

# Moules

D. Chessel, A.B. Dufour & J.R. Lobry

Introduction élémentaire

## Table des matières


<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Représenter les résultats</b>	<b>2</b>
<b>3</b>	<b>Représenter le modèle</b>	<b>3</b>
<b>4</b>	<b>Conclusion</b>	<b>4</b>
<b>5</b>	<b>Autres présentations</b>	<b>6</b>
	<b>Références</b>	<b>10</b>

## 1 Introduction

Les étudiants sont invités à mesurer l'influence de la température sur la fonction de filtration de l'eau par une espèce de moule *Mytilus edulis*. 10 individus sont placés dans un litre d'eau colorée à une température donnée. On mesure toutes les 5 minutes la densité optique de l'eau (DO) qui caractérise la quantité de colorant non encore piégée par les animaux.

Les résultats sont :

Expérience	1	2	3	4	5	6	7	8	9	10
Température	7.5	15	22	27	34	15	18	22	25	28
Temps	Densité Optique									
5	446	427	413	444	519	466	470	530	509	431
10	447	351	372	390	532	408	435	455	452	375
15	381	292	282	340	515	387	392	386	404	325
20	337	261	223	252	514	314	356	337	354	290
25	315	238	197	285	510	316	316	289	310	253
30	280	200	180	250	508	297	282	233	272	206
35	258	175	129	240	466	222	250	180	225	184
40	244	156	115	180	461	196	222	147	198	162
45	210	130	95	169	466	181	202	123	172	142
50	203	104	85	151	460	162	173	102	146	125
55	170	95	69	137	455	140	153	88	130	109

Importer les données sous  :

```
resu <- read.table("http://pbil.univ-lyon1.fr/R/donnees/moules.txt",
  sep = "\t", header = TRUE)
resu
```

```
      temps exp1 exp2 exp3 exp4 exp5 exp6 exp7 exp8 exp9 exp10
1         5  446  427  413  444  519  466  470  530  509  431
2        10  447  351  372  390  532  408  435  455  452  375
3        15  381  292  282  340  515  387  392  386  404  325
4        20  337  261  223  252  514  314  356  337  354  290
5        25  315  238  197  285  510  316  316  289  310  253
6        30  280  200  180  250  508  297  282  233  272  206
7        35  258  175  129  240  466  222  250  180  225  184
8        40  244  156  115  180  461  196  222  147  198  162
9        45  210  130  95  169  466  181  202  123  172  142
10       50  203  104  85  151  460  162  173  102  146  125
11       55  170  95  69  137  455  140  153  88  130  109
```

```
temp <- c(7.5, 15, 22, 27, 34, 15, 18, 22, 25, 28)
temp
```

```
[1] 7.5 15.0 22.0 27.0 34.0 15.0 18.0 22.0 25.0 28.0
```

## 2 Représenter les résultats

```
resu.vec <- unlist(resu[, -1], use.names = FALSE)
resu.vec
```

```
[1] 446 447 381 337 315 280 258 244 210 203 170 427 351 292 261 238 200 175 156 130
[21] 104 95 413 372 282 223 197 180 129 115 95 85 69 444 390 340 252 285 250 240
[41] 180 169 151 137 519 532 515 514 510 508 466 461 466 460 455 466 408 387 314 316
[61] 297 222 196 181 162 140 470 435 392 356 316 282 250 222 202 173 153 530 455 386
[81] 337 289 233 180 147 123 102 88 509 452 404 354 310 272 225 198 172 146 130 431
[101] 375 325 290 253 206 184 162 142 125 109
```

