Gas Turbine CO and NOx Emission Analysis

Aayushi Gupta, Kyle Kaminski, Rosa Lin, Ruben Martinez

Introduction

The combined cycle power plant, also known as combined cycle gas turbine plant, is an assembly of heat engines that combine to generate electricity (Tüfekci). A combined-cycle power plant (CCPP) is made up of gas turbines, steam turbines, and heat recovery steam generators. The electricity is generated and combined in one cycle by gas and steam turbines and then transferred from one turbine to another.

We are interested in identifying the process variables that impact carbon monoxide emissions. By determining the process variables that impact carbon monoxide emissions we will be able to find opportunities to reduce carbon monoxide emissions.

Gas Turbine CO and NOx Emission Data Set

The data comes from a gas turbine located in Turkey that studies the flue gas emissions of specifically carbon monoxide (CO) and nitrogen oxide (NOx) gases. The data set provides hourly statistics of 11 sensors. Data points were collected from a gas turbine from Jan 01 2011 to Dec 13 2015.

Description

The data file `gt_2015.csv` has 7384 observations and 11 variables from the UCI Gas Turbine CO and NOx Emission Data Set. We are going to explore and analyze the following variables:

- `AT` - Ambient Temperature
- `AP` - Ambient Pressure
- `AH` - Ambient Humidity
- `AFDP` - Air filter difference pressure
- `GTEP` - Gas turbine exhaust pressure
- `TIT` - Turbine inlet temperature
- `TAT` - Turbine after temperature
- `TEY` - Turbine energy yield
- `CDP` - Compressor discharge pressure

Here's a quick peek at the data set:

AT	AP	AH	AFDP	GTEP	TIT	TAT	TEY	CDP	CO	NOX
1.95320	1020.1	84.985	2.5304	20.116	1048.7	544.92	116.27	10.799	7.4491	113.250
1.21910	1020.1	87.523	2.3937	18.584	1045.5	548.50	109.18	10.347	6.4684	112.020
0.94915	1022.2	78.335	2.7789	22.264	1068.8	549.95	125.88	11.256	3.6335	88.147
1.00750	1021.7	76.942	2.8170	23.358	1075.2	549.63	132.21	11.702	3.1972	87.078
1.28580	1021.6	76.732	2.8377	23.483	1076.2	549.68	133.58	11.737	2.3833	82.515
1.83190	1021.7	76.411	2.8410	23.495	1076.4	549.92	133.58	11.829	2.0812	81.193

Here's some descriptive statistics of the data set:

```
##        AT               AP               AH             AFDP
##  Min.   :-6.235   Min.   : 989.4   Min.   :24.09   Min.   :2.369
##  1st Qu.:11.073   1st Qu.:1009.7   1st Qu.:59.45   1st Qu.:3.117
##  Median :17.456   Median :1014.0   Median :70.95   Median :3.538
##  Mean   :17.225   Mean   :1014.5   Mean   :68.65   Mean   :3.599
##  3rd Qu.:23.685   3rd Qu.:1018.3   3rd Qu.:79.65   3rd Qu.:4.195
##  Max.   :37.103   Max.   :1036.6   Max.   :96.67   Max.   :5.239
##       GTEP            TIT             TAT             TEY
##  Min.   :17.70   Min.   :1016   Min.   :516.0   Min.   :100.0
##  1st Qu.:23.15   1st Qu.:1070   1st Qu.:544.7   1st Qu.:126.3
##  Median :25.33   Median :1080   Median :549.7   Median :131.6
##  Mean   :26.13   Mean   :1079   Mean   :546.6   Mean   :134.0
##  3rd Qu.:30.02   3rd Qu.:1100   3rd Qu.:550.0   3rd Qu.:147.2
##  Max.   :40.72   Max.   :1100   Max.   :550.6   Max.   :179.5
##       CDP              CO               NOX
##  Min.   : 9.871   Min.   : 0.2128   Min.   : 25.91
##  1st Qu.:11.466   1st Qu.: 1.8082   1st Qu.: 52.40
##  Median :11.933   Median : 2.5334   Median : 56.84
##  Mean   :12.097   Mean   : 3.1300   Mean   : 59.89
##  3rd Qu.:13.148   3rd Qu.: 3.7026   3rd Qu.: 65.09
##  Max.   :15.159   Max.   :41.0970   Max.   :119.68
```

Goals

The goal for this project is to utilize this data set for the purpose of studying flue gas emissions, specifically carbon monoxide(CO) and nitrogen oxides (NOx). Our focus will be to find statistically significant relationships between the ambient and turbine variables and the emissions variables. We will limit the size of our model to more clearly demonstrate these relationships. Ultimately we will suggest which variables make the biggest impact on emission levels in order to decrease emissions overall.

Exploratory Data Analysis

Relationships between feature variables

Figure 1: Scatterplot Matrices to decide which feature variables have a linear relationship

