon, \text{ with } \varepsilon \sim \mathcal{N}(0_n, \sigma^2 I_n).$$

Vocabulary

Let $m \in \mathcal{M}$. Then

- if $m = \{1, \cdots, p\}$, the model m is called **complete**
- if $m^\star \subset m$ with $m \neq m^\star$, the model m is called **over-fitted**;
- if $m \subset m^\star$ with $m \neq m^\star$, the model m is called **under-fitted**,
- if $|m \cap m^\star| < |m^\star|$, the model m is called **false**

We will see in the sequel various approaches allowing, not to find $m^\star$, but at least to approach it. This corresponds to the principles of the **model selection**.

Outline

4 **Selection of explanatory variables**

- General framework
- Some criteria for model selection
- Variable selection algorithms

Coefficient of determination R^2

$$R_m^2 = 1 - \frac{SCR_m}{SCT} = 1 - \frac{\|Y - X_{(m)}\hat{\theta}_{(m)}\|^2}{\|Y - \overline{Y}\mathbb{1}_n\|^2}$$

- The more explanatory variables are used, the more the adequacy increases thus the maximization of R_m^2 leads to select the complete model.
- Using this criterion favors the selection of highly parameterized models
- Possible difficulties in interpreting the chosen model (because too complex)
- For models with the same cardinal $|m|$, this coefficient can be used to choose an optimal model.

Adjusted coefficient of determination $\widetilde{R^2}_m$

- Improve the R^2_m to allow the selection of models with a different number of explanatory variables
- The adjusted coefficient of determination $\widetilde{R}^2_m$:

$$\widetilde{R^2}_m = 1 - \frac{n-1}{n-|m|-1}.\frac{SCR}{SCT} = 1 - \frac{n-1}{n-|m|-1}.\frac{\|Y - X_m\hat{\theta}_{(m)}\|^2}{\|Y - \overline{Y}\mathbb{1}_n\|^2}.$$

- $\widetilde{R^2}_m$ allow the number of regressors to be taken into account and therefore offers a compromise between the suitability and the parameterization of the model.

Forward and backward selection strategies by Fisher's test

- **Initialisation:** Let be a threshold s and $m_{[0]} = \{1, \dots, p\}$
- **Iteration** t **:**
  - *Step 1:* For all $j \in m_{[t]}$, we compute the p-value p_j of the Fisher's test

$$(M_0) : m_{[t]} \setminus \{j\} \text{ against } (M_1) : m_{[t]}$$

  - *Step 2:* $\hat{\jmath} = \arg\max_{j \in m_{[t]}} p_j$
  - *Step 3:*
    - If $p_{\hat{\jmath}} > s$, $m_{[t+1]} = m_{[t]} \setminus \{\hat{\jmath}\}$ and we go back to Step 1
    - otherwise stop.

Forward and backward selection strategies by Fisher's test

- This strategy can be extremely time consuming depending on the number of variables.
  (we can have until $|m|!$ Fisher's tests).

- The forward selection of models uses exactly the same arguments, except that we start from the empty model (without regressor, only the intercept) and we gradually add the most significant variables (within the meaning of Fisher's test), until the p-values exceed a previously fixed threshold.

Quadratic risk

The quadratic risk is a usual criterion to measure the difference between the true model $m^\star$ and a given model $m \in \mathcal{M}$.

Definition

Let $m \in \mathcal{M}$. **The quadratic risk** between models m and $m^\star$ is defined by

$$\mathcal{R}(m, m^\star) = \mathbb{E}\left[\left\|\mu^\star - \widehat{Y}_{(m)}\right\|^2\right] = \mathbb{E}\left[\left\|X_{(m^\star)}\theta_{(m^\star)} - X_{(m)}\hat{\theta}_{(m)}\right\|^2\right],$$

where $\mu^\star = X_{(m^\star)}\theta_{(m^\star)}$ and $\widehat{Y}_{(m)} = X_{(m)}\hat{\theta}_{(m)}$.

Quadratic risk

In the sequel for all $m \in \mathcal{M}$, we define $\mu^\star_{(m)} = P_{[X_{(m)}]}\mu^\star$, the orthogonal projection of $\mu^\star$ on the subspace $Im(X_{(m)})$. It is then possible to calculate this quadratic risk explicitly.

Proposition

For all $m \in \mathcal{M}$,

$$\mathcal{R}(m, m^\star) = {\sigma^\star}^2(|m| + 1) + \|\mu^\star_{(m)} - \mu^\star\|^2.$$

In order to minimize the distance between m and $m^\star$, there is thus a compromise to be found. This bias-variance compromise is very usual in this model selection framework and is found in a large statistical frameworks.

Mallows' C_p criterion

Idea: estimate the quadratic risk from the data itself and then make a decision based on this estimation.

$$\hat{m}_{CP} = \arg\min_{m \in \mathcal{M}} C_p(m)$$

where

$$C_p(m) = \|Y - \widehat{Y}_{(m)}\|^2 + 2|m|\sigma^2$$

if the variance is known.

When the variance is unknown, we will use $\widehat{\sigma^2}$ which is the variance estimator for the complete model.

Kullback-Leibler divergence

The criteria AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion) are based on the minimization of the Kullback-Leibler divergence between the both models.

Definition

Let $\mathbb{P}$ and $\mathbb{P}^\star$ be two probability distributions dominated by the same measure (here Lebesgue measure). The Kullback-Leibler divergence between these two measures is defined by

$$KL(\mathbb{P}^\star, \mathbb{P}) = \mathbb{E}_{\mathbb{P}^\star}\left[\log \frac{d\mathbb{P}^\star}{d\mathbb{P}}\right].$$

If $f = \dfrac{d\mathbb{P}}{d\nu}$ and $f^\star = \dfrac{d\mathbb{P}^\star}{d\nu}$, then $KL(\mathbb{P}^\star, \mathbb{P}) = \begin{cases} \int f^\star \log \frac{f^\star}{f} d\nu \text{ si } \mathbb{P}^\star \ll \mathbb{P}, \\ +\infty \text{ otherwise.} \end{cases}$

Kullback-Leibler divergence

- "Divergence" because $KL(.,.)$ is not symmetric
- Like any "classic" distance, it checks that
  - $KL(\mathbb{P}^\star, \mathbb{P}) \geq 0$ for all measures $\mathbb{P}^\star$ and $\mathbb{P}$;
  - $KL(\mathbb{P}^\star, \mathbb{P}) = 0$ if and only if $\mathbb{P} = \mathbb{P}^\star$.
- When the errors are Gaussian:

Proposition

Let $m \in \mathcal{M}$ fixed. Then we have

$$KL(m^\star, m) = \frac{n}{2}\left[\log\left(\frac{\sigma^2_{(m)}}{\sigma^{\star\,2}}\right) + \frac{\sigma^{\star\,2}}{\sigma^2_{(m)}} - 1\right] + \frac{1}{2\sigma^2_{(m)}}\|\mu^\star - \mu^\star_{(m)}\|^2,$$

where $KL(m^\star, m)$ is the Kullback-Leibler divergence between $m^\star$ and m.

AIC criterion

- AIC criterion consists of selecting the model satisfying

$$\hat{m} = \arg \min_{m \in \mathcal{M}} \mathsf{AIC}(m)$$

$\mathsf{AIC}(m) = -2\text{logvraisemblance au maximum de vraisemblance}+2D_m$

where D_m is the dimension of the model m (nb of param. for m).

- In the Gaussian case,

$$\ln\left[(2\pi\tilde{\sigma}^2_{(m)})^{-n/2}\exp\left(-\frac{1}{2\tilde{\sigma}^2_{(m)}}\|Y-\hat{Y}_{(m)}\|^2\right)\right]$$

$$= -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\tilde{\sigma}^2_{(m)}) - \frac{n}{2} \text{ because } \tilde{\sigma}^2_{(m)} = \frac{1}{n}\|Y-\hat{Y}_{(m)}\|^2.$$

Thus

$$\hat{m} = \arg \min_{m \in \mathcal{M}} n\ln(\tilde{\sigma}^2_{(m)}) + 2(|m|+2).$$

Corrected AIC criterion

- This criterion works quite well for small collections of models. However numerical simulations show that the quality of the estimate tends to deteriorate when m increases.