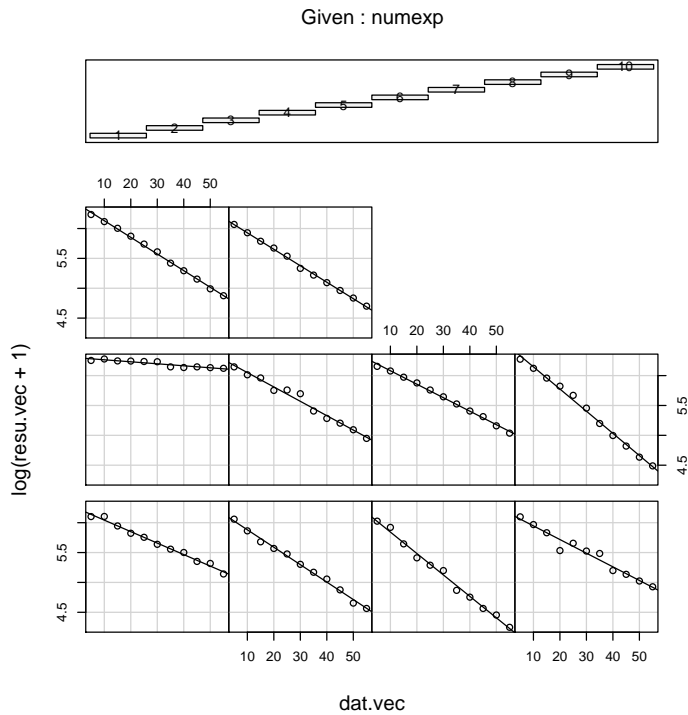
```
dat.vec <- rep(resu$temps, 10)
dat.vec
```

```
[1] 5 10 15 20 25 30 35 40 45 50 55 5 10 15 20 25 30 35 40 45 50 55 5 10 15 20
[27] 25 30 35 40 45 50 55 5 10 15 20 25 30 35 40 45 50 55 5 10 15 20 25 30 35 40
[53] 45 50 55 5 10 15 20 25 30 35 40 45 50 55 5 10 15 20 25 30 35 40 45 50 55 5
[79] 10 15 20 25 30 35 40 45 50 55 5 10 15 20 25 30 35 40 45 50 55 5 10 15 20 25
[105] 30 35 40 45 50 55
```

```
numexp <- factor(rep(1:10, rep(11, 10)))
numexp
```

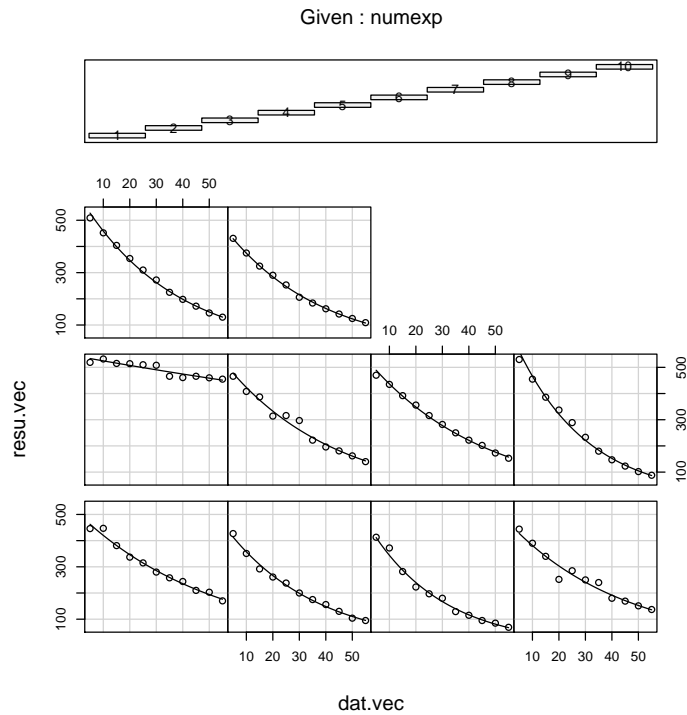
```
[1] 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3
[27] 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5
[53] 5 5 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 8
[79] 8 8 8 8 8 8 8 8 8 8 8 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 10
[105] 10 10 10 10 10 10
Levels: 1 2 3 4 5 6 7 8 9 10
```

```
coplot(log(resu.vec + 1) ~ dat.vec | numexp, panel = function(x,
  y, ...) {
  points(x, y)
  abline(lm(y ~ x))
})
```



### 3 Représenter le modèle

```
f1 <- function(x, y, ...) {
  points(x, y)
  lm0 <- lm(log(y + 1) ~ x)
  x0 <- seq(min(x), max(x), le = 30)
  z0 <- predict(lm0, newdata = list(x = x0))
  lines(x0, exp(z0) - 1)
}
coplot(resu.vec ~ dat.vec | numexp, panel = function(x, y, ...) f1(x,
y, ...))
```