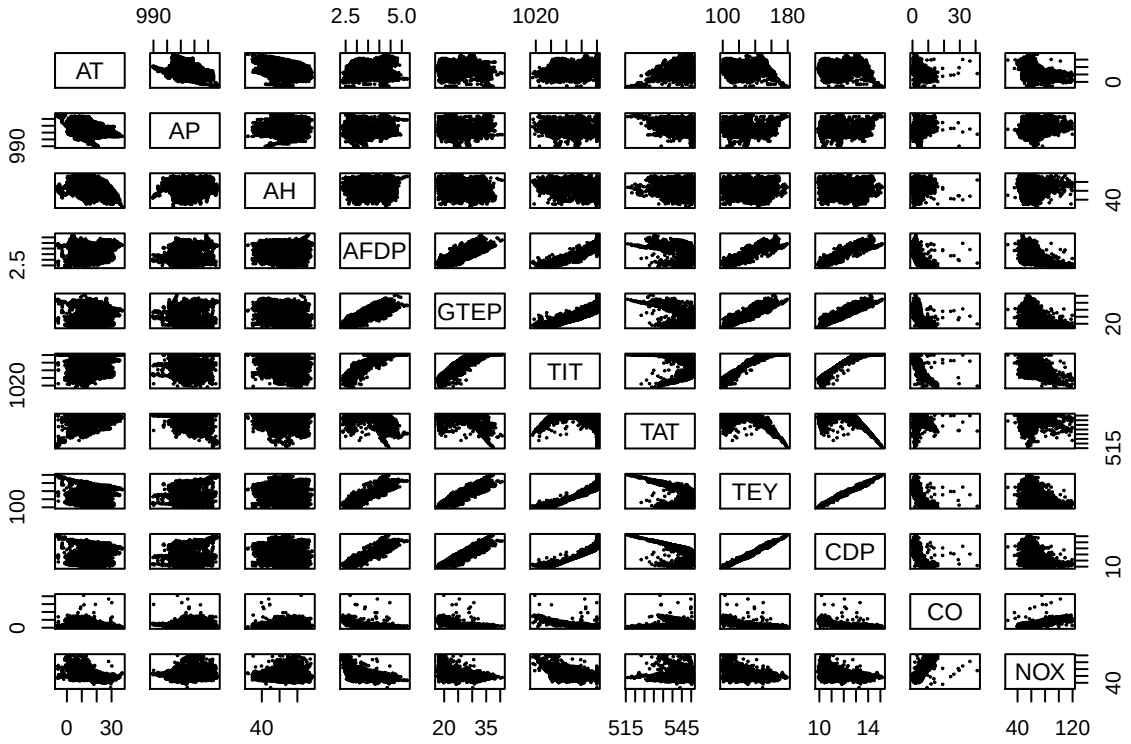


Figure 2:

Table 2: Pairwise Correlation Between Variables

	AT	AP	AH	AFDP	GTEP	TIT	TAT	TEY	CDP	CO	NOX
AT	1.00	-0.49	-0.47	0.47	0.19	0.33	0.21	0.11	0.20	-0.39	-0.59
AP	-0.49	1.00	0.08	-0.09	-0.04	-0.08	-0.29	0.05	0.03	0.20	0.21
AH	-0.47	0.08	1.00	-0.25	-0.30	-0.26	0.03	-0.18	-0.22	0.16	0.07
AFDP	0.47	-0.09	-0.25	1.00	0.84	0.92	-0.52	0.88	0.92	-0.64	-0.58
GTEP	0.19	-0.04	-0.30	0.84	1.00	0.89	-0.62	0.93	0.94	-0.56	-0.37
TIT	0.33	-0.08	-0.26	0.92	0.89	1.00	-0.40	0.95	0.95	-0.74	-0.52
TAT	0.21	-0.29	0.03	-0.52	-0.62	-0.40	1.00	-0.63	-0.66	0.03	0.05
TEY	0.11	0.05	-0.18	0.88	0.93	0.95	-0.63	1.00	0.99	-0.62	-0.40
CDP	0.20	0.03	-0.22	0.92	0.94	0.95	-0.66	0.99	1.00	-0.61	-0.44
CO	-0.39	0.20	0.16	-0.64	-0.56	-0.74	0.03	-0.62	-0.61	1.00	0.68
NOX	-0.59	0.21	0.07	-0.58	-0.37	-0.52	0.05	-0.40	-0.44	0.68	1.00

Remove variables that are highly correlated.

```
##       AT       AP       AH     AFDP     GTEP      TAT
## 3.866424 1.600597 1.718769 7.412520 5.909197 2.301015
```

Exploratory analysis shows possible linear relationships between the response variable CO and the feature variables CDP, TEY, TIT, GTEP and AFDP. Collinearity between some of the feature variables (TIT, CDP, and TEY) could cause some problems in our analysis and will likely lead to the removal of the redundant variables.

Methods

Linear Regression

We will create a multiple linear regression model using all feature variables mentioned in the description of Section 1. The implementation and parameters of this model can be obtained by the following equation where we will find estimates for the parameters β using:

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

Key assumptions are stated as:

- **L**inearity: can be written as a linear combination of the predictors.
- **I**ndependence: the errors are independent of each other (not highly correlated).
- **N**ormality: the distribution of the errors follow a normal distribution.
- **E**qual Variance: the error variance is the same.[1]

We will then use model selection using backward BIC to tune our model and remove any insignificant predictor variables. This selection prefers smaller models which aligns with our goal of limiting the size of our final model.

```
full_model = lm(CO ~ ., data = gt_2015)
linear_model = lm(CO ~ .-NOX - TIT - CDP - TEY, data = gt_2015)
summary(linear_model)
```

```
##
## Call:
## lm(formula = CO ~ . - NOX - TIT - CDP - TEY, data = gt_2015)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.839 -0.673 -0.132  0.481 34.242
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 145.099775   4.408695  32.912  < 2e-16 ***
## AT            0.028276   0.004060   6.965 3.57e-12 ***
## AP            0.001918   0.003067   0.625    0.532
## AH           -0.009753   0.001618  -6.026 1.76e-09 ***
## AFDP         -2.531044   0.074576 -33.939  < 2e-16 ***
## GTEP         -0.186308   0.009082 -20.513  < 2e-16 ***
## TAT          -0.237369   0.004619 -51.387  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.436 on 7377 degrees of freedom
## Multiple R-squared:  0.5874, Adjusted R-squared:  0.587
## F-statistic:  1750 on 6 and 7377 DF,  p-value: < 2.2e-16
```

