Introduction to Machine Learning

Lab 3

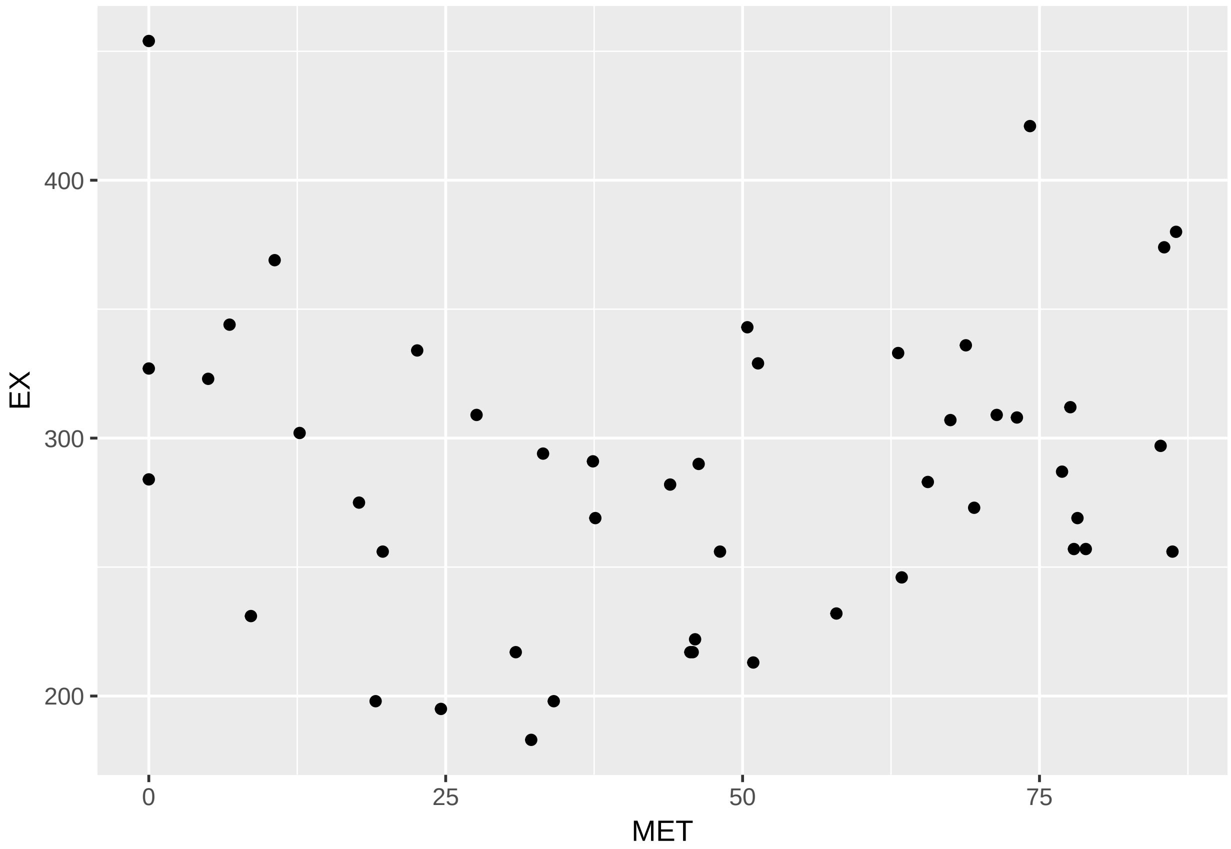
Rasmus Holm

2016-11-23

Contents

Assignment 1

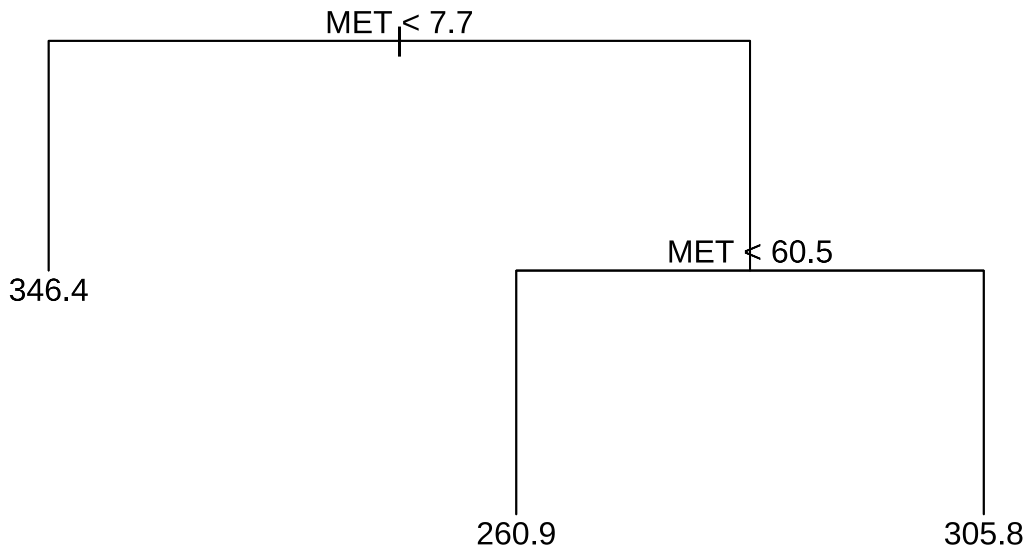
1

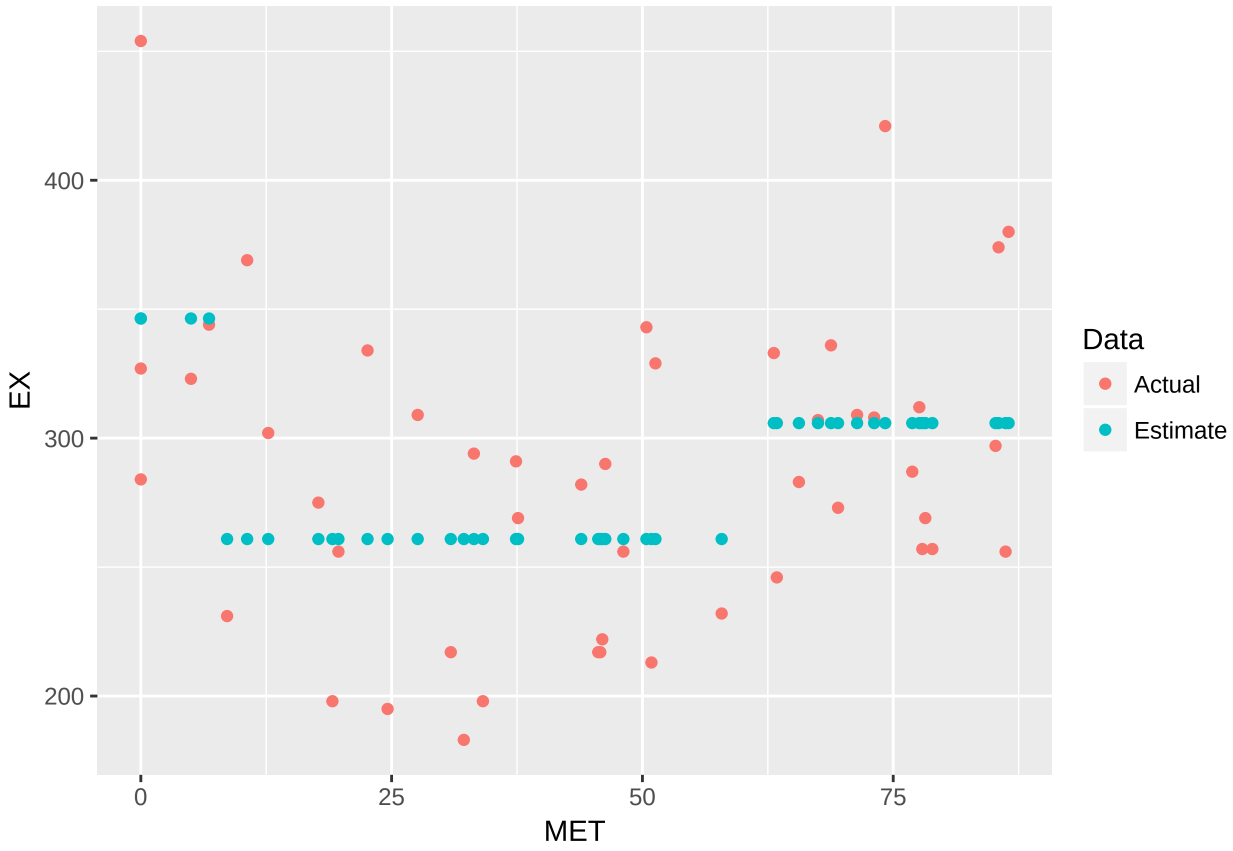


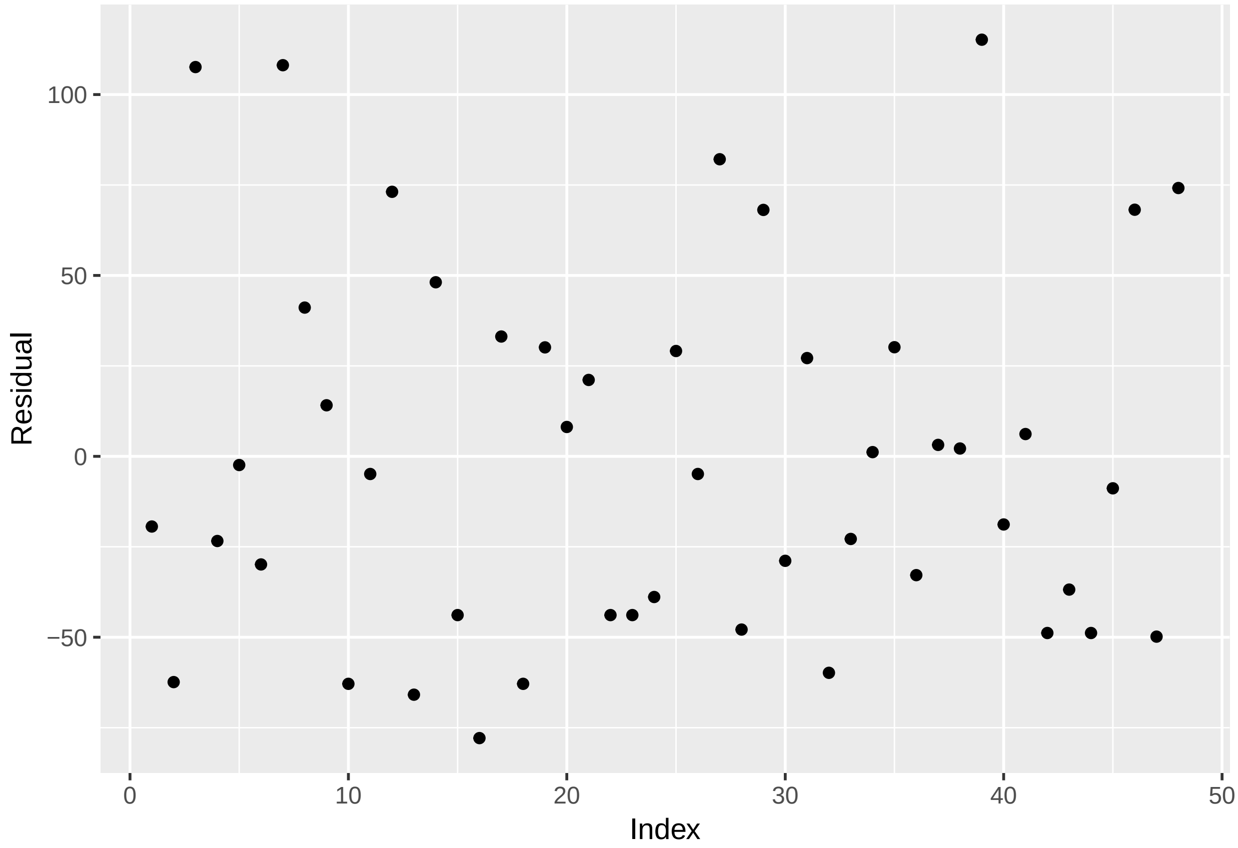
The data looks like it could be modelled by a cubic spline function.

2

```
#> $mincut
#> [1] 4
#>
#> $minsize
#> [1] 8
#>
#> $mindev
#> [1] 0.01
#>
#> $nmax
#> [1] 28
#>
#> $nobs
#> [1] 48
```


MET < 7.7
MET < 60.5
346.4
260.9
305.8


EX
400
300
200
0
25
50
75
MET
Data
