Régression linéaire simple en R
Exemple détaillé

Préambule : récupération et mise en forme des données

On utilise la base `forbes` de la librairie `MASS`.

```
> library(MASS)
> data(forbes)
```

On rappelle que pour avoir des informations sur les données contenues dans le package on utilise :

```
> help(forbes)
```

Ce paquet contient deux variables : la température d'ébullition de l'eau en degrés Farenheit et la pression barométrique en pouces de mercures. La commande

```
> attach(forbes)
```

permet de manipuler les variable contenues dans `forbes` par leur noms respectifs, ici `bp` et `pres`.
Afin de mieux comprendre ce que nous manipulons, nous allons considérer dans la suite la température en degré Celcius et l'altitude en mètres correspondant à la pression observée. Ceci est réalisé à l'aide des instructions :

```
> temp <- -160/9+5/9*bp
> alt <- 8170-274.5*pres
```

Nous créons ensuite un `data.frame` avec ces nouvelles variables :

```
> ebullition <- data.frame(alt=alt,temp=temp)
```

Les commandes suivantes permettent respectivement de résumer le contenu des données de `ebullition` et de donner les principales statistiques descriptives sur les variables :

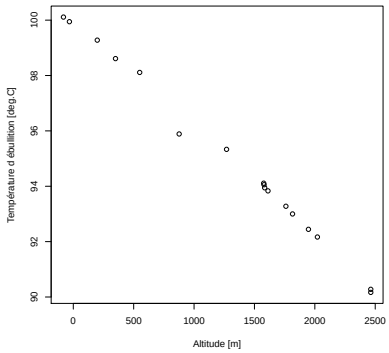
```
> str(ebullition)
> summary(ebullition)
> var(ebullition)
```

1 Modèle de régression

On souhaite expliquer la température `temp` par l'altitude `alt`.

- Représentation graphique

```
> plot(ebullition, main="",xlab="Altitude [m]",
                   ylab="Température d'ébullition [deg.C]")
```



Le graphique justifie ici pleinement la régression linéaire.

- On cherche à étudier la relation linéaire entre ces deux variables. On utilise pour cela la fonction `lm` de R.

  ```
  > eau.lm <- lm(temp∼ alt)
  ```

- Les différentes grandeurs calculées sont données par `names(eau.lm)`. On peut notamment extraire :
  - les prédictions : `fitted(eau.lm)` ou `predict(eau.lm)`,
  - les coefficients du modèle : `coef(eau.lm)`,
  - les résidus : `resid(eau.lm)`,
  - les résidus studentisés : `stdres(eau.lm)`. Cette commande nécessite le paquet `MASS`.

  Pour chacune des 3 premières commandes, on peut également utiliser respectivement `eau.lm$fitted.values`, `eau.lm$coefficients` ou `eau.lm$resid`. Par exemple ici, `coef(eau.lm)` retourne

  ```
   (Intercept)            alt
  99.944254402   -0.003848985
  ```

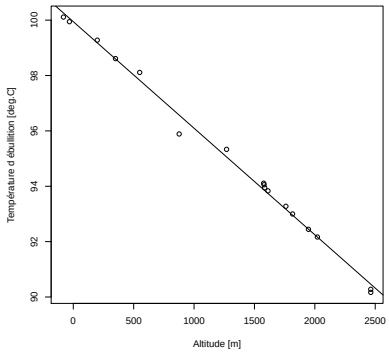
- Si l'on souhaite accéder à des grandeurs non citées par `names(eau.lm)`, elles sont a priori données par `summary(eau.lm)`. Vous pouvez le vérifier à l'aide de `names(summary(eau.lm))`. La syntaxe est alors la suivante :

  ```
  > summary(eau.lm)$r.squared
  ```

  qui retourne ici `0.9944282`. Le R^2 est donc très élevé, ce qui est cohérent avec le nuage de points représenté initialement et justifie un modèle linéaire.