[1] Dalpiaz David, Applied Statistics in R, https://daviddalpiaz.github.io/appliedstats/model-diagnostics.html

```r
#picking a new variable to test
AT_model = lm(CO ~ AT, data = gt_2015)
AP_model = lm(CO ~ AP, data = gt_2015)
AH_model = lm(CO ~ AH, data = gt_2015)
AFDP_model = lm(CO ~ AFDP, data = gt_2015)
GTEP_model = lm(CO ~ GTEP, data = gt_2015)
TAT_model = lm(CO ~ TAT, data = gt_2015)
BIC(AT_model)
```

```
## [1] 31634.69
```

```r
BIC(AP_model)
```

```
## [1] 32553.05
```

```r
BIC(AH_model)
```

```
## [1] 32668.32
```

```r
BIC(AFDP_model) #second best
```

```
## [1] 28953.71
```

```r
BIC(GTEP_model)
```

```
## [1] 30112.68
```

```r
BIC(TAT_model)
```

```
## [1] 32852.49
```

```r
BIC(linear_model) #this is the best model
```

```
## [1] 26365.45
```

```r
library(MASS)

n = length(resid(linear_model))
BIC_model = step(linear_model, direction = "backward", k = log(n))
```

```
## Start:  AIC=5401.66
## CO ~ (AT + AP + AH + AFDP + GTEP + TIT + TAT + TEY + CDP + NOX) -
##     NOX - TIT - CDP - TEY
##
##        Df Sum of Sq   RSS    AIC
## - AP    1       0.8 15218 5393.1
## <none>              15217 5401.7
## - AH    1      74.9 15292 5429.0
```

```
## - AT    1     100.1 15317 5441.1
## - GTEP  1     868.0 16085 5802.4
## - AFDP  1    2376.0 17593 6464.1
## - TAT   1    5447.0 20664 7652.1
##
## Step:  AIC=5393.14
## CO ~ AT + AH + AFDP + GTEP + TAT
##
##        Df Sum of Sq   RSS    AIC
## <none>              15218 5393.1
## - AH    1      86.8 15304 5426.2
## - AT    1     120.5 15338 5442.5
## - GTEP  1     991.0 16209 5850.1
## - AFDP  1    2564.3 17782 6534.1
## - TAT   1    5608.5 20826 7701.0
```

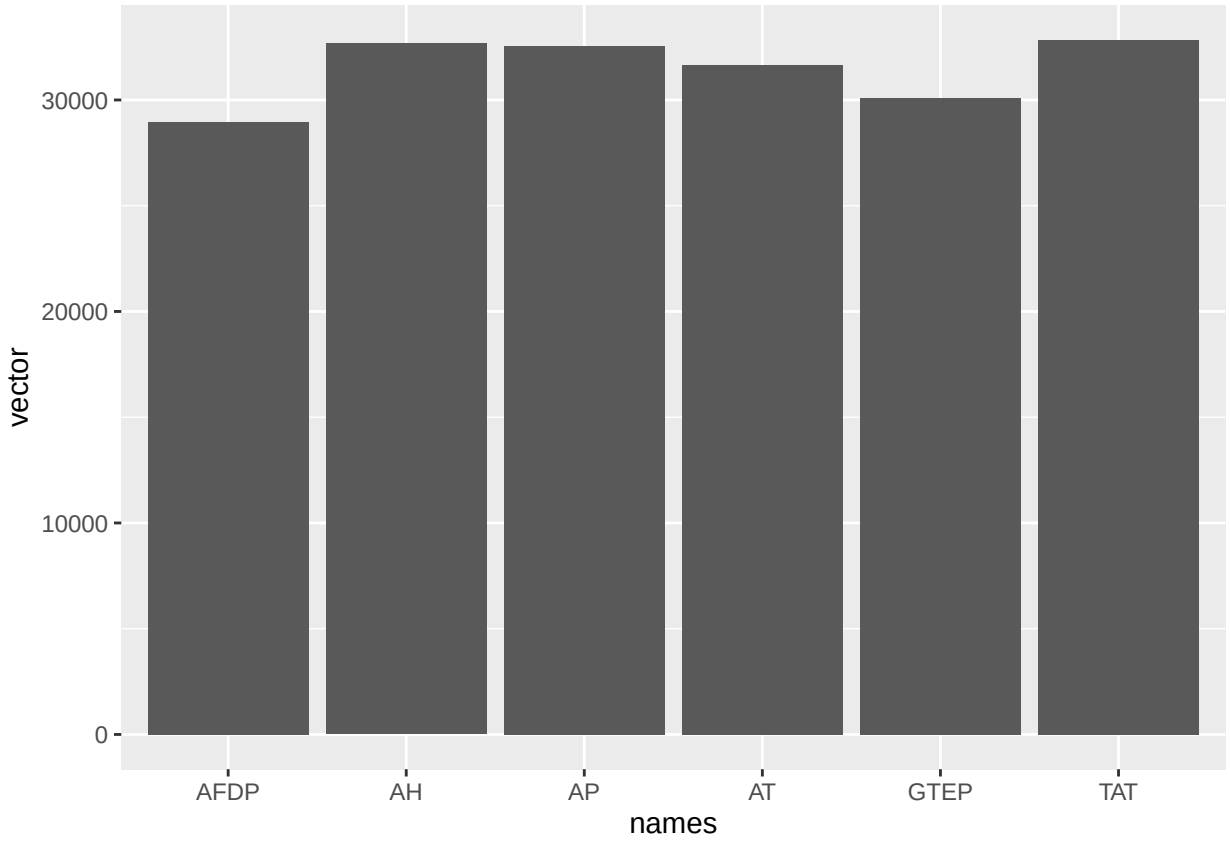
```
coef(BIC_model)
```

```
##  (Intercept)           AT           AH         AFDP         GTEP          TAT
## 147.33755305   0.02702808  -0.01004686  -2.51758434  -0.18816259  -0.23782668
```

