Hw5

Thibaud Bruyelle - Pablo Veyrat

11/13/2020

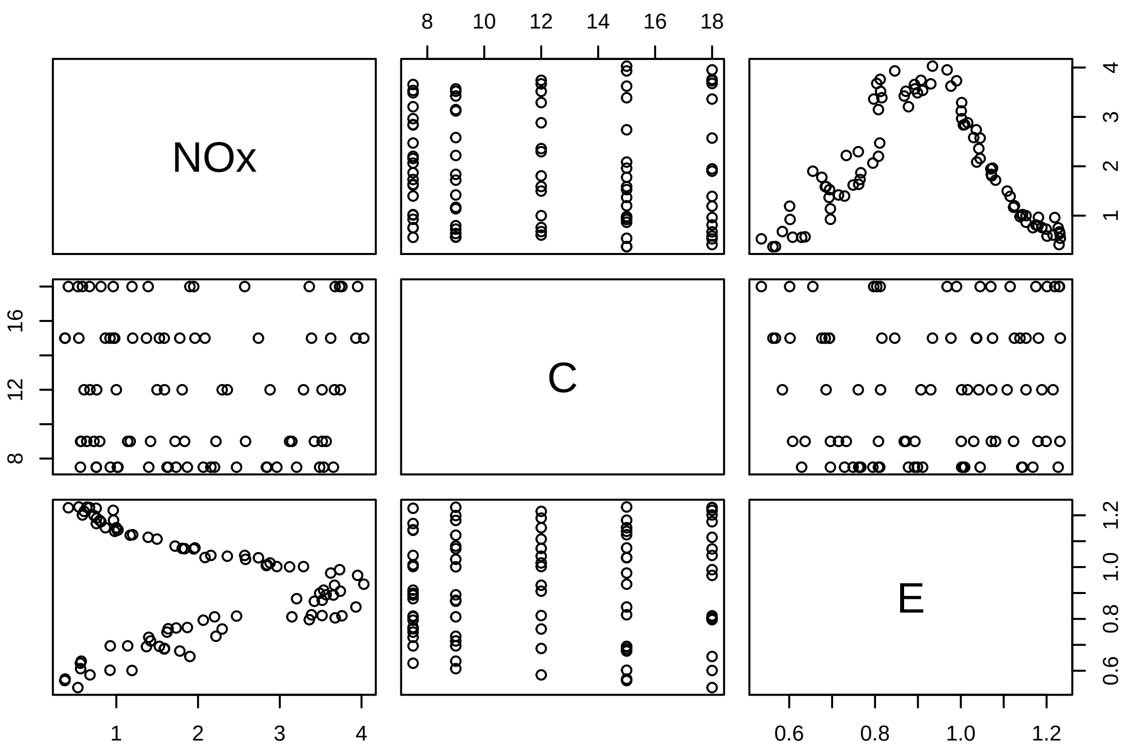
Problem 2: Varying Coefficient Model

Question (a)

Please see handwritten notes.

Question (b)

```
library(SemiPar)
library(tidyverse)
data(ethanol)
pairs(ethanol)
```




```
y = ethanol$NOx
x = ethanol$C
t = ethanol$E
n = length(x)

data <- data.frame("y" = y, "x" = x, "t" = t)
```