- On trace la droite de régression estimée par le modèle linéaire sur le nuage de points.

  ```
  > abline(eau.lm)
  ```



La droite de régression linéaire semble bien correspondre au nuage de points.

2 Principales statistiques

- Nous détaillons ici la sortie de

  `> summary(eau.lm)`

(1)
```
Call:
lm(formula = temp ~ alt)
```

(2)
```
Residuals:
     Min       1Q   Median       3Q      Max
-0.6816 -0.1232  0.0429  0.1094  0.2833
```

(3)
```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.994e+01  1.132e-01  883.12   <2e-16 ***
alt         -3.849e-03  7.439e-05  -51.74   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

(4)
```
Residual standard error: 0.2467 on 15 degrees of freedom
Multiple R-squared:  0.9944,        Adjusted R-squared:  0.9941
F-statistic:  2677 on 1 and 15 DF,  p-value: < 2.2e-16
```

  On retrouve dans cette sortie :

  (1) La formule initiale associée à la sortie.

  (2) Des statistiques descriptives sur les résidus.

  (3) Les estimations des paramètres, de leur écart-type, la valeur de la statistique de student associée, la p-valeur du test (H_0) coeff=0 contre (H_1) coeff≠0.

  (4) L'écart-type estimé des résidus.
  Les coefficients de détermination, avec et sans ajustement.
  Le test de pertinence de Fisher : valeur de la statistique et p-valeur.

  Dans cet exemple, nous pouvons constater notamment que la constante et le coefficient de la pente sont significatifs et que le test de Fisher ne rejette pas le modèle linéaire.

- Nous pouvons également obtenir la table d'analyse de la variance :

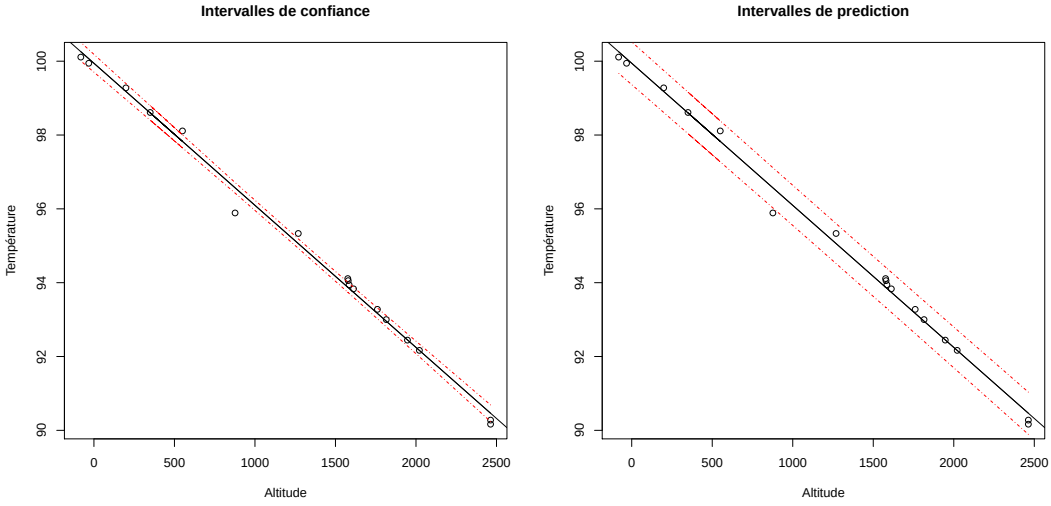
  `> anova(eau.lm)`

```
Analysis of Variance Table

Response: temp
          Df  Sum Sq Mean Sq F value    Pr(>F)
alt        1 162.909 162.909  2677.1 < 2.2e-16 ***
Residuals 15   0.913   0.061
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

- Représentation des intervalles de confiance et de prédiction.

```
> ICconf <- predict(eau.lm, interval = "confidence", level = 0.95)
> matlines(alt, ICconf, lty = c(1, 4, 4), col = c(1, 2, 2))
> ICpred <- predict(eau.lm, interval = "prediction", level = 0.95)
> matlines(alt, ICpred, lty = c(1, 4, 4), col = c(1, 3, 3))
```