- In order to overcome this problem, it is possible to use the corrected AIC criterion:

$$\mathrm{AIC}_c(m) = n \ln\left(\tilde{\sigma}^2_{(m)}\right) + n\frac{n+|m|-1}{n-|m|-3}.$$

BIC criterion

- The BIC (Bayesian Information Criterion) is an extension of the AIC criterion and uses a Bayesian point of view
- The BIC criterion is defined by

$$\begin{aligned} BIC(m) &= -2log(L) + D_m \log(n) \\ &= n\log(\hat{\sigma}^2_{(m)}) + \log n \times D_m. \end{aligned}$$

The selected model $\hat{m}_{BIC}$ is

$$\hat{m}_{BIC} = \arg \min_{m\in\mathcal{M}} BIC(m).$$

Remark: The BIC criterion is the only criterion among those proposed to asymptotically choose the “true” model with a probability tending towards 1. The other criteria (C_p, $\widetilde{R}^2$, AIC, AIC_c) always have a positive probability of over-fitting.

Outline

4 **Selection of explanatory variables**

- General framework
- Some criteria for model selection
- Variable selection algorithms

Variable selection algorithms

In practice, once a model selection criterion has been chosen, it is impossible to determine the "best" model by an exhaustive search because of the number of models to be explored.
We therefore use step-by-step methods:

1. Backward methods:
2. Forward methods:
3. Stepwise methods:
4. "s best subsets" method

Variable selection algorithms

1. Backward methods:
    - **Initialisation :** $m_{[0]} = \{1, \ldots, p\}$
    - **Iteration** t **:**
        - *Step 1 :* For all $j \in m_{[t]}$, we compute $c_j = \text{CRIT}(m_{[t]} \setminus \{j\})$.
        - *Step 2 :* $\hat{\jmath} = \underset{j \in m_{[t]}}{\arg\max}\ c_j$
        - *Step 3 :* $m_{[t+1]} = m_{[t]} \setminus \{\hat{\jmath}\}$
          If $m_{[t+1]} \neq \emptyset$, go back to Step 1
          Otherwise stop.
2. Forward methods:
3. Stepwise methods:
4. "s best subsets" method

Variable selection algorithms

1. Backward methods:
2. Forward methods:
   - **Initialisation :** $m_{[0]} = \emptyset$
   - **Iteration** t **:**
     - *Step 1 :* For all $j \in \{1, \dots, p\} \setminus m_{[t]}$, we compute $c_j = \mathrm{CRIT}(m_{[t]} \cup \{j\})$.
     - *Step 2 :* $\hat{\jmath} = \arg\min_j c_j$
     - *Step 3 :* $m_{[t+1]} = m_{[t]} \cup \{\hat{\jmath}\}$
       If $m_{[t+1]} \neq \{1, \dots, p\}$, go back to Step 1
       Otherwise stop.
3. Stepwise methods:
4. "s best subsets" method

Variable selection algorithms

1. Backward methods:

2. Forward methods:

3. Stepwise methods:
   Starting from a given model, we select a new variable (like with an ascending method), then we look to see if we can eliminate one of the variables from the model (like for a descending method) and so on. It is necessary to define for such a method an entry criterion and an exit criterion.

4. "s best subsets" method

Variable selection algorithms

1. Backward methods:
2. Forward methods:
3. Stepwise methods:
4. "s best subsets" method
   We search exhaustively among all the subsets of s variables, the best subset according to the chosen criterion.

Example

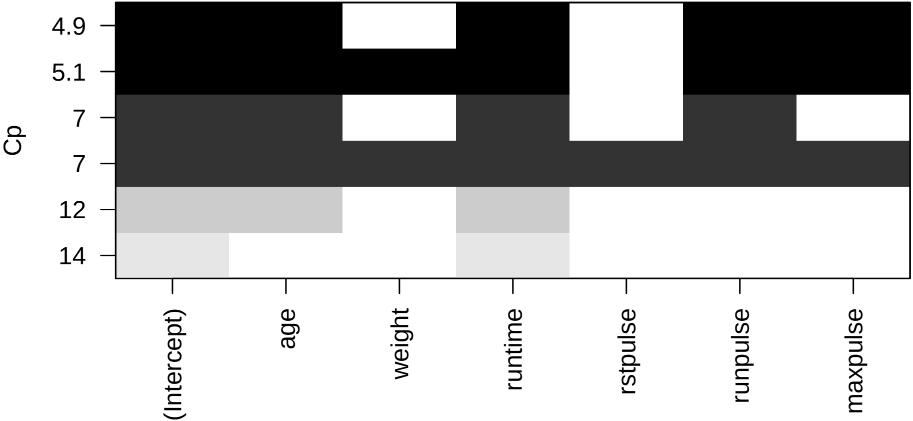
```
library(leaps)
choixb<-regsubsets(oxy~.,data=fitness,nbest=1,nvmax=10,method="backward")
summary(choixb)
```

```
Subset selection object
Call: regsubsets.formula(oxy ~ ., data = fitness, nbest = 1, nvmax = 10,
    method = "backward")
6 Variables  (and intercept)
         Forced in Forced out
age          FALSE      FALSE
weight       FALSE      FALSE
runtime      FALSE      FALSE
rstpulse     FALSE      FALSE
runpulse     FALSE      FALSE
maxpulse     FALSE      FALSE
1 subsets of each size up to 6
Selection Algorithm: backward
         age weight runtime rstpulse runpulse maxpulse
1  ( 1 ) " " " "    "*"     " "      " "      " "     
2  ( 1 ) "*" " "    "*"     " "      " "      " "     
3  ( 1 ) "*" " "    "*"     " "      "*"      " "     
4  ( 1 ) "*" " "    "*"     " "      "*"      "*"     
5  ( 1 ) "*" "*"    "*"     " "      "*"      "*"     
6  ( 1 ) "*" "*"    "*"     "*"      "*"      "*"     
```

```
choixf<-regsubsets(oxy~.,data=fitness,nbest=1,nvmax=10,method="forward")
```

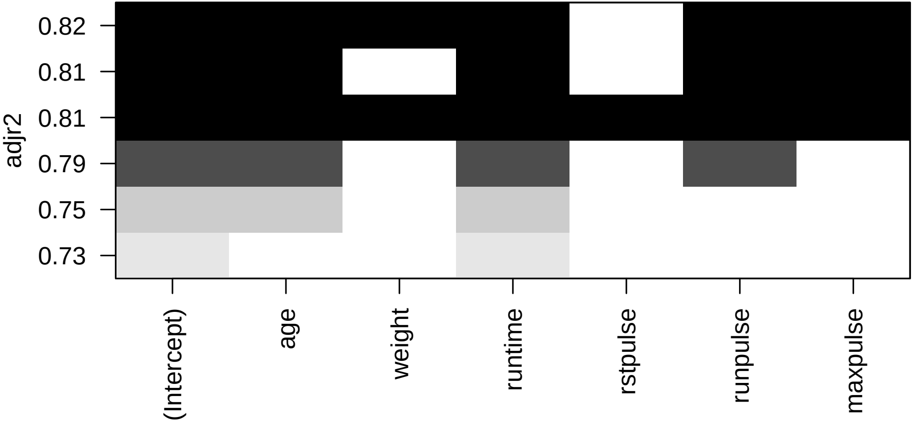
Example

```
plot(choixb,scale="Cp")
```



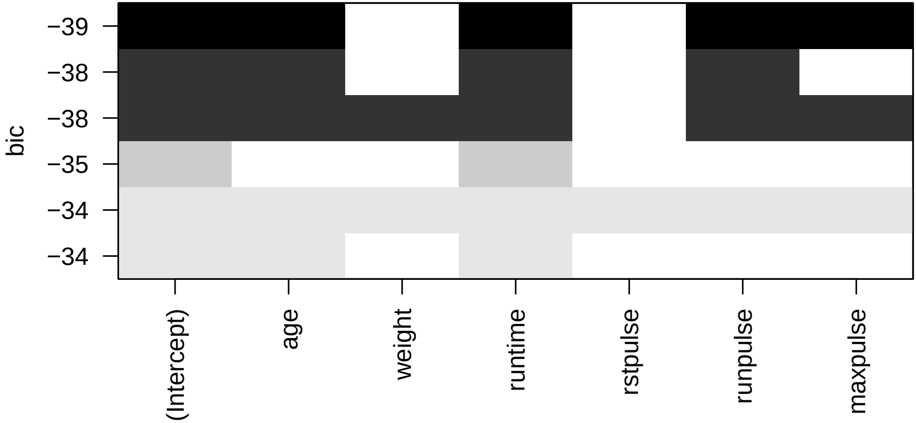
Example

```
plot(choixb,scale="adjr2")
```