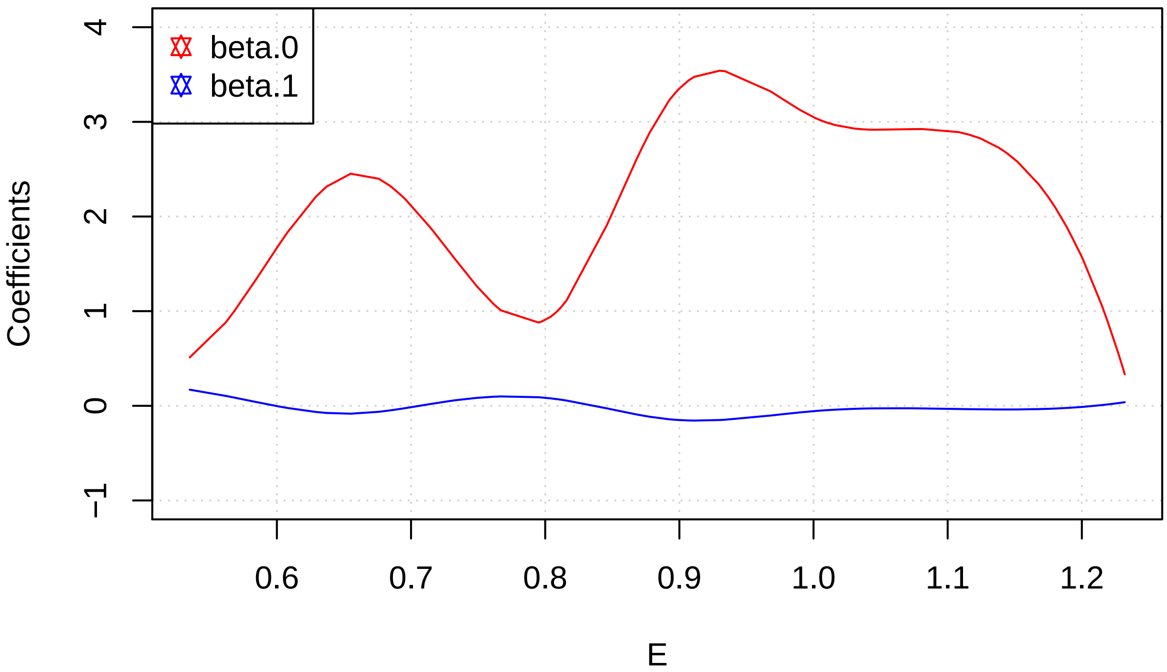
```
# order `df` with respect to `t`
data <- arrange(data, t)
```

Here we took `M=9` because it seems coherent with the data. We could perform a k-folds cross-validation with respect to this hyparemeter in order to select an optimal value of `M*`. Besides, since we are using cubic splines, it means that we have to select 5 knots. After ordering the data with respect to `E`, we selected evenly spread knots.

```
# build cubic splines basis matrix
M = 9
knots = c(data$t[10], data$t[20],
          data$t[30], data$t[40], data$t[50])
H_cubic <- matrix(ncol = M, nrow = n)
for (i in 0:3){
  H_cubic[,i+1] <- (data$t)^i
}
for(i in 1:(M-4)){
  H_cubic[,i+4] <- sapply(data$t,function(r)ifelse(r>=knots[i],(r-knots[i])**3,0))
}

# build predictor matrix X with `2 * M` predictors
X = matrix(nrow = n, ncol = 2*M)
# covariates from intercept term beta_0
for (i in 1:M){
  X[,i] = H_cubic[,i]
}
# covariates from coefficient of `x` beta_1
for (i in 1:M){
  X[,i+M] = data$x * H_cubic[,i]
}

# FIT THE MODEL
model.M <- lm(y ~ X -1)
```

Question (c)

```
# vector beta.0
theta.0 <- model.M$coefficients[1:9]
beta.0 <- H_cubic %*% theta.0
# vector beta.1
theta.1 <- model.M$coefficients[10:18]
beta.1 <- H_cubic %*% theta.1

# PLots
plot(data$t, beta.0, col = 'red', panel.first=grid(),
     type = 'l', ylim = c(-1, 4), xlab = 'E', ylab = 'Coefficients')
lines(data$t, beta.1, col = 'blue', type = 'l')
legend('topleft', legend = c('beta.0', 'beta.1'), col = c('red', 'blue'), pch = 11)
```



Question (d)

```
# build restricted model
X_restricted = X[,1:11]
model_restricted <- lm(y ~X_restricted - 1)
RSS_restricted <- sum((model_restricted$residuals)^2)
RSS_ur <- sum((model.M$residuals)^2)
F.stat <- ((RSS_restricted - RSS_ur) * (n - 2*M)) / (RSS_ur * M-2)
print(F.stat)
```

```
## [1] 0.3732751
```

```
qf(0.99, df1 = M-2, df2 = n-2*M )
```

```
## [1] 2.906032
```

Conclusion: From the statistic test performed above, let us notice that we fail to reject $\mathcal{H}_0$ at all confidence levels. Consequently, it is *statistically* significant to consider that the slope function is linear in `E`.

