TP Regression : temperature sensor

S. Jaubert

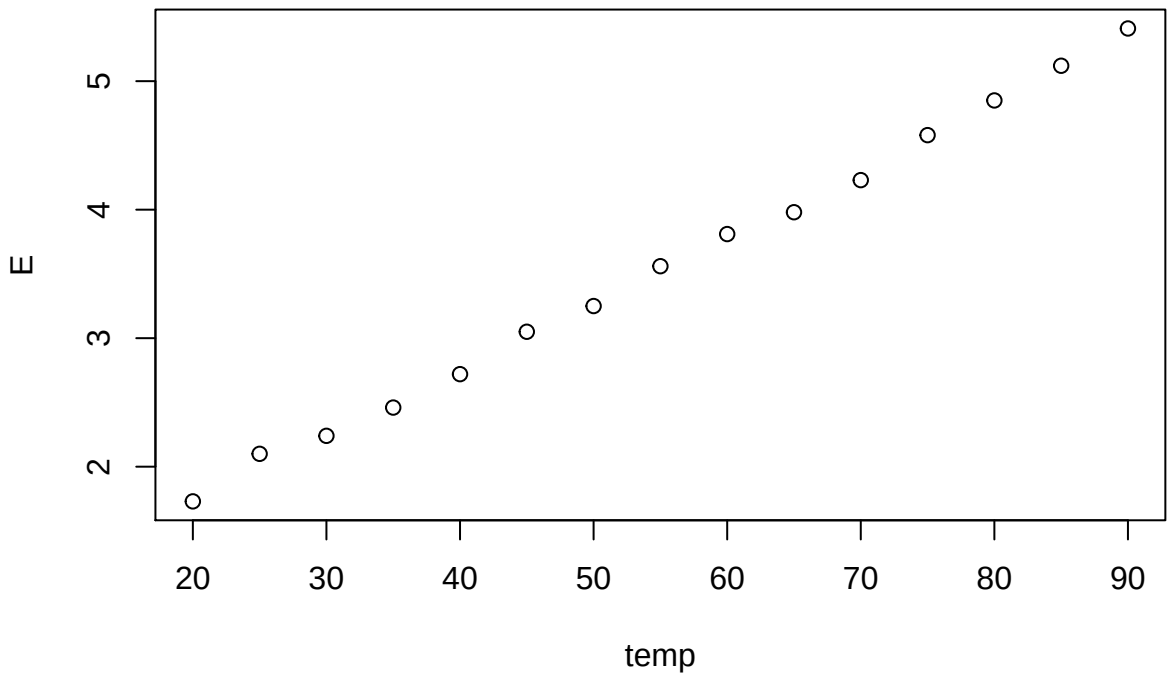
04 janvier 2021

Un capteur de température est utilisé pour mesurer la température entre 20°C et 150°C. La sortie du capteur est conditionnée par un circuit électronique approprié pour obtenir la production E en Volt.

Les données ci-dessous ont été recueillies en laboratoire en exposant le capteur à un environnement thermique approprié dont la température a été systématiquement variée et contrôlée par un capteur standard.

```
temp<-seq(from=20,to=90,by=5)
E<-c(1.73,2.10,2.24,2.46,2.72,3.05,3.25,3.56,3.81,3.98,4.23,4.58,4.85,5.12,5.41)
temp_sensor<-data.frame(temp=temp,E=E)

plot(x=temp,y=E)
```



Rechercher un modèle linéaire semble bien justifié !

1. Calculer les paramètres suivants :

$\sigma_t^2 = \mathbb{E}(t^2) - \mathbb{E}(t)^2$

$\sigma_E^2 = \mathbb{E}(E^2) - \mathbb{E}(E)^2$

$\sigma_{tE} = \mathbb{E}(tE) - \mathbb{E}(E)\mathbb{E}(t)$

2. Déterminer la pente du modèle linéaire $E = \alpha.temp + b$ ainsi que son "intercept" puis faire sa représentation graphique