## 4 Conclusion

```
a <- cbind.data.frame(resu.vec, dat.vec)
names(a) <- c("y", "x")
f2 <- function(x) {
  y <- x[, 1]
  x <- x[, 2]
  coefficients(lm(log(y + 1) ~ x))
}
a.lis <- split(a, numexp)
coeff <- matrix(unlist(lapply(a.lis, f2)), ncol = 2, byrow = T)
coeff
```

```
      [,1]      [,2]
[1,] 6.234932 -0.019226082
[2,] 6.173261 -0.029195748
[3,] 6.201924 -0.035883488
[4,] 6.174665 -0.022841169
[5,] 6.298346 -0.003352273
[6,] 6.291274 -0.024053037
[7,] 6.307490 -0.022582296
[8,] 6.517707 -0.037125753
[9,] 6.409480 -0.027866501
[10,] 6.203522 -0.027565887
```

```
coeff <- data.frame(coeff)
names(coeff) <- c("b", "a")
coeff
```

```
      b      a
1 6.234932 -0.019226082
2 6.173261 -0.029195748
3 6.201924 -0.035883488
4 6.174665 -0.022841169
5 6.298346 -0.003352273
```

```
6 6.291274 -0.024053037
7 6.307490 -0.022582296
8 6.517707 -0.037125753
9 6.409480 -0.027866501
10 6.203522 -0.027565887
```

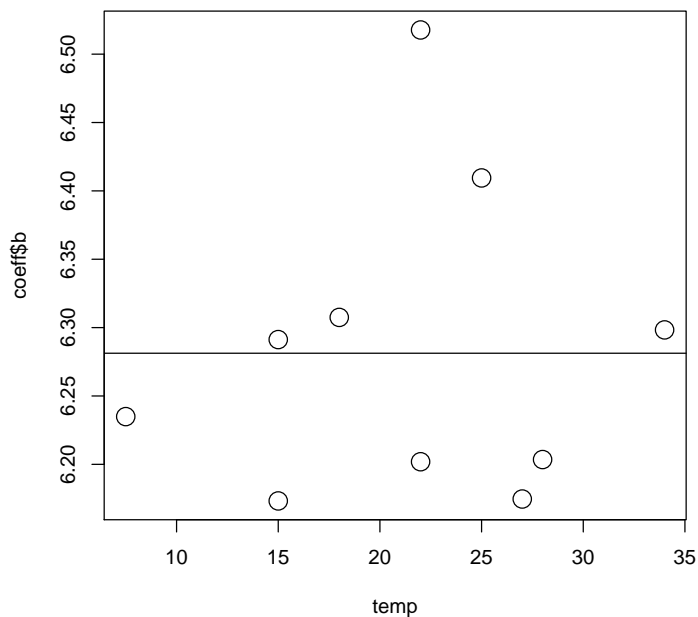
```
anova(lm(coeff$b ~ temp))
```

#### Analysis of Variance Table

Response: coeff\$b

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
temp	1	0.001331	0.001331	0.0971	0.7633
Residuals	8	0.109607	0.013701		

```
plot(temp, coeff$b, cex = 2)
abline(h = mean(coeff$b))
```



```
anova(lm(coeff$a ~ temp + I(temp^2)))
```

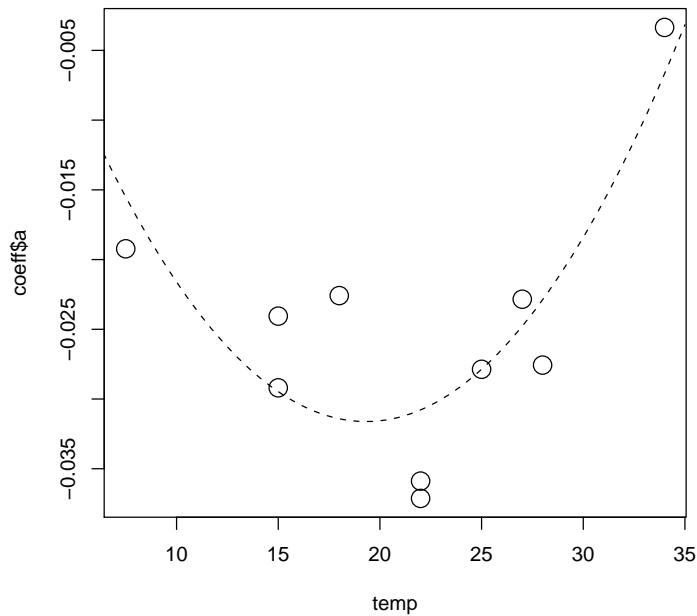
#### Analysis of Variance Table

Response: coeff\$a

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
temp	1	0.00005808	0.00005808	1.812	0.220233
I(temp^2)	1	0.00052877	0.00052877	16.495	0.004801 **
Residuals	7	0.00022439	0.00003206		

---  
Signif. codes: 0

```
plot(temp, coeff$a, cex = 2)
x0 <- seq(5, 35, le = 50)
z0 <- predict(lm(coeff$a ~ temp + I(temp^2)), newdata = list(temp = x0))
lines(x0, z0, lty = 2)
```



Conclure.