Question (e)

```
# apply the same method than above
X_restricted2 = X[,1:10]
model_restricted2 <- lm(y ~X_restricted2 - 1)
RSS_restricted2 <- sum((model_restricted2$residuals)^2)
RSS_ur <- sum((model.M$residuals)^2)
F.stat <- ((RSS_restricted2 - RSS_ur) * (n - 2*M)) / (RSS_ur * M-1)
print(F.stat)
```

```
## [1] 0.3739223
```

```
qf(0.99, df1 = M-2, df2 = n-2*M )
```

```
## [1] 2.906032
```

Conclusion: Again, we fail to reject the null hypothesis $\mathcal{H}_0$ so it is *statisically* significant to say that this slope function is constant in `E`.

STATS 305A : HW5

Problem 2

(a) We want to fit the model:

$$f(x, t) = \beta_0(t) + \beta_1(t)x + \varepsilon_t$$

where t is a value of the variable T, and x a value of the predictor X. In order to estimate (β_0, β_1) with respect to T, we will use a polynomial splines with a truncated power basis.

By choosing $n-4$ knots evenly spread over the values of T we would have the basis:

$$1, x, x^2, x^3, (x-h_1)^3_+, \ldots, (x-h_{n-4})^3_+$$

and also we could define the basis matrix of splines in T: $\quad H_T \;(m \times n \text{ matrix})$

s.t:

$$\begin{cases} \underline{\beta_0} = H_T \underline{\theta_0} \\ \underline{\beta_1} = H_T \underline{\theta_1} \end{cases}$$

In other words, our model

writes:

$$\mathbb{E}(Y \mid T, X) = \sum_{m=1}^{M} \theta_{0m} h_m(t) + \sum_{j=1}^{p} X_j \sum_{m=1}^{M} \theta_{jm} h_m(t)$$

So, when $p = 1$ like in our case, there are exactly:

$$\boxed{2 \times M},$$

parameters to estimate which are:

$$\theta_{01}, \dots, \theta_{0M}, \theta_{11}, \dots, \theta_{1M}.$$

For $i \in [\![1, n]\!]$, as the model

writes:

$$y_i = \sum_{m=1}^{M} \theta_{0m} h_m(t_i) + \sum_{m=1}^{M} \theta_{1m} h_m(t_i) x_i$$

we can see it as a regression model with covariates:

$$h_1(T), \dots, h_M(T), h_1(T)X, \dots, h_M(T)X$$

So by writing:

$$\tilde{X} = \left(h_1(T), \dots, h_M(T), X h_1(T), \dots, X h_M(T) \right)$$

we can fit the linear regression :

$$y = \hat{\tilde{X}} \alpha + \varepsilon$$

where α is a vector of size $2n$.

(d)

- The slope function writes:

$$\hat{\beta}_1(t) = H_{T=t} \times \hat{\theta}_1 , \forall t$$

where

$$\hat{\theta}_1 = (\hat{\theta}_{11}, \ldots, \hat{\theta}_{n1})$$

defined as `theta.1` in

the code part.
So we have:

$$\hat{\beta}_1(t) = \sum_{m=1}^{M_1} \hat{\theta}_{m1} h_m(t).$$

Besides, for $m > 2$, $h_m(t)$ is polynomial in t with degree at least 2. Consequently, in order to test whether $\hat{\beta}_1(\cdot)$ is linear in E, we need to compute a

F-test

with null hypothesis:

$$\mathcal{H}_0: \beta_{31} = \beta_{41} = \dots = \beta_{M1} = 0$$

The test statistic can be defined:

$$\boxed{F_{stat} = \frac{(RSS_{restricted} - RSS_{ur})/n-2}{RSS_{ur}/(n-2M)}}$$

where $RSS_{restricted}$ are the residuals sum of squares for the restricted model with only $n+2$ predictors.

• The functional form of this model is:

$$\mathbb{E}(Y|X,T) = \beta_0(T) + \overbrace{h_1(t)}^{=1}\theta_{11} x_1 + \underbrace{h_2(t)}_{=t}\theta_{21} x_1$$

$$\boxed{= \beta_0(T) + (\theta_{11} \cdot 1 + \theta_{21} \cdot t) x_1}$$

The results are right after the code part.

(c) Let us compute the same test than above with a different restricted model that writes:

$$\beta_1(T)\, x_1$$

$$\boxed{\mathbb{E}(Y \mid X, T) = \beta_0(T) + \theta_{11}\, x_1}$$

So the F-stat is s.t:

$$\underline{F\text{-stat} \sim \text{Fisher}(n-1, n-2n)}$$

The results are given after the code section.