$\alpha = \frac{\sigma_{tE}}{\sigma_t^2}$

$b = \mathbb{E}(E) - \alpha.\mathbb{E}(temp)$

3. Retrouver le coefficient de détermination R^2 directement et en détaillant les calculs

Pour rappel :

$$\begin{array}{ccccc} \sum_i (Y_i - \bar{Y})^2 & = & \sum_i (Y_i - \hat{Y}_i)^2 & + & \sum_i (\hat{Y}_i - \bar{Y})^2 \\ SCT & = & SCres & + & SCreg \end{array}$$

et $R^2 = \frac{SCreg}{SCT}$

4. Déterminer la moyenne et la variance des erreurs $\epsilon_i = y_i - \hat{y}_i$

5. Illustrer graphiquement la distribution des ϵ_i pouvons nous considérer qu'ils suivent une loi normale ?

6. Représenter les valeurs ajustées en fonction des valeurs observées

7. Prévoyez le voltage en sortie pour pour des températures de 100°C, 110°C et 120°C

Correction

1.

$\sigma_t^2 = \mathbb{E}(t^2) - \mathbb{E}(t)^2$

obtenu directement par :

```
var(temp)*14/15
```

```
## [1] 466.6667
```

ou par :

```
mean(temp^2)-mean(temp)^2
```

```
## [1] 466.6667
```

idem pour σ_E^2 :

```
var(E)*14/15
```

```
## [1] 1.249113
```

Pour $\sigma_{tE} = \mathbb{E}(tE) - \mathbb{E}(E)\mathbb{E}(t)$:

```
cov(temp,E)*14/15
```

```
## [1] 24.12
```

ou :

```
mean(temp*E)-mean(temp)*mean(E)
```

```
## [1] 24.12
```

2. Déterminer la pente du modèle linéaire $E = \alpha.temp + b$ ainsi que son "intercept"

Pour cette question nous pouvons directement l'obtenir avec R

```
reg<-lm(E~temp,data = temp_sensor)
summary(reg)
```

```
##
## Call:
## lm(formula = E ~ temp, data = temp_sensor)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.084619 -0.037476  0.006952  0.024095  0.111238
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.6966190  0.0375778   18.54 9.88e-11 ***
## temp        0.0516857  0.0006359   81.28  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05321 on 13 degrees of freedom
## Multiple R-squared:  0.998,  Adjusted R-squared:  0.9979
## F-statistic:  6606 on 1 and 13 DF,  p-value: < 2.2e-16
```

Nous obtenons des statistiques sur les résidus, avec le minimum, le maximum et les 3 quartiles, ainsi que des statistiques sur les coefficients obtenus : leur valeur, leur écart-type, la statistique de test de Student, et la p-valeur (le test effectué sur le paramètre est ici le test de significativité : le paramètre vaut 0 versus le paramètre est différent de 0). Les p-valeurs sont très faibles. À un niveau de test de 5 %, on rejette donc l'hypothèse selon laquelle le paramètre est égal à 0 : les paramètres sont donc significativement différents de 0. Ici, on voit que les variables temp et intercept sont significatives. Quant au R^2 , il est de l'ordre de 0.998 ceci est logique au vu de la dispersion du nuage de points originel.

Nous pouvons avoir facilement l'intervalle de confiance des paramètres ;

```
confint(reg,level = 0.95)
```

```
##                 2.5 %     97.5 %
## (Intercept) 0.61543724 0.77780085
## temp        0.05031185 0.05305957
```

Bien sûr nous pouvons obtenir ces résultats par les calculs :

```
(alpha<-cov(temp,E)/var(temp))
```

```
## [1] 0.05168571
```