Example

```
plot(choixb,scale="bic")
```



Example

```
library(MASS)
modselect_aic=stepAIC(reg.multi,trace=F,direction="backward")
modselect_bic=stepAIC(reg.multi,trace=T,direction="backward",k=log(nrow(fitness)))
```

```
Start:  AIC=68.2
oxy ~ age + weight + runtime + rstpulse + runpulse + maxpulse

           Df Sum of Sq    RSS    AIC
- rstpulse  1     0.571 129.41 64.903
- weight    1     9.911 138.75 67.063
<none>                  128.84 68.200
- maxpulse  1    26.491 155.33 70.562
- age       1    27.746 156.58 70.812
- runpulse  1    51.058 179.90 75.114
- runtime   1   250.822 379.66 98.268

Step:  AIC=64.9
oxy ~ age + weight + runtime + runpulse + maxpulse

           Df Sum of Sq    RSS     AIC
- weight    1      9.52 138.93  63.669
<none>                  129.41  64.903
- maxpulse  1     26.83 156.23  67.309
- age       1     27.37 156.78  67.417
- runpulse  1     52.60 182.00  72.041
- runtime   1    320.36 449.77 100.087

Step:  AIC=63.67
oxy ~ age + runtime + runpulse + maxpulse

           Df Sum of Sq    RSS    AIC
<none>                  138.93 63.669
- maxpulse  1     21.90 160.83 64.773
```

Example

```r
reg.fin<-lm(oxy~age+runtime+maxpulse+runpulse,data=fitness)
anova(reg.fin,reg.multi)
```

```
Analysis of Variance Table

Model 1: oxy ~ age + runtime + maxpulse + runpulse
Model 2: oxy ~ age + weight + runtime + rstpulse + runpulse + maxpulse
  Res.Df    RSS Df Sum of Sq    F Pr(>F)
1     26 138.93
2     24 128.84  2    10.092 0.94 0.4045
```

Outline

1. Introduction

2. Estimation

3. Tests, confidence intervals and prediction intervals

4. Selection of explanatory variables

5. **Regularized regressions**

6. Model validation

Outline

5. **Regularized regressions**
   - Principle of regularized regression
   - Ridge regression
   - Lasso regression
   - Elastic-Net regression

Context

- Singular model ($\mathrm{rg}(X) < k$) $\Rightarrow X'X$ is not invertible and $\widehat{\theta} \nexists$
  This case arises when
  - the number of explanatory variables is greater than the number of observations ($n < p$)
  - $n > p$ but some variables are linearly redundant
- $\Gamma_{\widehat{\theta}} = \sigma^2(X'X)^{-1}$ thus the precision of $\widehat{\theta}$ decreases when $X'X$ approaches a non-invertible matrix.
- In prediction: the quality (quadratic deviation) between the prediction $\hat{Y}^\star$ and the true response $Y^\star$ is equal to bias^2 + variance.
  $\Rightarrow$ we may prefer a slight increase of the bias to have a decrease of the variance.

Regularized regression methods

- Regularized regression methods : minimize a criterion

$$\underset{\theta \in \mathbb{R}^k}{\text{argmin}} \|Y - X\theta\|^2 + \lambda \, \text{pen}(\theta)$$

  where $\lambda > 0$ to choose and pen(θ) based on the control of a norm.

- In practice:
  - We start by centering and reducing the explanatory variables $\Rightarrow \tilde{X}$.
  - The intercept θ_0 ensures that the model is positioned around the average behavior of Y thus $\tilde{Y} = Y - \bar{Y}\mathbb{1}_n$ (and we can potentially reduce it)

- Thus we will consider the following linear model

$$\tilde{Y} = \tilde{X}\theta + \varepsilon \text{ with } \theta = (\theta_1, \ldots, \theta_p)'$$

  (thus $k = p$ and without intercept).

Regularized regression methods

- Goal: minimize the regularized empirical risk (for the quadratic loss)

$$\underset{\theta \in \mathbb{R}^k}{\text{argmin}} \left\{ \|\tilde{Y} - \tilde{X}\theta\|^2 + \lambda \|\theta\|_q^q \right\} \text{ where } \|\theta\|_q^q = \sum_{j=1}^{p} (\theta_j)^q.$$

- The most known: **Ridge regression** ($q = 2$), **Lasso regression** ($q = 1$) and **Elasticnet regression** (combine the first two)

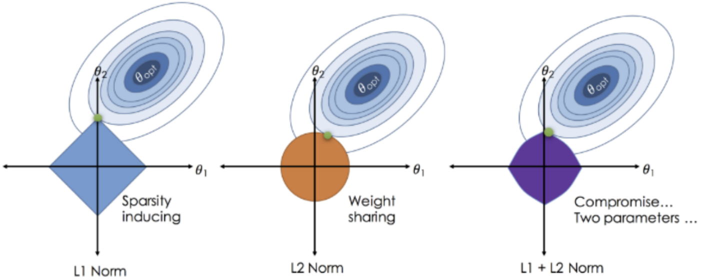
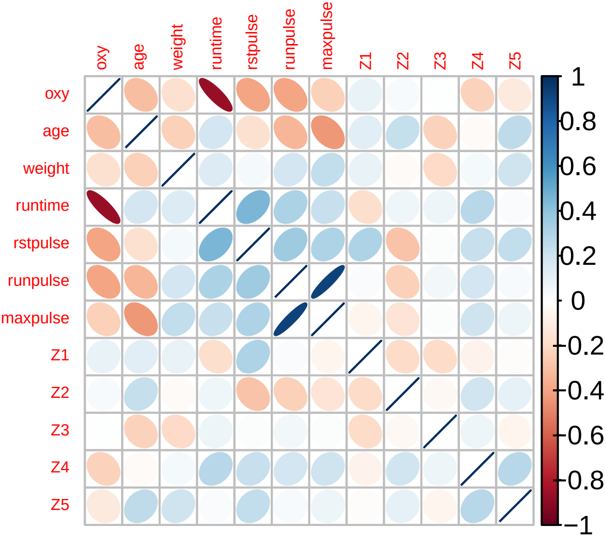


Figure 1: An image visualising how ordinary regression compares to the Lasso, the Ridge and the Elastic Net Regressors. Image Citation: Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net.

Example

- Data: "fitness" + 5 noise variables $\sim \mathcal{N}(0,1)$



Outline

5 **Regularized regressions**

- Principle of regularized regression
- Ridge regression
- Lasso regression
- Elastic-Net regression

Difficulty of invertibility

- Difficulty of invertibility of $\tilde{X}'\tilde{X} \in \mathcal{M}_p(\mathbb{R})$.
- $\tilde{X}'\tilde{X}$ is a positive semi-definite matrix so its eigenvalues are non-negative and we order decreasingly them $\tau_1 \geq \tau_2 \geq \ldots \geq \tau_p \geq 0$. If $\tilde{X}'\tilde{X}$ is not invertible, at least one of its eigenvalues is zero.

Proposition

Let $\lambda > 0$. The matrices $\tilde{X}'\tilde{X}$ and $\tilde{X}'\tilde{X} + \lambda I_p$ have the same eigenvectors but their eigenvalues are $\{\tau_j\}_{j\in[|1,p|]}$ and $\{\tau_j + \lambda\}_{j\in[|1,p|]}$ respectively. Then, $det(\tilde{X}'\tilde{X} + \lambda I_p) > det(\tilde{X}'\tilde{X})$, thus $\tilde{X}'\tilde{X} + \lambda I_p$ is "more likely" to be invertible than $\tilde{X}'\tilde{X}$.

