```
#### Lab 2 - Assignment 3 - Uncertainty estimation
### Libraries
library(tree)
library(boot)

### Setup
data.state = read.csv2("State.csv")
data.state = data.state[order(data.state$MET),] # sorted data
n = dim(data.state)[1]                          # number of observations
tree.control = tree.control(n, minsize=8)

### Functions

# Returns a fitted tree of given data
get_tree = function(..data){
  return(tree(formula=EX~MET, data=..data, control=tree.control))
}

# Returns a fitted pruned tree of given data and number of leaves
get_pruned_tree = function(..data, ..leaves){
  tree = get_tree(..data)
  return(prune.tree(tree, best=..leaves))
}

# Returns predictions for non-parametic bootstrapping
statistic1 = function(..data, ..indices){
  data = ..data[..indices,]
  pruned.tree = get_pruned_tree(data, 3)
  yfit = predict(pruned.tree, data.state)
  return(yfit)
}

# Returns predictions for parametic bootstrapping
statistic2 = function(data){
  pruned.tree = get_pruned_tree(data, 3)
  yfit = predict(pruned.tree, data.state)
  return(yfit)
}

# Returns predictions for parametic bootstrapping, with a distribution
added to the predicionts
statistic3 = function(data){
  pruned.tree = get_pruned_tree(data, 3)
  yfit = predict(pruned.tree, data.state)
  yfit = rnorm(n, yfit, sd(residuals(tree.selected)))
  return(yfit)
}

# Returns data with EX values replaced with their prediction based on
given MET
# The predictio is "moved" around under the assumption that Y~N(μi,
sigma^2)
rng = function(..data, model){
  data = data.frame(EX=..data$EX, MET=..data$MET)
```

```
  yfit = predict(model, data)
  data$EX = rnorm(n, yfit, sd(residuals(model)))
  return(data)
}

# Plots band
plot_band = function(e, yfit, ..main){
  plot(data.state$MET, data.state$EX, ylim=c(150, 475), main=..main,
xlab="MET", ylab="EX")
  lines(data.state$MET, yfit)
  lines(data.state$MET, e$point[2,], col="blue")
  lines(data.state$MET, e$point[1,], col="blue")
}

### Implementation
## Task 1 - Plot EX vs MET
plot(data.state$MET, data.state$EX, col="forestgreen",
     main="EX vs MET", xlab="MET", ylab="EX")
# Comment: A quadratic linear regression model would fit well

## Task 2 -
# "in which the number of the leaves is selected by cross-validation,
use the entire
# data set and set minimum number of observations in a leaf equal to 8
(setting minsize in tree.control)."
tree = get_tree(data.state)
set.seed(12345)
cv.tree = cv.tree(tree)
plot(cv.tree$size, cv.tree$dev, col="darkblue", type="b",
     main="CV resuls: deviance based on # of leaves", xlab="# of
leaves", ylab="deviance")
# Comment: minimum deviance at size=3 leaves, selecting 3 leaves

# Fit regression tree
tree.selected = get_pruned_tree(data.state, 3)
# Plot selected tree
plot(tree.selected)
text(tree.selected, pretty=0)

# Make predictions
yfit.tree.selected = predict(tree.selected, data.state)
plot(data.state$MET, data.state$EX, col="blue")
points(data.state$MET, yfit.tree.selected, col="red")
residuals = residuals(tree.selected)
hist(residuals, main="Residuals", xlab="Error value",
col="forestgreen", xlim=c(-125, 125))

# Task 3 - Non-parametic: Confidence band
set.seed(12345)
boot1 = boot(data.state, statistic=statistic1, R=1000)
e1 = envelope(boot1)
plot_band(e1, yfit.tree.selected, "Confidence band for Non-Parametic
Bootstrap")

# Task 4 - Parametic: Confidence band and Prediction band
```

```
set.seed(12345)
# Confidence band
boot2 = boot(data.state, statistic=statistic2, R=1000,
mle=tree.selected, ran.gen=rng, sim="parametric")
e2 = envelope(boot2)
plot_band(e2, yfit.tree.selected, "Confidence band for Parametic
Bootstrap")

# Prediction band
set.seed(12345)
boot3 = boot(data.state, statistic=statistic3, R=1000,
mle=tree.selected, ran.gen=rng, sim="parametric")
e3 = envelope(boot3)
plot_band(e3, yfit.tree.selected, "Prediction band for Parametic
Bootstrap")
```