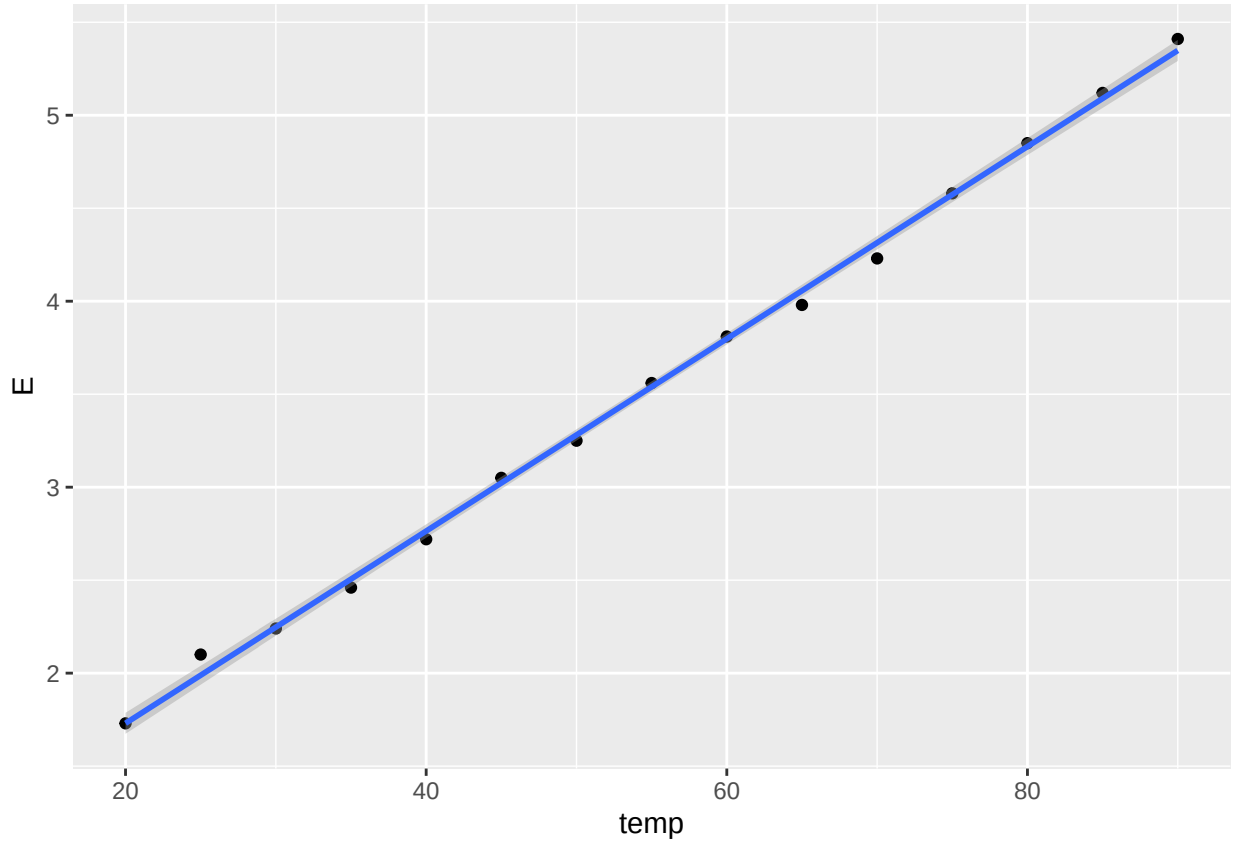
```
(b<-mean(E)-alpha*mean(temp))
```

```
## [1] 0.696619
```

Représentons à présent cette droite :

```
library(ggplot2) #il est plus esthétique d'utiliser la librairie ggplot2
ggplot(temp_sensor,aes(x=temp,y=E))+
  geom_point()+
  stat_smooth(method="lm")+
  xlab("temp")+
  ylab("E")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



(nous pourrions faire plus simplement avec le code :

plot(x = temp,y = E,type="p") abline(reg,col="red"))

3. Retrouver le coefficient de détermination R^2 directement et en détaillant les calculs

Nous allons retrouver le R-squared: 0.998

Pour rappel :

$$\begin{array}{ccccc} \sum_i (Y_i - \bar{Y})^2 & = & \sum_i (Y_i - \hat{Y}_i)^2 & + & \sum_i (\hat{Y}_i - \bar{Y})^2 \\ SCT & = & SCres & + & SCreg \end{array}$$

La somme des carrés des résidus :

```
E_hat<-reg$fitted.values
(SCres<-sum((E-E_hat)^2))
```

```
## [1] 0.0368019
```

La somme des carrés expliqués par la régression :

```
 (Screg<-sum((E_hat-mean(E))^2))
```

```
## [1] 18.69989
```

La somme des carrés totaux:

```
 (ScT<-sum((E-mean(E))^2))
```

```
## [1] 18.73669
```

qui est bien égal à

```
0.0368019 + 18.69989
```

```
## [1] 18.73669
```

On a alors :

```
(R2<-Screg/ScT)
```

```
## [1] 0.9980358
```

4. Déterminer la moyenne et la variance des erreurs ϵ_i

Les résidus sont obtenus par :

```
reg$residuals
```

```
##             1             2             3             4             5
## -0.0003333333  0.1112380952 -0.0071904762 -0.0456190476 -0.0440476190
##             6             7             8             9            10
##  0.0275238095 -0.0309047619  0.0206666667  0.0122380952 -0.0761904762
##            11            12            13            14            15
## -0.0846190476  0.0069523810  0.0185238095  0.0300952381  0.0616666667
```

ou par la différence entre les valeurs observées et celles prédites :

```
(epsilon<-E-E_hat)
```

```
##             1             2             3             4             5
## -0.0003333333  0.1112380952 -0.0071904762 -0.0456190476 -0.0440476190
##             6             7             8             9            10
##  0.0275238095 -0.0309047619  0.0206666667  0.0122380952 -0.0761904762
##            11            12            13            14            15
## -0.0846190476  0.0069523810  0.0185238095  0.0300952381  0.0616666667
```

Moyenne et variance des ϵ :

```
mean(epsilon)
```

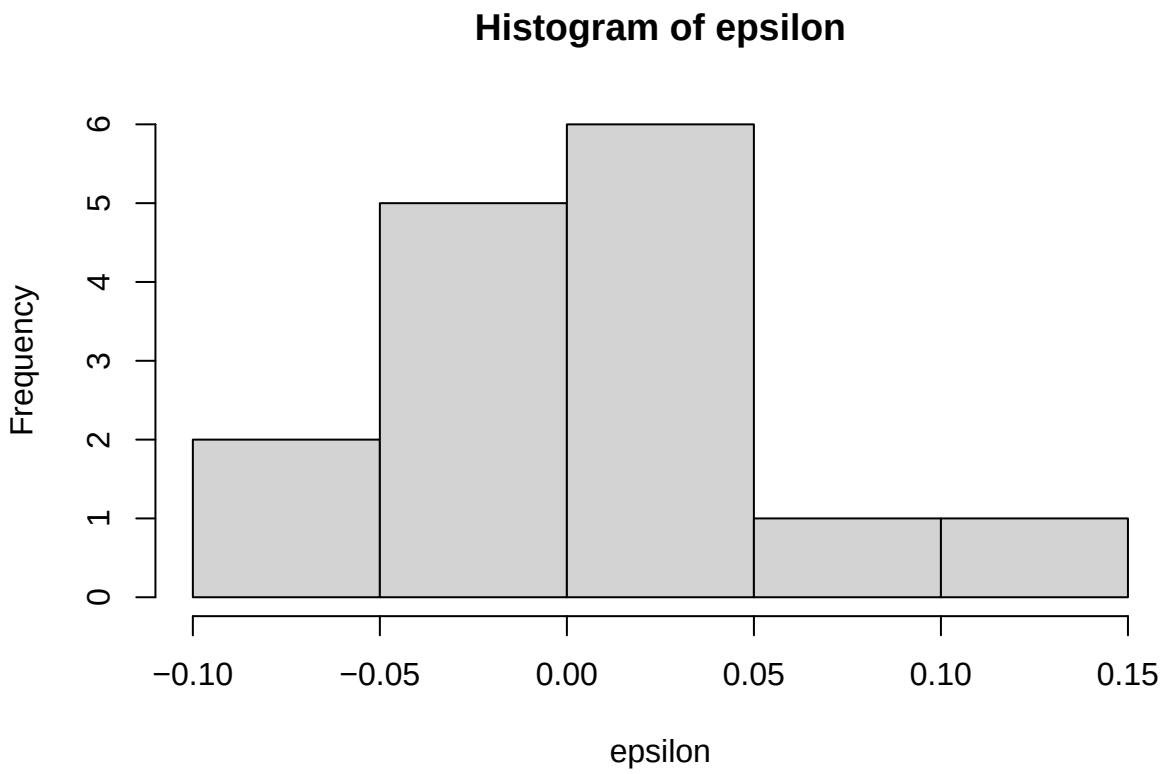
```
## [1] -8.881875e-17
```