- Idea: replace $(\tilde{X}'\tilde{X})^{-1}$ by $(\tilde{X}'\tilde{X} + \lambda I_p)^{-1}$ in $\widehat{\theta}$

Ridge estimator

- The ridge estimator is given by

$$\widehat{\theta}_{\text{ridge}}(\lambda) = (\tilde{X}'\tilde{X} + \lambda I_p)^{-1}\tilde{X}'\tilde{Y}.$$

- The ridge estimator is solution of the following optimization problem

$$\widehat{\theta}_{\text{ridge}}(\lambda) \in \underset{\theta\in\mathbb{R}^p}{\text{argmin}}\ \|\tilde{Y} - \tilde{X}\theta\|_2^2 + \lambda\|\theta\|_2^2,$$

  which can be written as the following constraint minimization problem:

$$\|\tilde{Y} - \tilde{X}\theta\|_2^2 \text{ under the constraint } \|\theta\|_2^2 \leq r(\lambda)$$

  where $r(.)$ is bijective.

- The ridge regression keeps all the variables but with the constraint $\|\theta\|_2^2 \leq r(\lambda)$, it avoids estimators to take too large values and thus limits the variance of predictions. We speak of "shrinkage" because the range of possible values of the estimated parameters is shrinking.

Properties

Proposition

Let $\widehat{\theta}_{\text{ridge}}(\lambda) = (\tilde{X}'\tilde{X} + \lambda I_p)^{-1}\tilde{X}'\tilde{Y}$ be the ridge estimator. We have

- $\mathbb{E}[\widehat{\theta}_{\text{ridge}}(\lambda)] = \theta - \lambda(\tilde{X}'\tilde{X} + \lambda I_p)^{-1}\theta$ thus it is biased.
- Variance decrease:

$$\begin{aligned} \text{Var}(\widehat{\theta}_{\text{ridge}}(\lambda)) &= \sigma^2(\tilde{X}'\tilde{X} + \lambda I_p)^{-1}(\tilde{X}'\tilde{X})(\tilde{X}'\tilde{X} + \lambda I_p)^{-1} \\ &\leq \sigma^2(\tilde{X}'\tilde{X})^{-1} = \text{Var}(\widehat{\theta}) \end{aligned}$$

- The fitted values for Y are

$$\widehat{Y}_{\text{ridge}}(\lambda) = \tilde{X}\widehat{\theta}_{\text{ridge}}(\lambda) + \bar{Y}\mathbb{1}_n$$

- When $\lambda \to +\infty$, $\widehat{\theta}_{\text{ridge}}(\lambda) \to 0$
- When $\lambda \to 0$, $\widehat{\theta}_{\text{ridge}}(\lambda) \to \widehat{\theta}$

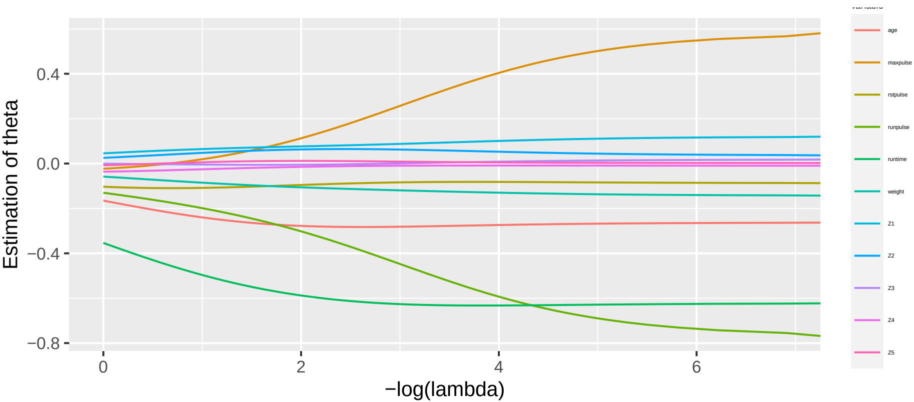
Regularization paths

- The ridge estimator $\widehat{\theta}_{\text{ridge}}(\lambda)$ depends on the choice of λ which is a tough point
- Impossible to make this choice a priori.
- Plot the ridge *regularization paths*

$$\lambda \mapsto (\widehat{\theta}_{\text{ridge}}(\lambda))_j \text{ for } j = 1, \ldots, p$$

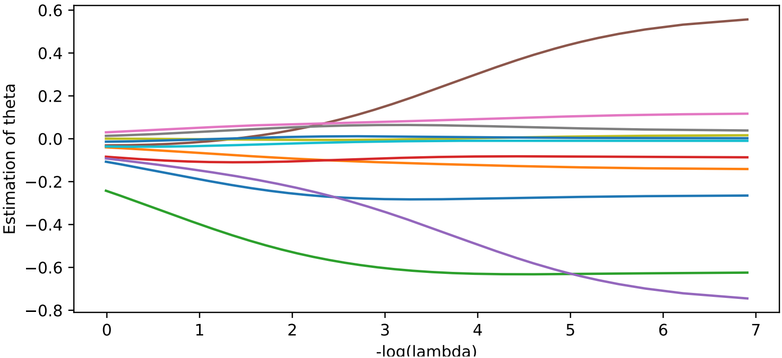
Example

```r
lambda_seq <- seq(0, 1, by = 0.001)
fitridge <- glmnet(tildeX,tildeY, alpha = 0, lambda  = lambda_seq,family=c("gaussian"),intercept=F)
df=data.frame(tau = rep(-log(fitridge$lambda),ncol(tildeX)), theta=as.vector(t(fitridge$beta)),
              variable=rep(colnames(x_var),each=length(fitridge$lambda)))
g1 = ggplot(df,aes(x=tau,y=theta,col=variable))+
  geom_line()+
  ylab('Estimation of theta')+xlab("-log(lambda)")+
  theme(legend.title = element_text(size = 5),legend.text = element_text(size = 3))
g1
```



Example

```
from sklearn.linear_model import Ridge
lambdas=np.arange(0.001,1.01,0.001)
lambdasbis=2*31*lambdas
p=Xtildepy.shape[1]
coefs=np.empty((0,p),float)
for a in lambdasbis:
  ridge = Ridge(alpha=a, fit_intercept=False);
  ridge.fit(Xtildepy, ytildepy);
  coefs=np.append(coefs,ridge.coef_,axis=0);
```



Calibration of λ

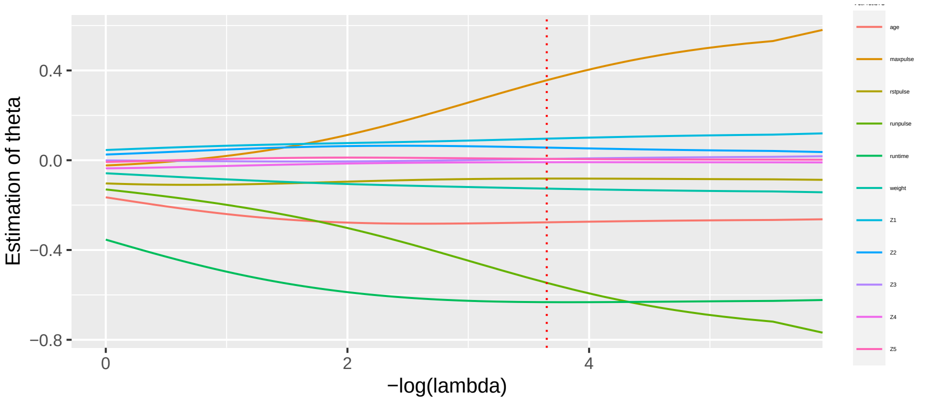
- Calibration by training/test:
  - We start by separating the data into a training set (Y_a, X_a) and a test set (Y_v, X_v).
  - We then estimate the ridge regression on the training set for each value of λ in a chosen grid of values
  - We predict the response on the test set for each value of λ: $\widehat{Y}_{\text{ridge},v}(\lambda)$.
  - The quality of the model is then obtained by comparing the true values Y_v and the predicted values $\widehat{Y}_{\text{ridge},v}(\lambda)$. For example,

    $$PRESS(\lambda) = \|Y_v - \widehat{Y}_{\text{ridge},v}(\lambda)\|^2.$$

  - Finally, we choose the value of λ which minimizes this criterion.
- Cross-validation = repeat the division between test and training several times and consider the average of the values of the criterion for each value of λ.