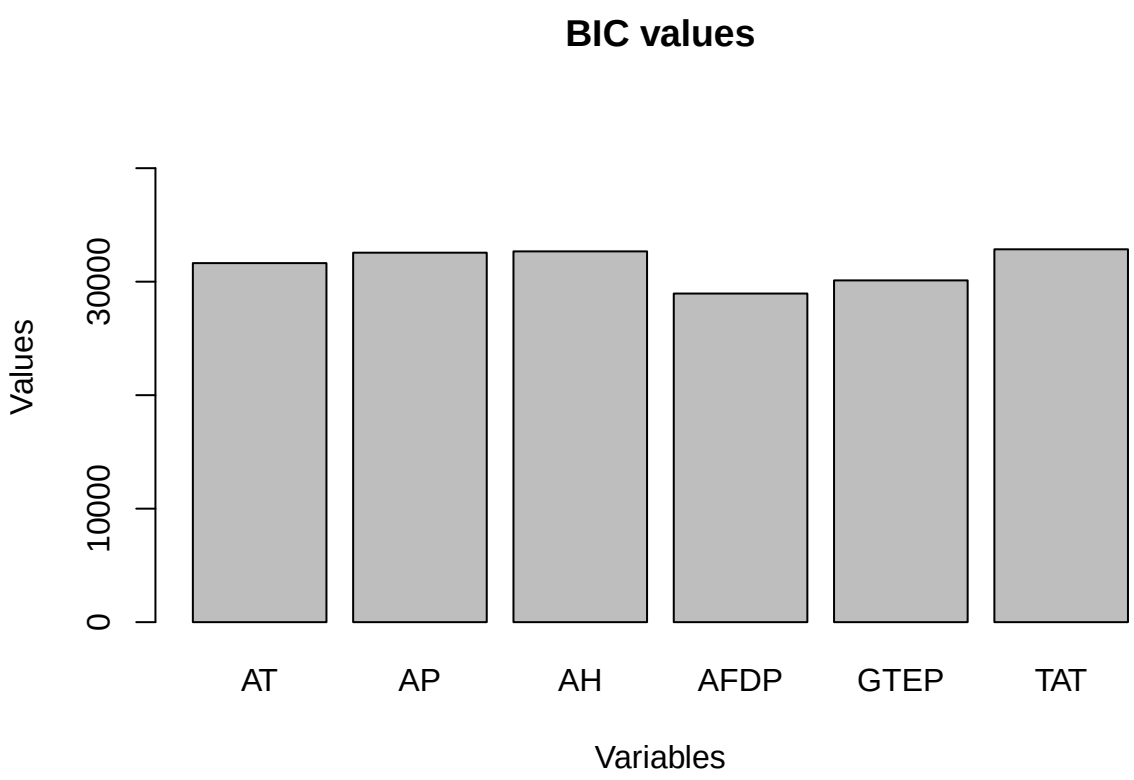
```
stepAIC(linear_model, direction = "backward")
```

```
## Start:  AIC=5353.31
## CO ~ (AT + AP + AH + AFDP + GTEP + TIT + TAT + TEY + CDP + NOX) -
##     NOX - TIT - CDP - TEY
##
##        Df Sum of Sq   RSS    AIC
## - AP    1       0.8 15218 5351.7
## <none>              15217 5353.3
## - AH    1      74.9 15292 5387.6
## - AT    1     100.1 15317 5399.7
## - GTEP  1     868.0 16085 5760.9
## - AFDP  1    2376.0 17593 6422.6
## - TAT   1    5447.0 20664 7610.7
##
## Step:  AIC=5351.7
## CO ~ AT + AH + AFDP + GTEP + TAT
##
##        Df Sum of Sq   RSS    AIC
## <none>              15218 5351.7
## - AH    1      86.8 15304 5391.7
## - AT    1     120.5 15338 5408.0
## - GTEP  1     991.0 16209 5815.5
## - AFDP  1    2564.3 17782 6499.6
## - TAT   1    5608.5 20826 7666.5

##
## Call:
## lm(formula = CO ~ AT + AH + AFDP + GTEP + TAT, data = gt_2015)
##
## Coefficients:
## (Intercept)           AT           AH         AFDP         GTEP          TAT
##   147.33755      0.02703     -0.01005     -2.51758     -0.18816     -0.23783
```

```
vector <- c(BIC(AT_model), BIC(AP_model), BIC(AH_model), BIC(AFDP_model), BIC(GTEP_model), BIC(TAT_model

library(ggplot2)
df <- data.frame(vector = c(BIC(AT_model), BIC(AP_model), BIC(AH_model), BIC(AFDP_model), BIC(GTEP_model
ggplot(data = df, aes(x = names, y = vector), ylim = c(0, 50000)) + geom_bar(stat = "identity")
```




```
barplot(vector, main = "BIC values", xlab = "Variables",ylab = "Values", names.arg = c("AT", "AP", "AH"
```