```
var(epsilon)
```

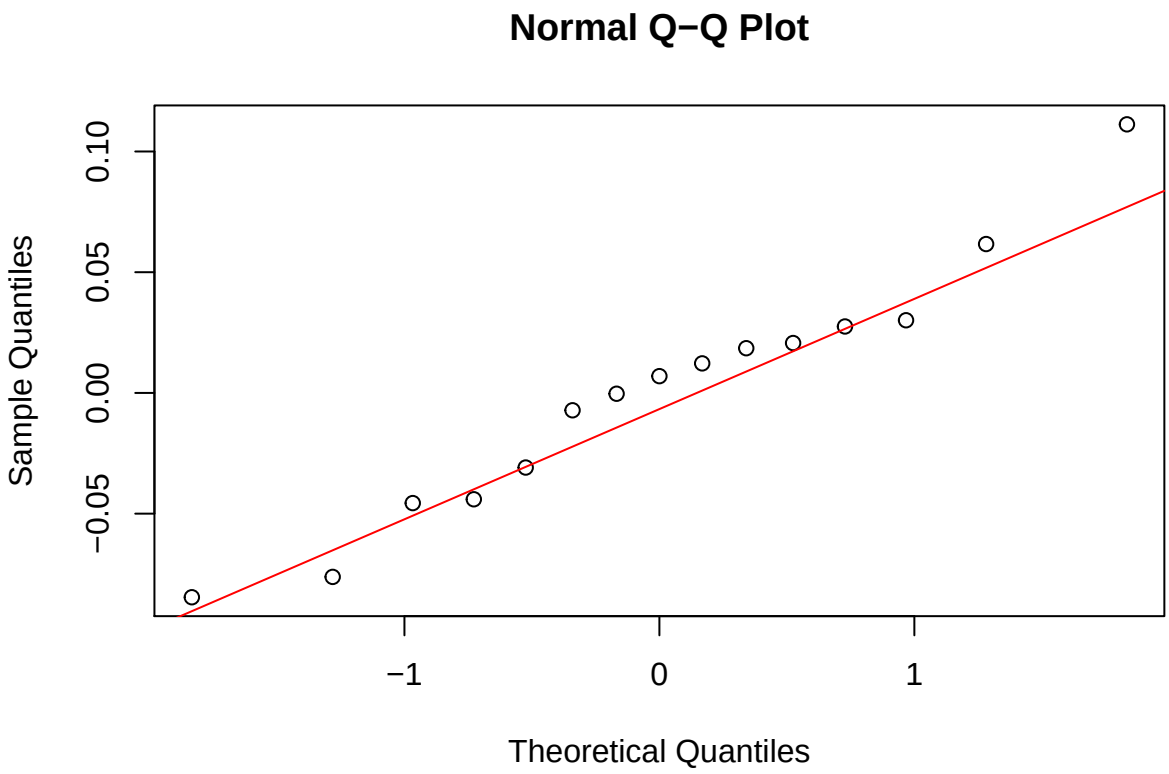
```
## [1] 0.002628707
```

5. Illustrer graphiquement la distribution des ϵ_i pouvons nous considérer qu'ils suivent une loi normale ?