Example

```
ridge_cv <- cv.glmnet(tildeX, tildeY, alpha = 0, lambda = lambda_seq,nfolds=10, type.measure=c("mse"),intercept
best_lambda <- ridge_cv$lambda.min
g1+geom_vline(xintercept = -log(best_lambda),linetype="dotted",color = "red")+
  xlim(c(0,-log(best_lambda)+2))
```



Outline

5 **Regularized regressions**

- Principle of regularized regression
- Ridge regression
- Lasso regression
- Elastic-Net regression

Lasso estimator

- LASSO = Least Absolute Selection and Shrinkage Operator (Tibshirani,96)

- Idea : set to zero coefficients of θ in order to have a **sparse** estimator

- This induces a variable selection making the model more interpretable and a matrix of explanatory variables with better properties than $X'X$.

- The Lasso estimator is defined for $\lambda > 0$ by

$$\widehat{\theta}_{\text{lasso}}(\lambda) \in \underset{\theta \in \mathbb{R}^p}{\operatorname{argmin}} \|\tilde{Y} - \tilde{X}\theta\|_2^2 + \lambda\|\theta\|_1$$

Lasso regression

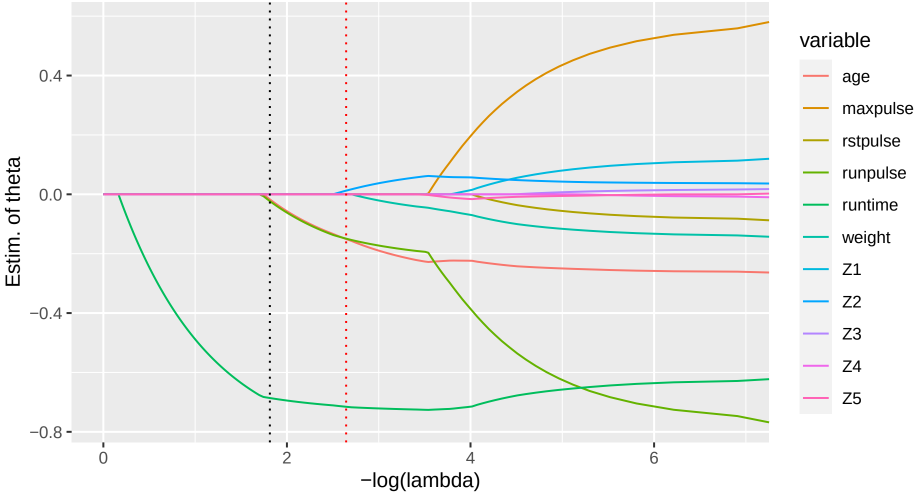
- This minimization problem is equivalent to minimize $\|\tilde{Y} - \tilde{X}\theta\|_2^2$ under the constraint $\|\theta\|_1 \leq r(\lambda)$ with $r(.)$ bijective.
- The solution may not be unique but the vector of the resulting fitted values $\tilde{X}\widehat{\theta}_{\text{lasso}}(\lambda)$ is always unique.
- When $\lambda = 0$, $\widehat{\theta}_{\text{lasso}}(0) = \widehat{\theta}$; when $\lambda \to +\infty$, $\widehat{\theta}_{\text{lasso}}(+\infty) = 0$.
- The choice of λ is tricky, it is impossible to make this choice a priori. We can plot the Lasso *regularization paths*

$$\lambda \mapsto \widehat{\theta}_{\text{lasso}}(\lambda)_j \text{ pour } j = 1, \ldots, p$$

A cross validation procedure is used to stabilize the choice of λ

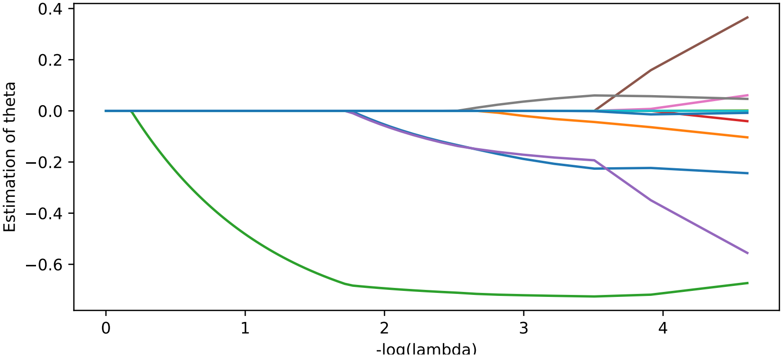
Example

```
lambda_seq=seq(0,1,0.001)
fitlasso <- glmnet(tildeX,tildeY, alpha = 1, lambda  = lambda_seq,family=c("gaussian"),intercept=F)
lasso_cv <- cv.glmnet(tildeX, tildeY, alpha = 1, lambda = lambda_seq,nfolds=10,type.measure=c("mse"),intercept=
best_lambda <-lasso_cv$lambda.min  # red
best_lambda.1se <- lasso_cv$lambda.1se # black
```



Example

```
from sklearn.linear_model import lasso_path
lambdas=np.arange(0.01,1.01,0.01)
alphas_lasso, coefs_lasso, _ = lasso_path(Xtildepy, ytildepy, alphas=lambdas)
```



Outline

5. **Regularized regressions**
   - Principle of regularized regression
   - Ridge regression
   - Lasso regression
   - Elastic-Net regression

Principle

- Elastic-Net regression combines the advantages of ridge regression and lasso regression.

- The Elastic-Net estimator is defined for $\lambda > 0$ and $\alpha > 0$ by

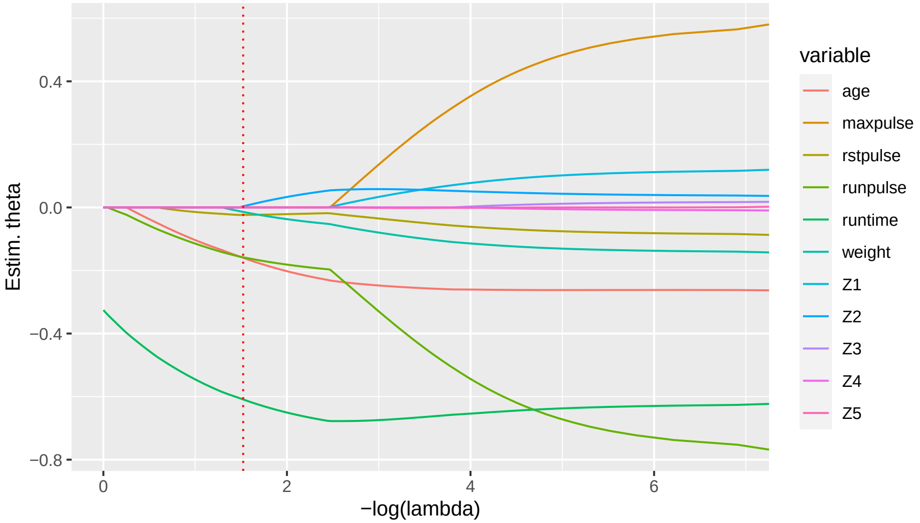
$$\widehat{\theta}_{\text{net}}(\lambda, \alpha) \in \underset{\theta \in \mathbb{R}^p}{\text{argmin}} \|\tilde{Y} - \tilde{X}\theta\|_2^2 + \lambda\{\alpha\|\theta\|_1 + (1-\alpha)\|\theta\|_2^2\}$$

- Use of an optimization algorithm to determine $\widehat{\theta}_{\text{net}}(\lambda, \alpha)$