###push attempt

```
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'lattice'
```

```
## The following object is masked from 'package:faraway':
##
##     melanoma
```

```
#simplest linear model
simple_linear_model <- lm(CO ~ . - TIT - CDP - TEY - NOX, data = gt_2015)

#5-fold cross validation
cv_5 <- trainControl(method = "cv", number = 5)

#AIC stepwise selected linear model
linear_mod <- train(
          form = CO ~ . - TIT - CDP - TEY- NOX,
          data = gt_2015,
          method = "lmStepAIC",
          trControl = cv_5,
```

```
          trace = FALSE
)

#Linear log model
linear_mod_2 <- train(
          form = log(CO) ~ . - TIT - CDP - TEY - NOX,
          data = gt_2015,
          method = "lmStepAIC",
          trControl = cv_5,
          nvmax = 10,
          trace = FALSE
)

#Lasso model
lasso_mod <- train(
          form = CO ~ . - TIT - CDP - TEY - NOX,
          data = gt_2015,
          method = "lasso",
          trControl = cv_5
)
```

Decision Trees

```
# install.packages('tree')
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v purrr   0.3.4     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x purrr::lift()   masks caret::lift()
## x dplyr::select() masks MASS::select()
```

```
library(tree)
```

```
## Registered S3 method overwritten by 'tree':
##   method     from
##   print.tree cli
```

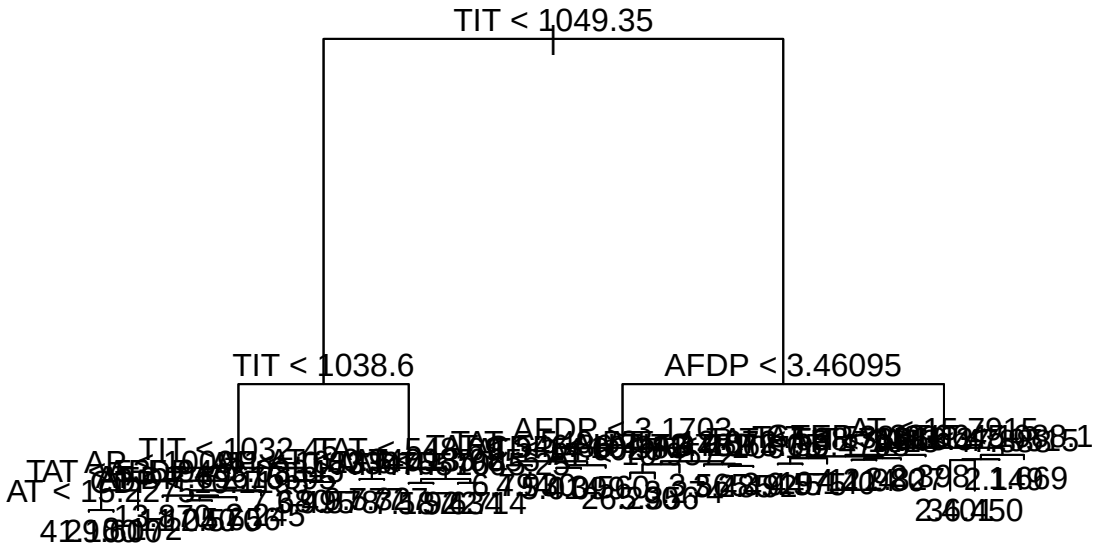
```
RMSE <- function(y, y_hat) {
  rmse <- sqrt(sum(((y_hat - y)^2)/length(y)))
  print(rmse)
}
```

```r
set.seed(1)
train <- gt_2015 %>% dplyr::select(-NOX) %>% sample_frac(0.8)
test <- gt_2015 %>% dplyr::select(-NOX) %>% setdiff(train)
```

```r
tree_CO <- tree(CO ~ . , train,
                  control = tree.control(nobs = length(train$CO),
                                         minsize = 2, mindev=0.001), method = "recursive.partition")
summary(tree_CO)
```

```
##
## Regression tree:
## tree(formula = CO ~ ., data = train, control = tree.control(nobs = length(train$CO),
##     minsize = 2, mindev = 0.001), method = "recursive.partition")
## Variables actually used in tree construction:
## [1] "TIT"  "AP"   "TAT"  "AT"   "AFDP" "CDP"  "AH"   "GTEP"
## Number of terminal nodes:  39
## Residual mean deviance:  0.592 = 3474 / 5868
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -4.75000 -0.35380 -0.03801  0.00000  0.29350 16.99000
```