## 5 Autres présentations

Présenter toute l'information nécessaire sous forme d'une liste, pour obtenir :

`moules`

```
$time
[1] 5 10 15 20 25 30 35 40 45 50 55

$res
  exp1 exp2 exp3 exp4 exp5 exp6 exp7 exp8 exp9 exp10
1  446  427  413  444  519  466  470  530  509  431
2  447  351  372  390  532  408  435  455  452  375
3  381  292  282  340  515  387  392  386  404  325
4  337  261  223  252  514  314  356  337  354  290
5  315  238  197  285  510  316  316  289  310  253
6  280  200  180  250  508  297  282  233  272  206
7  258  175  129  240  466  222  250  180  225  184
8  244  156  115  180  461  196  222  147  198  162
9  210  130  95  169  466  181  202  123  172  142
10 203  104  85  151  460  162  173  102  146  125
11 170  95  69  137  455  140  153  88  130  109

$temp
[1] 7.5 15.0 22.0 27.0 34.0 15.0 18.0 22.0 25.0 28.0
```

Présenter toute l'information nécessaire sous forme d'un tableau, pour obtenir :

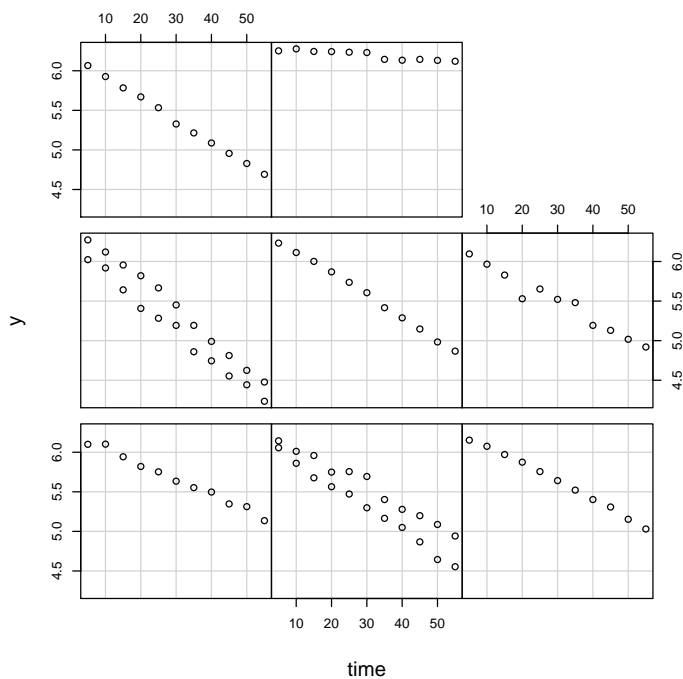
`df.moules`

	reponse	time	temp	ind
exp11	446	5	7.5	1
exp12	447	10	7.5	1
exp13	381	15	7.5	1
exp14	337	20	7.5	1
exp15	315	25	7.5	1
exp16	280	30	7.5	1
exp17	258	35	7.5	1
exp18	244	40	7.5	1
exp19	210	45	7.5	1
exp110	203	50	7.5	1
exp111	170	55	7.5	1
exp21	427	5	15.0	2
exp22	351	10	15.0	2
exp23	292	15	15.0	2
exp24	261	20	15.0	2
exp25	238	25	15.0	2
exp26	200	30	15.0	2
exp27	175	35	15.0	2
exp28	156	40	15.0	2
exp29	130	45	15.0	2
exp210	104	50	15.0	2
exp211	95	55	15.0	2
exp31	413	5	22.0	3
exp32	372	10	22.0	3
exp33	282	15	22.0	3
exp34	223	20	22.0	3
exp35	197	25	22.0	3
exp36	180	30	22.0	3
exp37	129	35	22.0	3
exp38	115	40	22.0	3
exp39	95	45	22.0	3
exp310	85	50	22.0	3
exp311	69	55	22.0	3
exp41	444	5	27.0	4
exp42	390	10	27.0	4
exp43	340	15	27.0	4
exp44	252	20	27.0	4
exp45	285	25	27.0	4
exp46	250	30	27.0	4
exp47	240	35	27.0	4
exp48	180	40	27.0	4
exp49	169	45	27.0	4
exp410	151	50	27.0	4
exp411	137	55	27.0	4
exp51	519	5	34.0	5
exp52	532	10	34.0	5
exp53	515	15	34.0	5
exp54	514	20	34.0	5
exp55	510	25	34.0	5
exp56	508	30	34.0	5
exp57	466	35	34.0	5
exp58	461	40	34.0	5
exp59	466	45	34.0	5
exp510	460	50	34.0	5
exp511	455	55	34.0	5
exp61	466	5	15.0	6
exp62	408	10	15.0	6