- Calibration of thresholds λ and α by cross-validation procedure

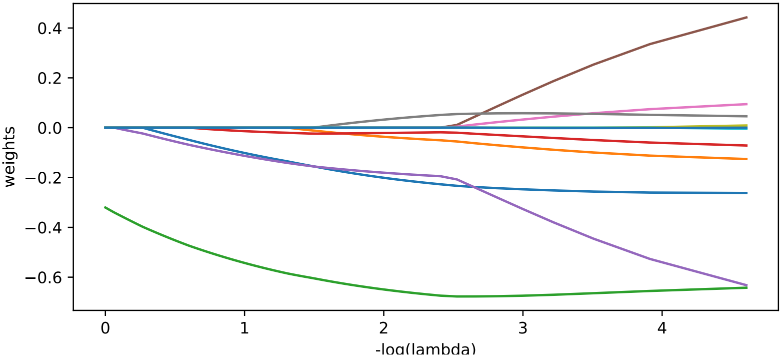
Example

```
fitEN <- glmnet(tildeX,tildeY, alpha = 0.3, lambda  = lambda_seq,family=c("gaussian"),intercept=F)
EN_cv <- cv.glmnet(tildeX, tildeY, alpha = 0.3, lambda = lambda_seq,nfolds=10,type.measure=c("mse"),intercept=F
```



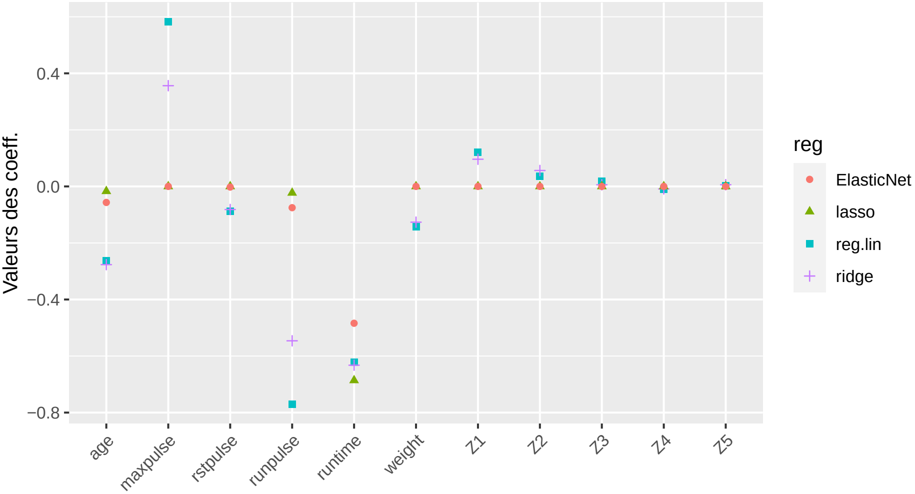
Example

```
from sklearn.linear_model import enet_path
lambdas=np.arange(0.01,1.01,0.01)
alphas_enet, coefs_enet, _ = enet_path(Xtildepy, ytildepy, alphas=lambdas, l1_ratio=0.3)
```



Example

Comparison of the coefficient values for each method:



Outline

1. Introduction

2. Estimation

3. Tests, confidence intervals and prediction intervals

4. Selection of explanatory variables

5. Regularized regressions

6. **Model validation**

Outline

6 Model validation

- A posteriori graphical controls
- To check H_1 and H_2: adequacy and homoscedasticity
- To check hypothesis H_3: independence
- To check hypothesis H_4: Gaussianity
- Detection of outliers / high leverage points

A posteriori graphical controls

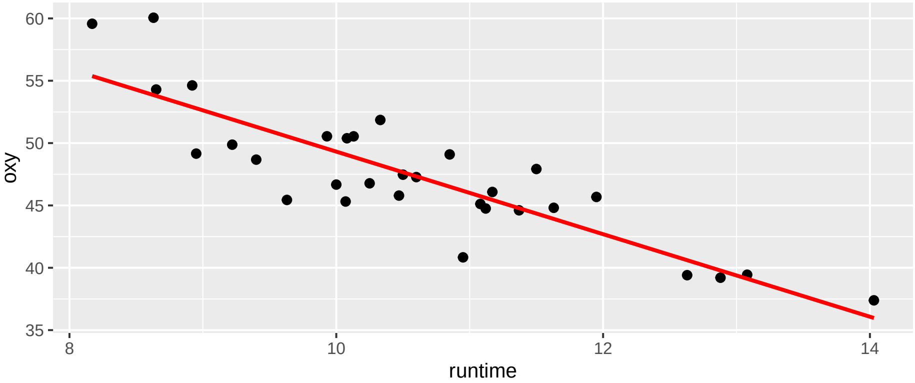
Once the model has been implemented, we can verify *a posteriori* the "statistical validity" of this model by controlling

- the normality hypothesis
- the adequacy of the fitted values $\widehat{Y_i}$ to the observed values Y_i
- the absence of outliers.

It is therefore important to empirically (graphically) control the four fundamental assumptions.

A posteriori graphical controls

- The graphical comparison between the point cloud (z_i, Y_i) and the estimated regression line gives almost exhaustive information.

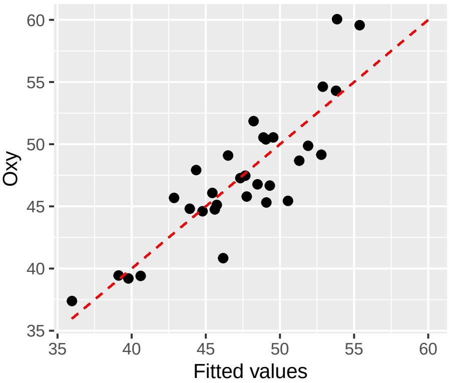


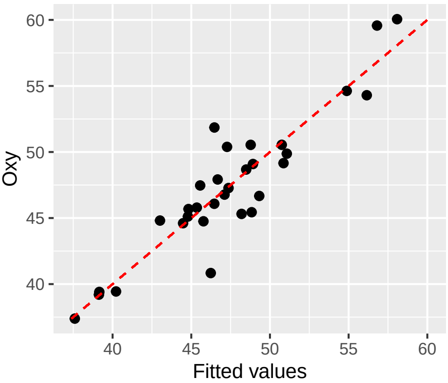
A posteriori graphical controls

- In multiple linear regression, this type of plot cannot be used because there are several regressors. The various hypotheses must therefore be checked on the terms of the errors ε_i which are unfortunately unobservable. We use the residuals $\widehat{\varepsilon_i} = Y_i - \widehat{Y_i}$.

- The plot of the n points $(Y_i, \widehat{Y_i})$ is also very informative. It is then sufficient to check if the points are aligned according to the first bisector.

A posteriori graphical controls





Outline

6 Model validation

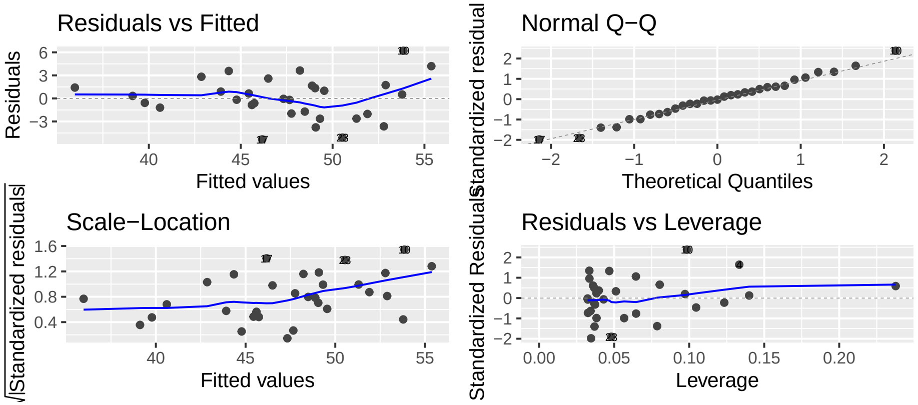
- A posteriori graphical controls
- To check H_1 and H_2: adequacy and homoscedasticity
- To check hypothesis H_3: independence
- To check hypothesis H_4: Gaussianity
- Detection of outliers / high leverage points