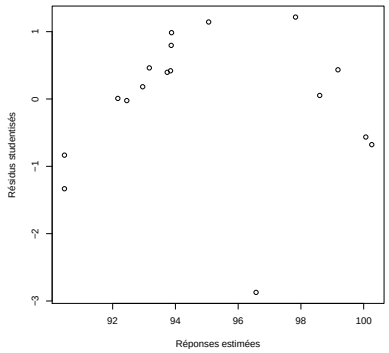
3 Etude des résidus

- On commence par étudier l'adéquation du modèle ainsi que le caractère identiquement distribué. En particulier, l'homogénéité de la variance.
  On représente les résidus studentisés en fonction des $(\widehat{y_i})$:

```
> plot(fitted(eau.lm),stdres(eau.lm),main="",xlab="Réponses estimées",
      ylab="Résidus studentisés")
```

  ou encore

```
> plot(eau.lm,which=1)
```



Nous validons l'hypothèse d'indépendance (même si une légère corrélation apparaît). L'homogénéité de la variance est également vérifiée graphiquement.

- On teste ensuite l'hypothèse de normalité.

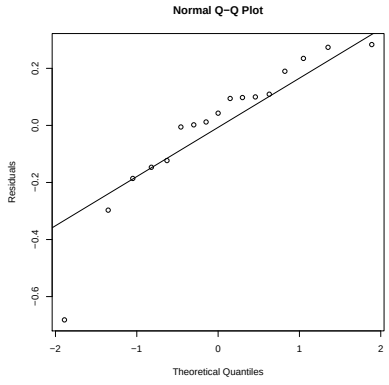
  ```
  > shapiro.test(resid(eau.lm))
  ```

  On obtient une p-valeur de 3.36%. On rejette l'hypothèse de normalité avec un seuil de 5%. L'hypothèse gaussienne n'est pas vérifiée.
  On peut également représenter un QQ-plot afin de vérifier l'hypothèse gaussienne

  ```
  > qqnorm(residuals(eau.lm), ylab="Residuals")
  > qqline(residuals(eau.lm))
  ```

  ou encore

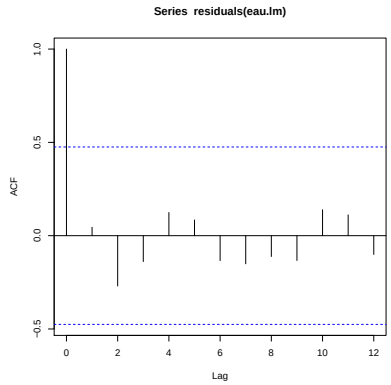
  ```
  plot(eau.lm,which=2)
  ```



- On a également fait l'hypothèse d'indépendance des résidus. On peut représenter l'autocorrélation des résidus

  ```
  > acf(residuals(eau.lm))
  ```

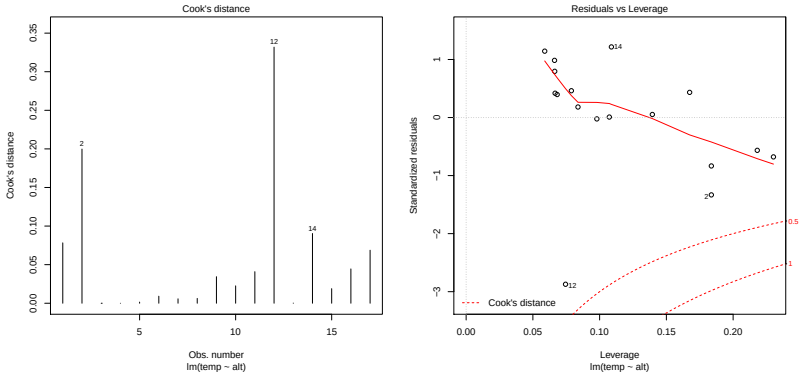
  Si une barre, exeptée la première, dépasse des seuils en pointillés, l'indépendance est remise en cause. Des tests (Durbin Watson notamment) sont possibles mais non traités ici.