```
hist(epsilon)
```




```
qqnorm(epsilon);qqline(epsilon,col="red")
```



Graphiquement la représentation ne nous permet pas de rejeter l'hypothèse de normalité, un test est cependant préférable :

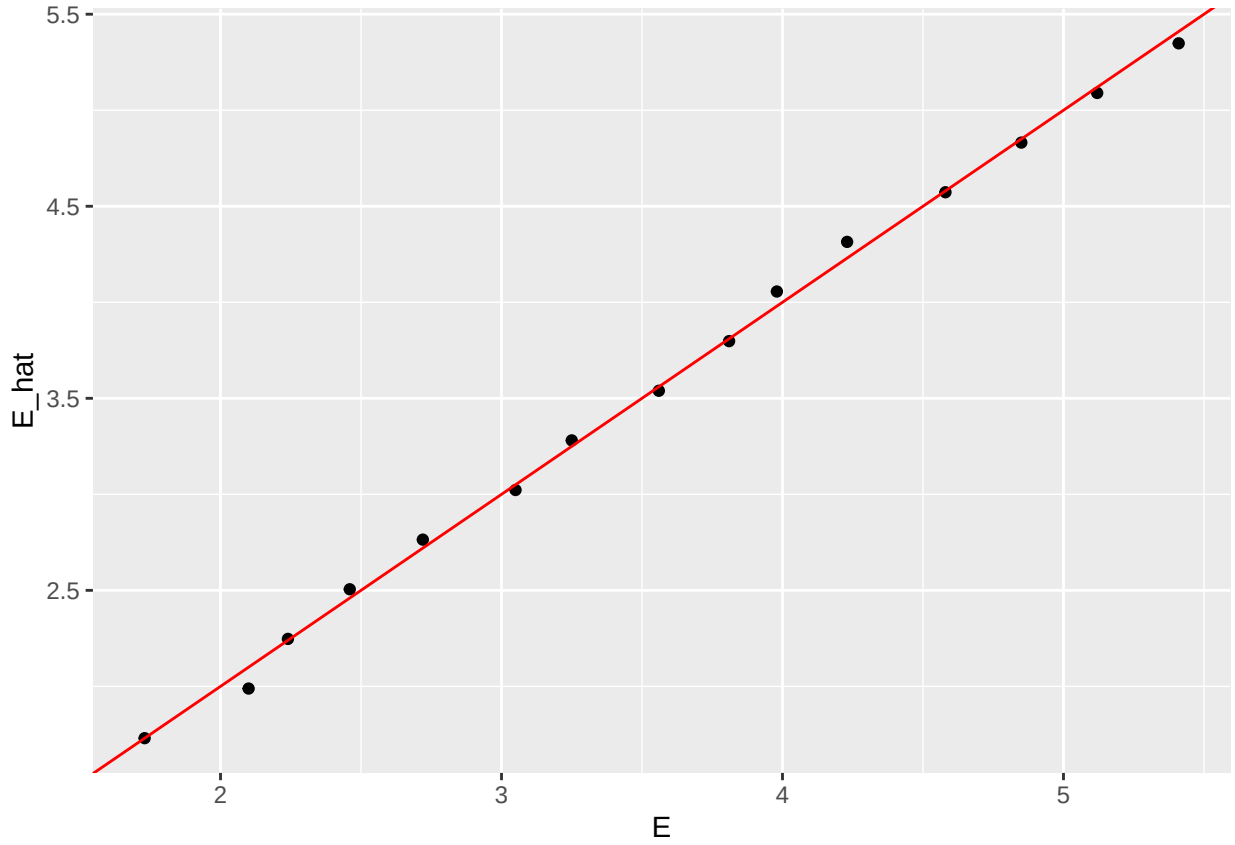
```
shapiro.test(epsilon)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  epsilon
## W = 0.9663, p-value = 0.8
```

la p-value est trop importante pour rejeter l'hypothèse de normalité.

6. Représenter les valeurs ajustées en fonction des valeurs observées

```
temp_sensor$E_hat<-E_hat
ggplot(temp_sensor, aes(x=E,y=E_hat))+
  geom_point()+
  geom_abline(intercept=0,slope=1,color="red")+
  xlab("E")+
  ylab("E_hat")
```



La droite qui s'affiche est la première bissectrice. Comme on peut le voir le modèle est très bon, les valeurs réelles et les valeurs ajustées sont quasi égales et alignées sur la droite d'équation y=x.

7. Prévoyez le voltage en sortie pour des températures de 100°C, 110°C et 120°C

```
reg<-lm(E~temp,data = temp_sensor)
new_temp<-data.frame(temp=c(100,110,120))
predict(reg,newdata = new_temp,interval = "prediction")
```

```
##        fit      lwr      upr
## 1 5.865190 5.731342 5.999039
## 2 6.382048 6.241325 6.522770
## 3 6.898905 6.750352 7.047457
```