Check to H_1 and H_2

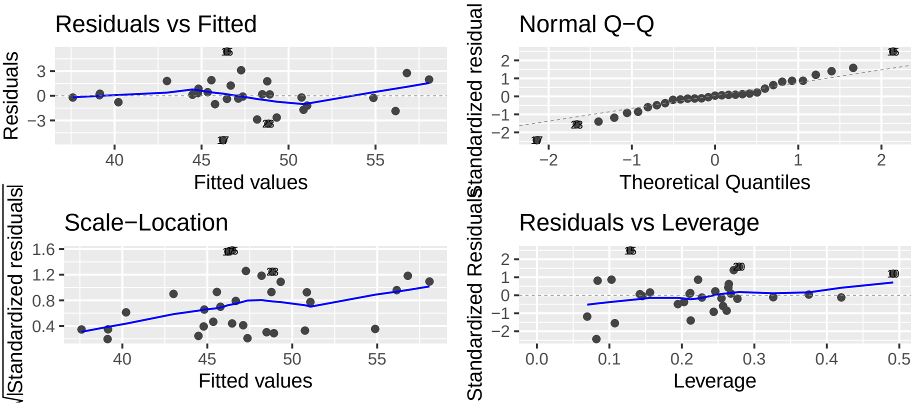
- Plot the residuals $(\widehat{\varepsilon}_i)_i$ against the fitted values $(\widehat{Y_i})_i$.
- If the four hypotheses H1-H4 are satisfied, there is independence between these 2 vectors which are centered and Gaussian (according to Cochran's theorem). However, from this plot, we will only be able to see the possible deficiency of the hypotheses H_1 and H_2.

Example régression simple

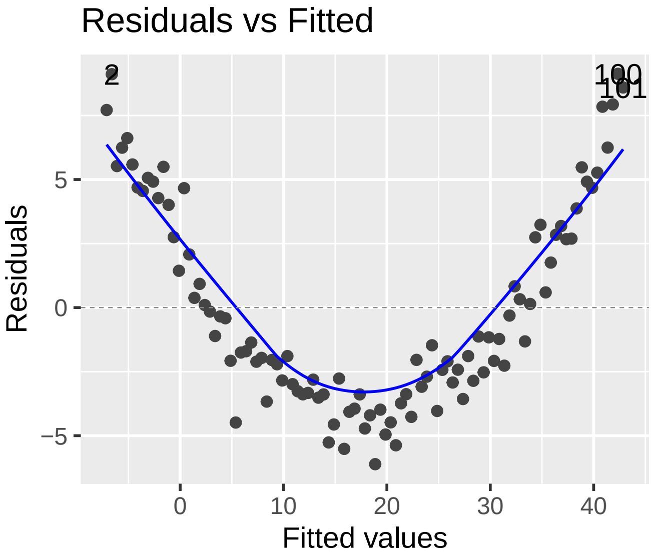
```
autoplot(reg.simple, label.size = 2)
```



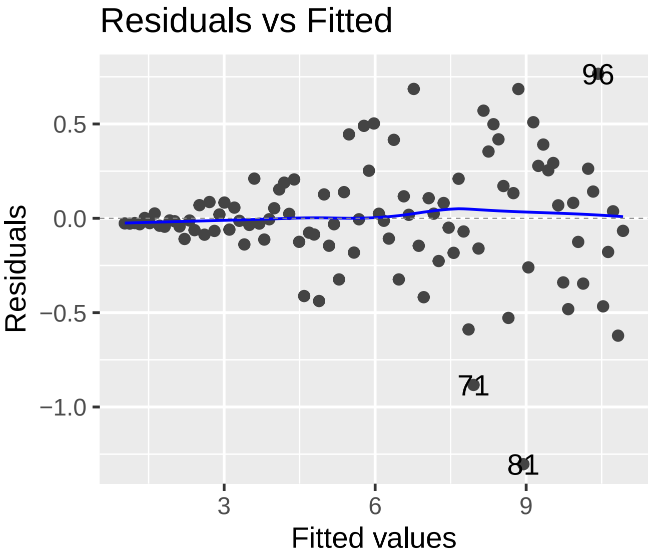
Example régression multiple



Pathological case: “en banane”



Pathological case: “en trompette”



Possible modifications to the model

- We can freely transform the regressors $z^{(1)}, \cdots, z^{(p)}$ by all known algebraic or analytical transformations provided that the new model remains interpretable.

- On the other hand, we can only consider transforming Y if the graphs suggest heteroskedasticity.

Relationship	Definition field of Y	Transformation
$\sigma = (cste)Y^k,\ k \neq 1$	$\mathbb{R}^{+*}$	$Y \mapsto Y^{1-k}$
$\sigma = (cste)\sqrt{Y}$	$\mathbb{R}^{+*}$	$Y \mapsto \sqrt{Y}$
$\sigma = (cste)Y$	$\mathbb{R}^{+*}$	$Y \mapsto \log(Y)$
$\sigma = (cste)Y^2$	$\mathbb{R}^{+*}$	$Y \mapsto Y^{-1}$
$\sigma = (cste)\sqrt{Y(1-Y)}$	$[0;1]$	$Y \mapsto \arcsin\sqrt{Y}$
$\sigma = (cste)\sqrt{1-Y}Y^{-1}$	$[0;1]$	$Y \mapsto (1-Y)^{\frac{1}{2}} - \frac{1}{3}(1-Y)^{\frac{3}{2}}$
$\sigma = (cste)(1-Y)^{-2}$	$[-1;1]$	$Y \mapsto \log(1+Y) - \log(1-Y)$

Outline

6 **Model validation**

- A posteriori graphical controls
- To check H_1 and H_2: adequacy and homoscedasticity
- To check hypothesis H_3: independence
- To check hypothesis H_4: Gaussianity
- Detection of outliers / high leverage points

To check hypothesis H_3: independence

- Plot of residuals $\widehat{\varepsilon}_i$ as a function of data order (when it makes sense, especially if it represents time).

- It is suspect if the residuals tend to be grouped when they are on either side of 0.

- We can confirm these doubts by performing a test runs. This test is based on the number of runs, i.e. on the number of groups of consecutive residuals with the same sign.

Outline

6 **Model validation**

- A posteriori graphical controls
- To check H_1 and H_2: adequacy and homoscedasticity
- To check hypothesis H_3: independence
- To check hypothesis H_4: Gaussianity
- Detection of outliers / high leverage points

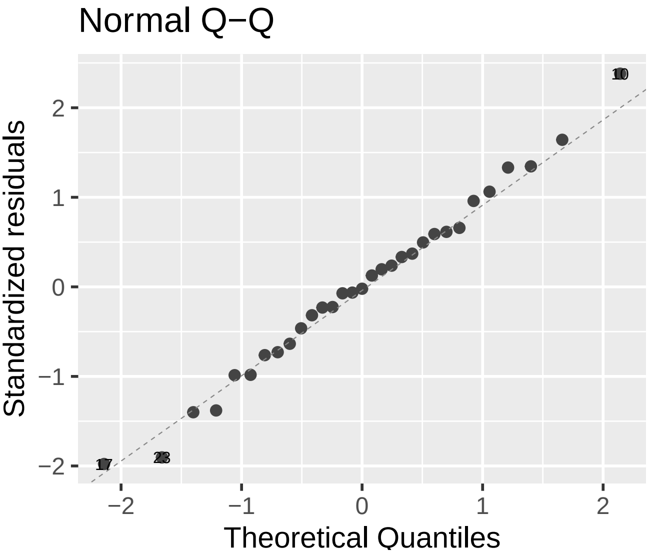
To check hypothesis H_4: Gaussianity

- Avoid the usual suitability tests of Kolmogorov-Smirnov, Cramer-Von Mises, ..., because they will be applied to the residuals $\widehat{\varepsilon}_i$, which are (almost) never independent.

- Graphical check from the Henry line (particular case of the so-called QQ-plot): the studentized residuals are represented as a function of the theoretical quantiles of a reduced centered normal distribution. This type of plot allows above all to see if a “heavy distribution tail” law could not be more adequate (in this case, the points move away from Henry’s line at its extremities).

