Hw5

Thibaud Bruyelle - Pablo Veyrat

11/13/2020

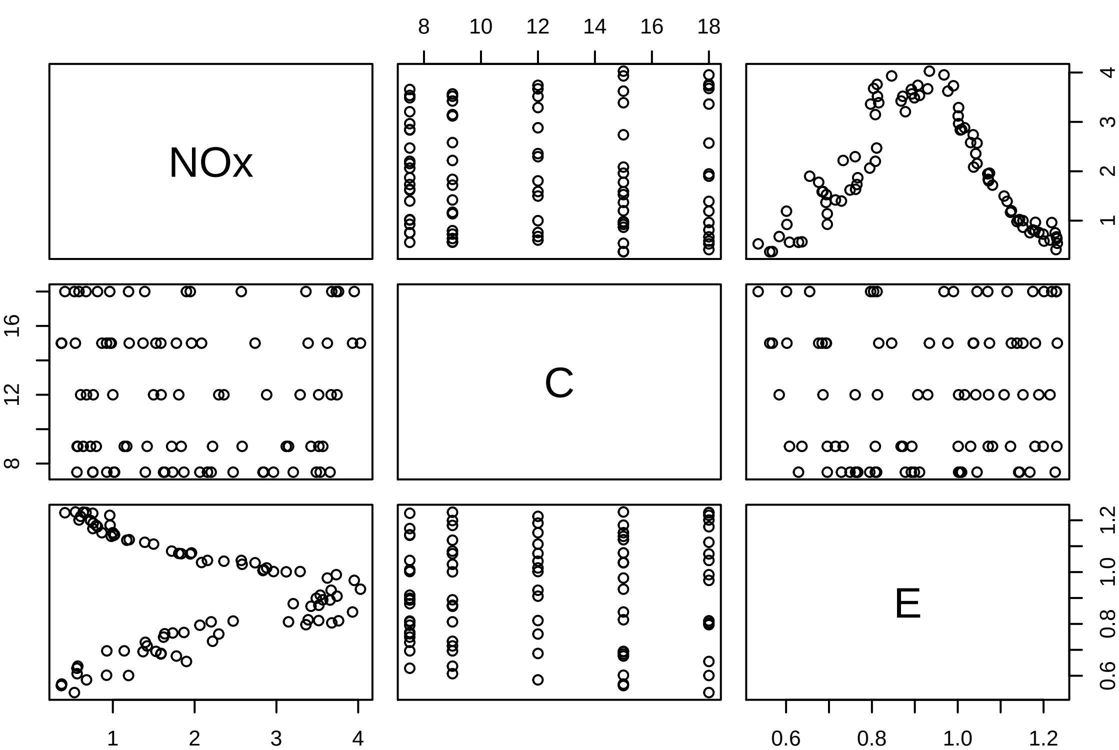
Problem 2: Varying Coefficient Model

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)
```

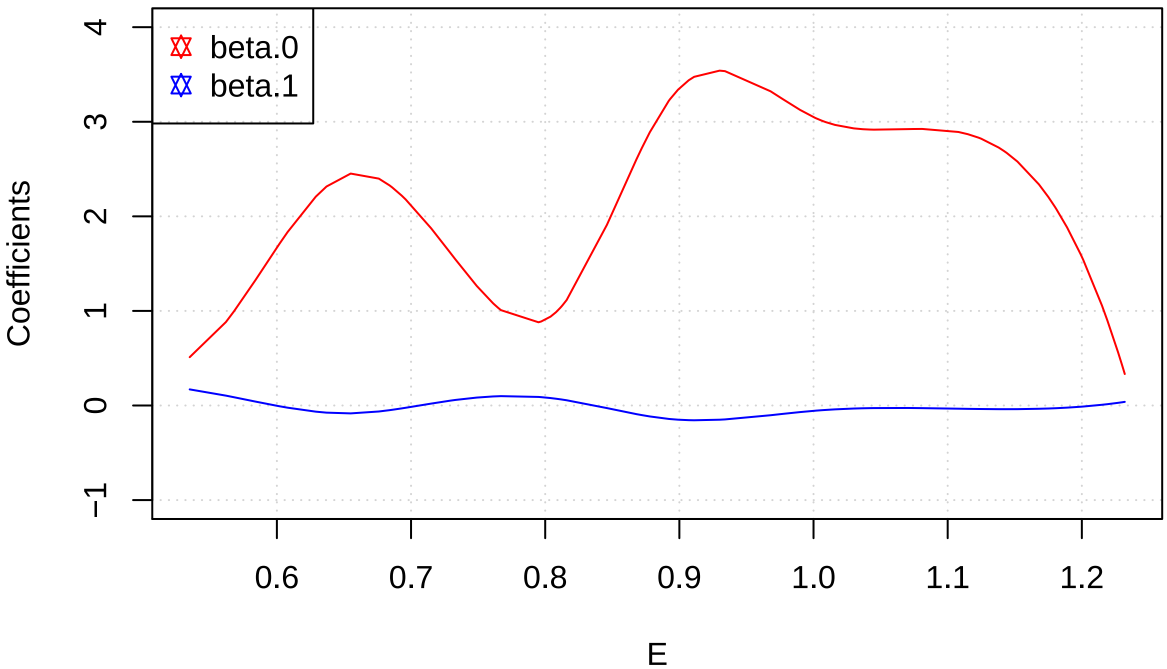
```r
# order `df` with respect to `t`
data <- arrange(data, t)
```

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.

```r
# 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)

```r
# 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)

```r
# 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
```

```r
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)

```r
# 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
```

```r
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`.