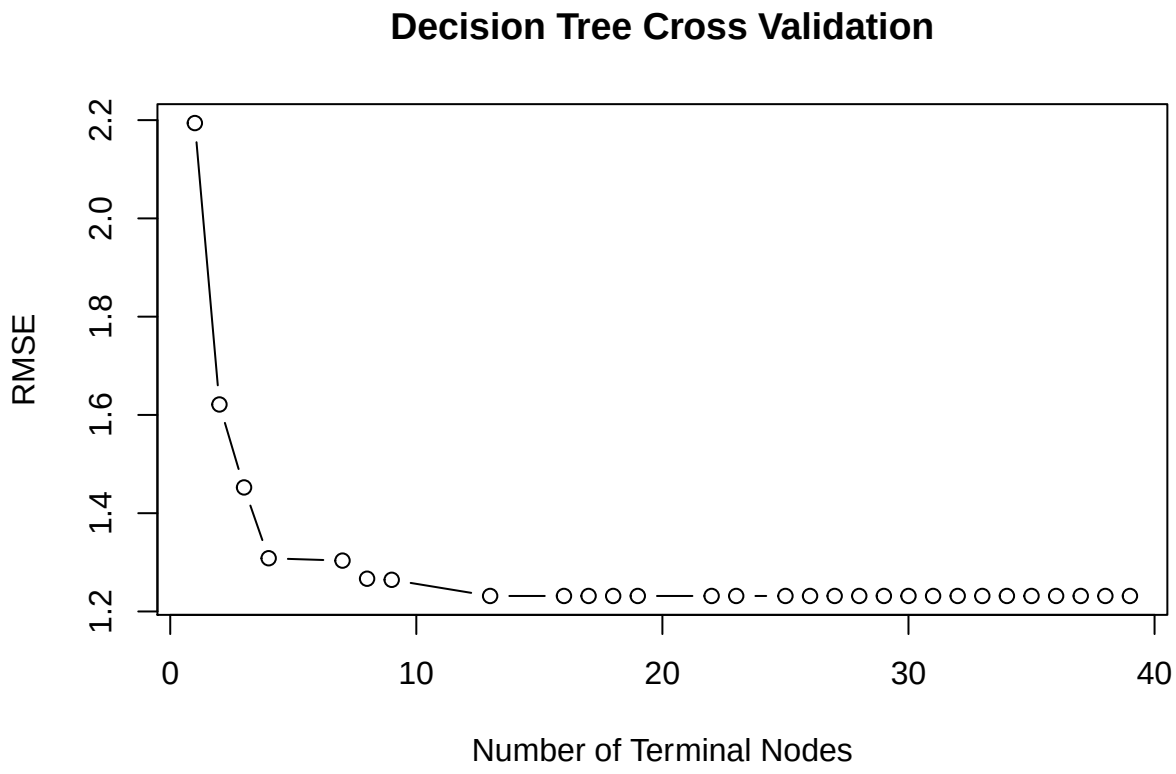
```r
plot(tree_CO)
text(tree_CO, pretty = 0)
```



```
tree_pred <- predict(tree_CO, test)
RMSE(test$CO, tree_pred)
```

```
## [1] 1.237462
```

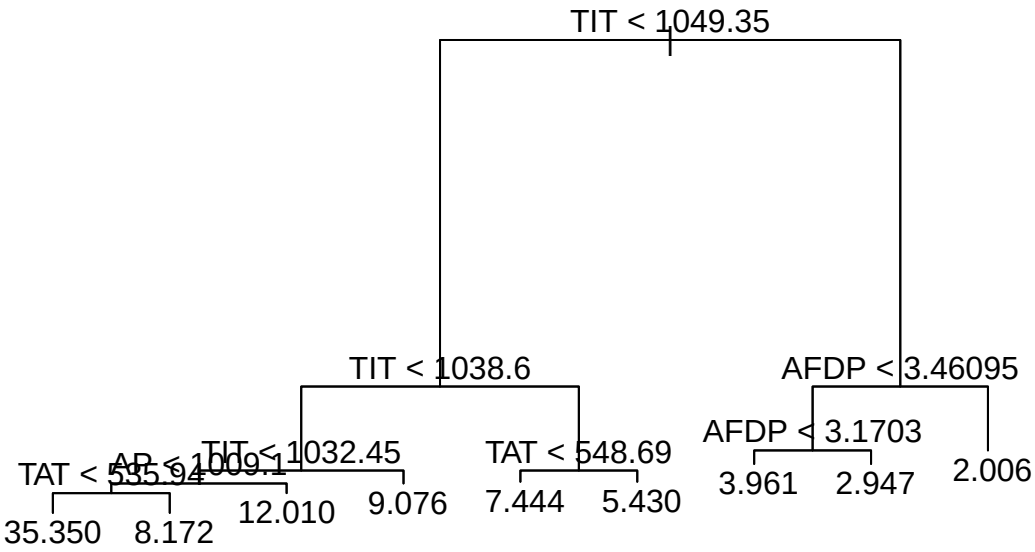
```
cv_info <- cv.tree(tree_CO, FUN = prune.tree)
plot(cv_info$size, sqrt(cv_info$dev / nrow(train)), type = "b", xlab = "Number of Terminal Nodes", ylab
```




```
pruned_tree <- prune.tree(tree_CO, best = 9)
summary(pruned_tree)
```

```
##
## Regression tree:
## snip.tree(tree = tree_CO, nodes = c(10L, 9L, 32L, 11L, 12L, 17L,
## 13L, 7L))
## Variables actually used in tree construction:
## [1] "TIT"  "AP"   "TAT"  "AFDP"
## Number of terminal nodes:  9
## Residual mean deviance:  1.254 = 7398 / 5898
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -10.4400  -0.5023  -0.1009   0.0000   0.4191  34.4500
```