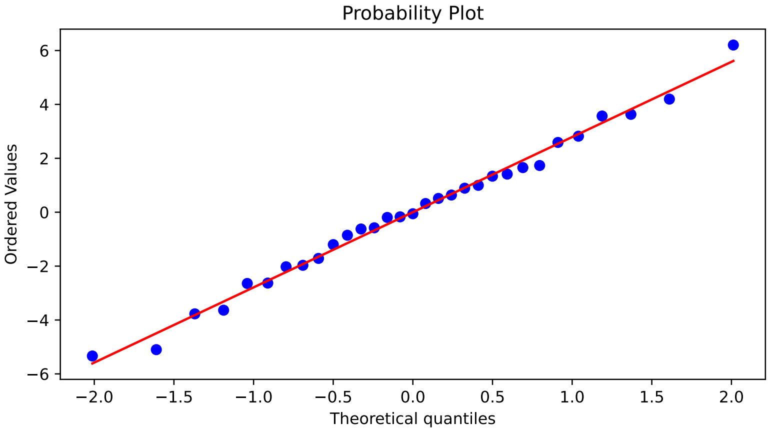
Example (simple regression)

```
autoplot(reg.simple, label.size = 2,which=c(2))
```



Example (simple regression)

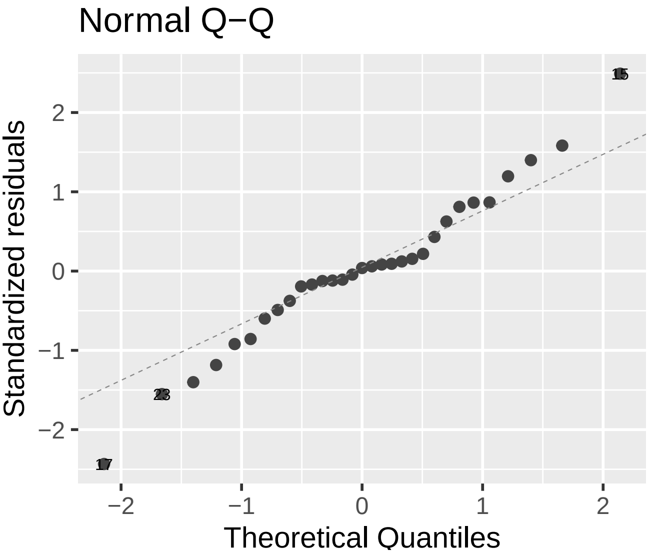
```
from scipy import stats
plt.figure(figsize=(8,4));
stats.probplot(resultsregsimple.resid, dist="norm", plot= plt);
plt.show()
```



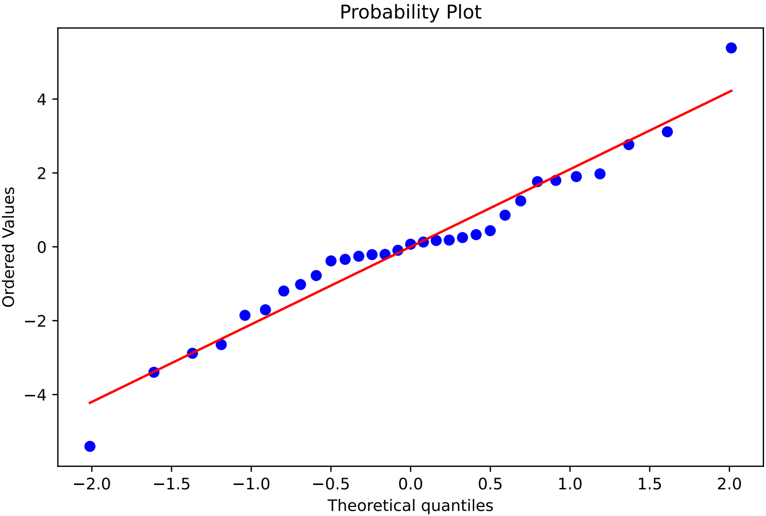
```
plt.close()
```

Example (multiple regression)

```
autoplot(reg.multi, label.size = 2,which=c(2))
```



Example (multiple regression)



Outline

6 **Model validation**

- A posteriori graphical controls
- To check H_1 and H_2: adequacy and homoscedasticity
- To check hypothesis H_3: independence
- To check hypothesis H_4: Gaussianity
- Detection of outliers / high leverage points

Leverage effect

- Hat matrix: $H = P_{[X]} = X(X'X)^{-1}X'$
- Prediction for the i-th individual:

$$\hat{Y}_i = (X\hat{\theta})_i = (HY)_i = H_{ii}Y_i + \sum_{j\neq i} H_{ij}Y_j.$$

  - If $H_{ii} = 1$: $\hat{Y}_i$ is entirely determined by the i-th observation
  - If $H_{ii} = 0$: the i-th observation has no influence on $\hat{Y}_i$

- Thus, to measure the influence of an observation on its own estimate, one can examine the bar diagram of the diagonal terms of H. In practice, the i-th observation is a **high leverage point** if H_{ii} exceeds $2k/n$ or $3k/n$.

Cook's distance

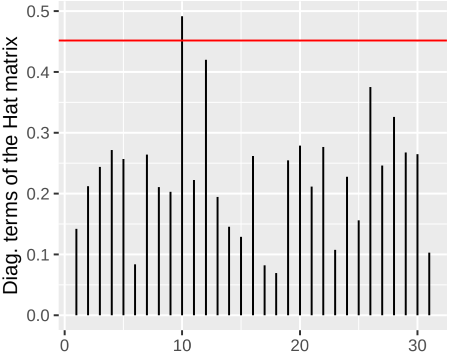
- The influential points are the points such that, if we remove them from the study, the estimate of the model coefficients will be strongly modified.
- The most usual measure of influence is the Cook's distance:

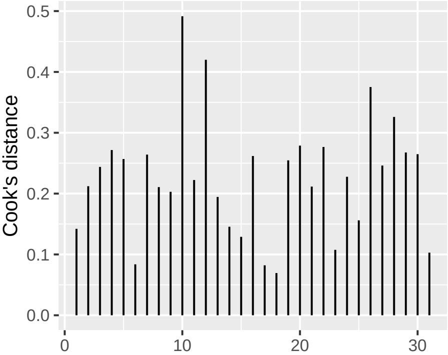
  $$DC_i = (\widehat{\theta} - \widehat{\theta}^{(-i)})' T' T (\widehat{\theta} - \widehat{\theta}^{(-i)})$$

  where T is the vector of studentized residuals and $\widehat{\theta}^{(-i)}$ is the estimator without the i-th observation.

Here again we can draw the barplot of the values DC_i. If a distance turns out to be great compared to the others then this point will be considered as influential. We must therefore seek to understand why it is influential: it is leverage, outlier . . . .

Example (multiple regression)





References I

[1] Hirotugu Akaike. “A Bayesian analysis of the minimum AIC procedure”. In: *Annals of the Institute of Statistical Mathematics* 30.1 (1978), pp. 9–14.

[2] Lucien Birgé and Pascal Massart. “Gaussian model selection”. In: *Journal of the European Mathematical Society* 3.3 (2001), pp. 203–268.

[3] Jean-Jacques Daudin. *Le modèle linéaire et ses extensions-Modèle linéaire général, modèle linéaire généralisé, modèle mixte, plans d’expériences (Niveau C)*. 2015.

[4] X Guyon. “Modele linéaire et économétrie”. In: *Ellipse, Paris* (2001).

[5] Arthur E Hoerl, Robert W Kannard, and Kent F Baldwin. “Ridge regression: some simulations”. In: *Communications in Statistics-Theory and Methods* 4.2 (1975), pp. 105–123.

[6] Gareth James et al. *An introduction to statistical learning*. Vol. 112. Springer, 2013.

References II

[7] Colin L Mallows. “Some comments on Cp”. In: *Technometrics* 42.1 (2000), pp. 87–94.

[8] Bernard Prum. *Modèle linéaire: Comparaison de groupes et régression*. INSERM, 1996.

[9] Gideon Schwarz et al. “Estimating the dimension of a model”. In: *The annals of statistics* 6.2 (1978), pp. 461–464.

[10] Robert Tibshirani. “Regression shrinkage and selection via the lasso”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1 (1996), pp. 267–288.

[11] Hui Zou and Trevor Hastie. “Regularization and variable selection via the elastic net”. In: *Journal of the royal statistical society: series B (statistical methodology)* 67.2 (2005), pp. 301–320.