Actual
Estimate



3

4

5

Assignment 2

1

2

3

4

Appendix

Code for Assignment 1

```r
library(ggplot2)
library(tree)
library(reshape)

data <- read.csv2("../data/State.csv", header=TRUE, sep=";")
data <- data[order(data$MET),]
ggplot(data) +
    geom_point(aes(x=MET, y=EX))

tree.control(nobs=nrow(data), minsize=8)
treefit <- tree(EX ~ MET, data=data)

treefit.cv <- cv.tree(treefit, FUN=prune.tree, K=10)
optimal_leaf_count<- treefit.cv$size[which.min(treefit.cv$dev)]

optimal_tree <- prune.tree(treefit, best=optimal_leaf_count)
plot(optimal_tree)
text(optimal_tree, pretty=0)
predicted <- predict(optimal_tree, data)
plot_data <- data.frame(MET=data$MET, Actual=data$EX, Estimate=predicted)
plot_data <- melt(plot_data, id.vars="MET")
names(plot_data) <- c("MET", "Data", "EX")

ggplot(plot_data) +
    geom_point(aes(x=MET, y=EX, color=Data))
residuals <- resid(optimal_tree)

plot_data <- data.frame(x=1:length(residuals), y=residuals)

ggplot(plot_data) +
    xlab("Index") +
    ylab("Residual") +
    geom_point(aes(x=x, y=y))
```

Code for Assignment 2