L'hypothèse d'indépendance n'est pas mise en défaut.

- On peut vérifier la présence de points ayant une forte contribution à l'aide de la fonction influence `influence()` ou de la distance de Cook.

```
> cooks.distance(eau.lm)
> plot(eau.lm,which=4)
> plot(eau.lm,which=5)
```

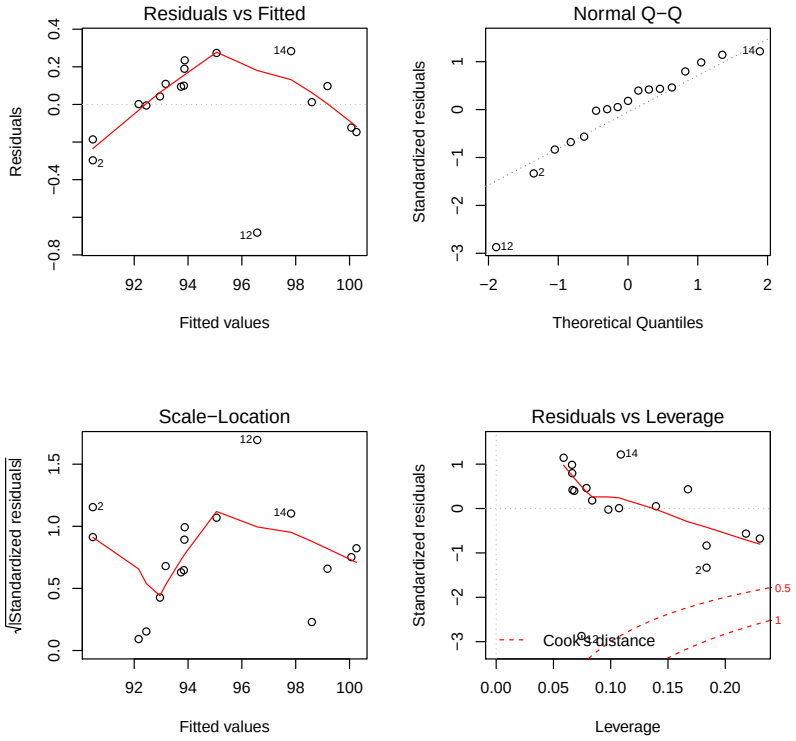


Les distances sont inférieures à 1, donc il n'y a pas de point perturbant l'estimation. Cependant le point 12 semble atypique. Si nous l'enlevons nous pouvons en effet constater que l'hypothèse de normalité est vérifiée :

```
> eau.lm.new <- lm(temp ~ alt, subset=(1:17)[-12])
> shapiro.test(resid(eau.lm.new))
```

retourne une p-valeur de 40.83%.

Notons que les 4 principaux graphiques fournis par la fonction `plot(eau.lm)` sont les suivants :



Ce qu'on attend de ces graphiques :

— *Residuals vs Fitted* : le nuage de points doit être sans structure

— *Normal Q-Q* : les points doivent être proches de la bissectrice

— *Scale-Location* : le nuage de points doit être sans structure

— *Cook's distance* : les points doivent être entre les lignes pointillées correspondant à 1 et il faut regarder les points au-dessus.

Bilan

Pour ajuster un modèle linéaire simple gaussien $y_i = a\,x_i + b + \varepsilon_i$
`> modele <-lm(y ~ x)`

Pour voir les principales statistiques du modèle
`> summary(modele)`

Pour étudier la validité des hypothèses

- Adéquation et homoscédasticité : `plot(fitted(modele),stdres(modele))`
- Indépendance : `acf(res(modele))`
- Normalité : `shapiro.test(resid(modele))` et QQ-plot
- Points aberrants : `cooks.distance(modele)`