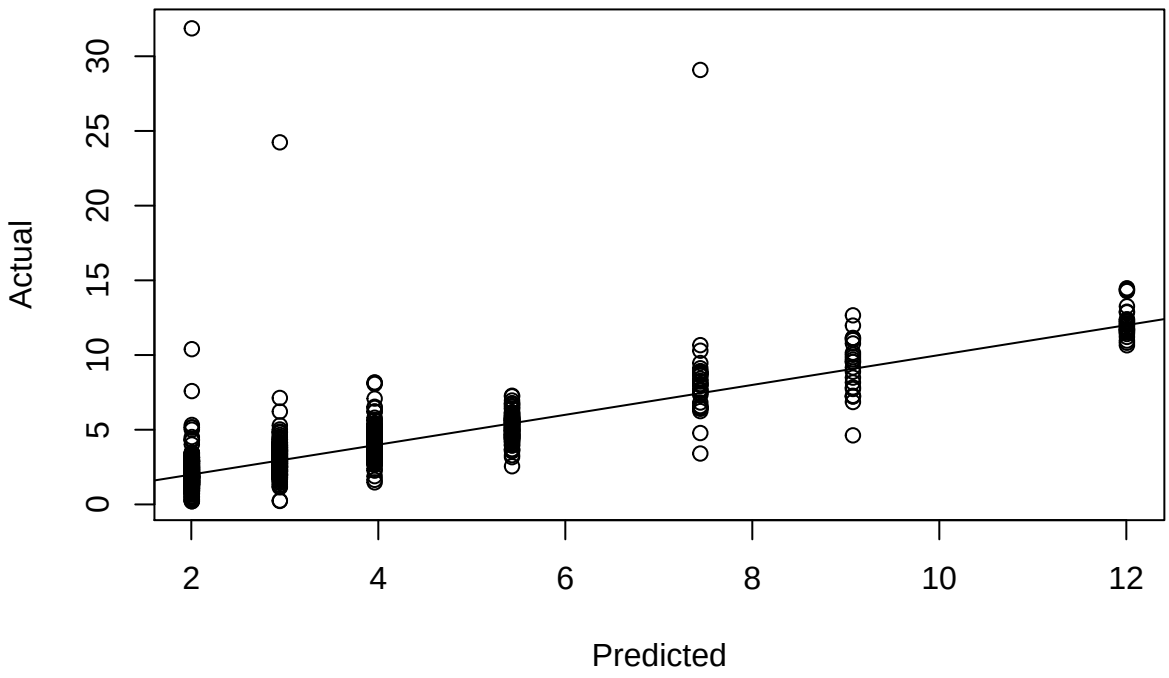
```
plot(pruned_tree)
text(pruned_tree, pretty = 0)
```



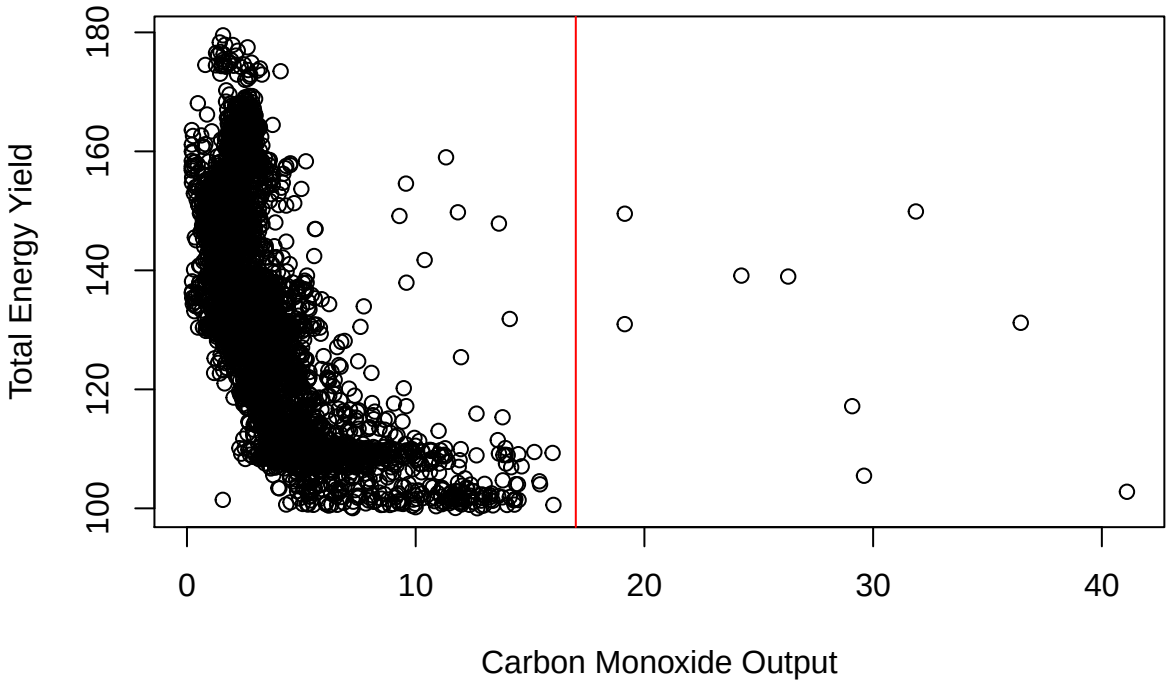

```
tree_pred <- predict(pruned_tree, test)
RMSE(test$CO, tree_pred)
```

```
## [1] 1.405592
```

```
plot(tree_pred, test$CO, xlab = "Predicted", ylab = "Actual")
abline(0, 1)
```




```
plot(gt_2015$CO, gt_2015$TEY, ylab = "Total Energy Yield", xlab = "Carbon Monoxide Output")
abline(v = 17, col = "red")
```