exp63	387	15	15.0	6
exp64	314	20	15.0	6
exp65	316	25	15.0	6
exp66	297	30	15.0	6
exp67	222	35	15.0	6
exp68	196	40	15.0	6
exp69	181	45	15.0	6
exp610	162	50	15.0	6
exp611	140	55	15.0	6
exp71	470	5	18.0	7
exp72	435	10	18.0	7
exp73	392	15	18.0	7
exp74	356	20	18.0	7
exp75	316	25	18.0	7
exp76	282	30	18.0	7
exp77	250	35	18.0	7
exp78	222	40	18.0	7
exp79	202	45	18.0	7
exp710	173	50	18.0	7
exp711	153	55	18.0	7
exp81	530	5	22.0	8
exp82	455	10	22.0	8
exp83	386	15	22.0	8
exp84	337	20	22.0	8
exp85	289	25	22.0	8

```
exp86      233  30 22.0  8
exp87      180  35 22.0  8
exp88      147  40 22.0  8
exp89      123  45 22.0  8
exp810     102  50 22.0  8
exp811      88  55 22.0  8
exp91      509   5 25.0  9
exp92      452  10 25.0  9
exp93      404  15 25.0  9
exp94      354  20 25.0  9
exp95      310  25 25.0  9
exp96      272  30 25.0  9
exp97      225  35 25.0  9
exp98      198  40 25.0  9
exp99      172  45 25.0  9
exp910     146  50 25.0  9
exp911     130  55 25.0  9
exp101     431   5 28.0 10
exp102     375  10 28.0 10
exp103     325  15 28.0 10
exp104     290  20 28.0 10
exp105     253  25 28.0 10
exp106     206  30 28.0 10
exp107     184  35 28.0 10
exp108     162  40 28.0 10
exp109     142  45 28.0 10
exp1010    125  50 28.0 10
exp1011    109  55 28.0 10
```

```
y <- log(df.moules$reponse)
time <- df.moules$time
temp <- df.moules$temp
coplot(y ~ time | as.factor(temp), show = F)
```

Given : as.factor(temp)



```
anova(lm(y ~ time + temp + I(temp^2)))
```

Analysis of Variance Table

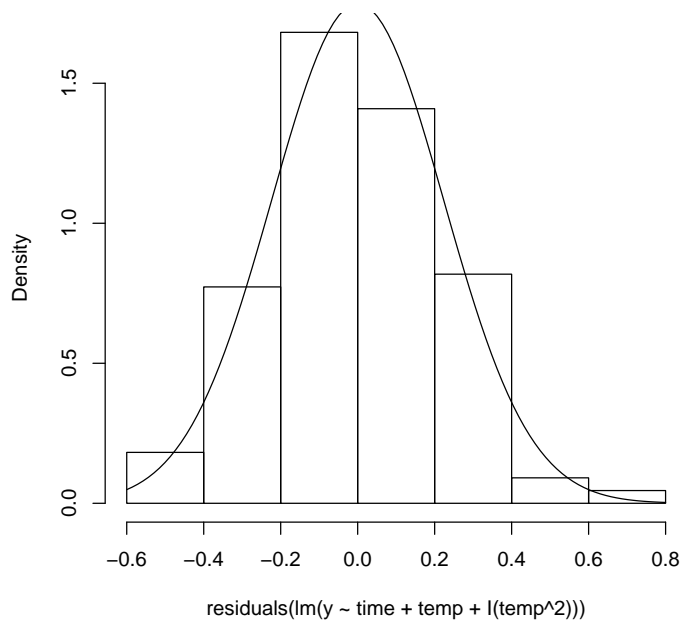
Response: y



```
time      Df Sum Sq Mean Sq F value    Pr(>F)
temp      1  0.7775   0.7775   14.873 0.0001977 ***
I(temp^2) 1  4.2705   4.2705   81.689 8.18e-15 ***
Residuals 106  5.5413   0.0523
---
Signif. codes:  0
```

```
hist(residuals(lm(y ~ time + temp + I(temp^2))), prob = T)
x0 <- seq(-0.6, 0.8, le = 100)
lines(x0, dnorm(x0, 0, sqrt(0.05)))
```

Histogram of residuals(lm(y ~ time + temp + I(temp^2)))



```
coefficients(lm(y ~ time + temp + I(temp^2)))
```

```
(Intercept)      time      temp      I(temp^2)
7.212190458 -0.025087695 -0.118326075  0.003130954
```

## Références