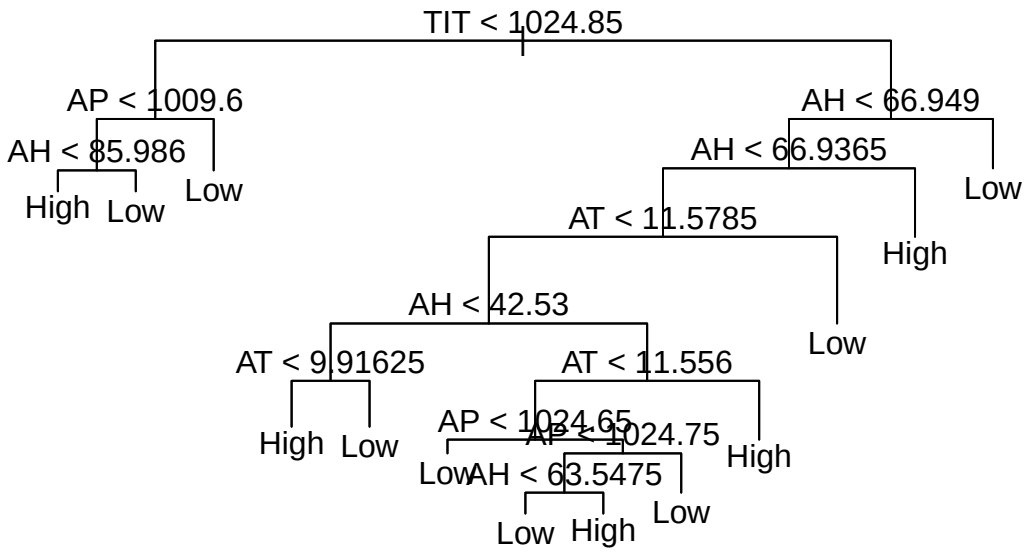



```
data <- gt_2015 %>% mutate(Emissions = as.factor(ifelse(CO > 17, "High", "Low"))) %>% dplyr::select(-NO
high_CO <- data %>% filter(CO > 17) %>% dplyr::select(-CO)
low_CO <- data %>% dplyr::select(-CO) %>% setdiff(high_CO)

set.seed(1)
train <- bind_rows(low_CO %>% sample_frac(7/9), high_CO %>% sample_frac(7/9))
test <- data %>% dplyr::select(-CO) %>% setdiff(train)
```

```
tree <- tree(Emissions ~ . , train,
                  control = tree.control(nobs = length(train$Emissions),
                                         minsize = 1))
summary(tree)
```

```
##
## Classification tree:
## tree(formula = Emissions ~ ., data = train, control = tree.control(nobs = length(train$Emissions),
##     minsize = 1))
## Variables actually used in tree construction:
## [1] "TIT" "AP"  "AH"  "AT"
## Number of terminal nodes:  13
## Residual mean deviance:  0 = 0 / 5730
## Misclassification error rate: 0 = 0 / 5743
```

```
plot(tree)
text(tree, pretty = 0)
```




```
tree_pred <- predict(tree, train, type = "class")
table(predicted = tree_pred, actual = train$Emissions)
```

```
##          actual
## predicted High  Low
##      High    7    0
##      Low     0 5736
```

```
tree_pred <- predict(tree, test, type = "class")
table(predicted = tree_pred, actual = test$Emissions)
```

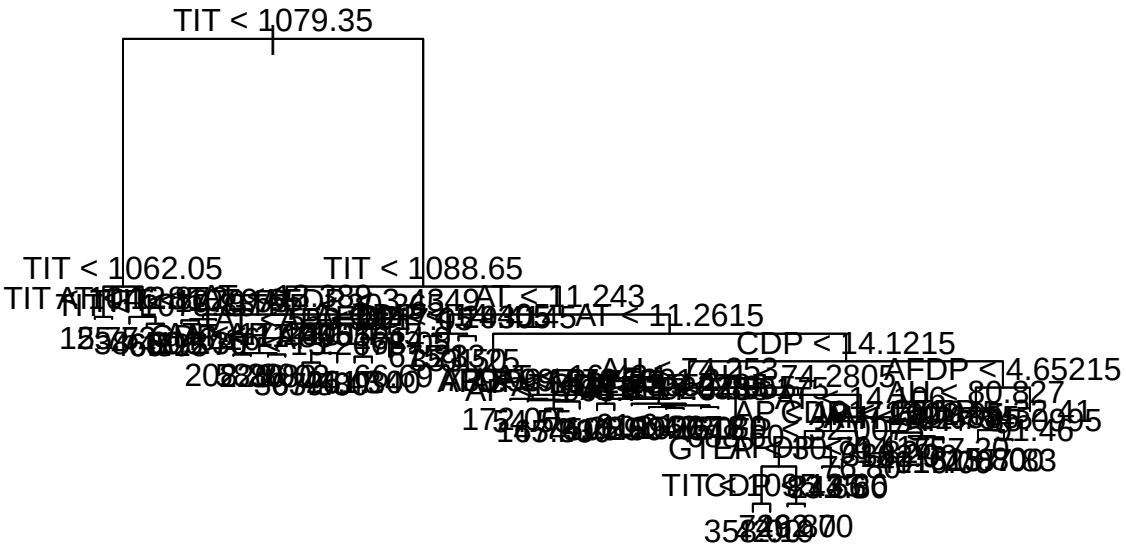
```
##          actual
## predicted High  Low
##      High    0    0
##      Low     2 1639
```

```
set.seed(1)
# train <- gt_2015 %>% mutate(Energy_CO_Ratio = TEY / CO) %>% sample_frac(0.8)
# test <- gt_2015 %>% mutate(Energy_CO_Ratio = TEY / CO) %>% setdiff(train)
train <- gt_2015 %>% mutate(Energy_CO_Ratio = TEY / CO) %>% dplyr::select(-c(NOX, TEY, CO)) %>% sample_
test <- gt_2015 %>% mutate(Energy_CO_Ratio = TEY / CO) %>% dplyr::select(-c(NOX, TEY, CO)) %>% setdiff(
```

```
tree_Energy_CO_Ratio <- tree(Energy_CO_Ratio ~ . , train,
                  control = tree.control(nobs = length(train$Energy_CO_Ratio),
                                         minsize = 2, mindev=0.001), method = "recursive.partition")
summary(tree_Energy_CO_Ratio)
```

```
##
## Regression tree:
## tree(formula = Energy_CO_Ratio ~ ., data = train, control = tree.control(nobs = length(train$Energy_C
##     minsize = 2, mindev = 0.001), method = "recursive.partition")
## Number of terminal nodes:  55
## Residual mean deviance:  496.1 = 2903000 / 5852
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -89.800  -8.799  -1.732   0.000   4.794 359.500
```

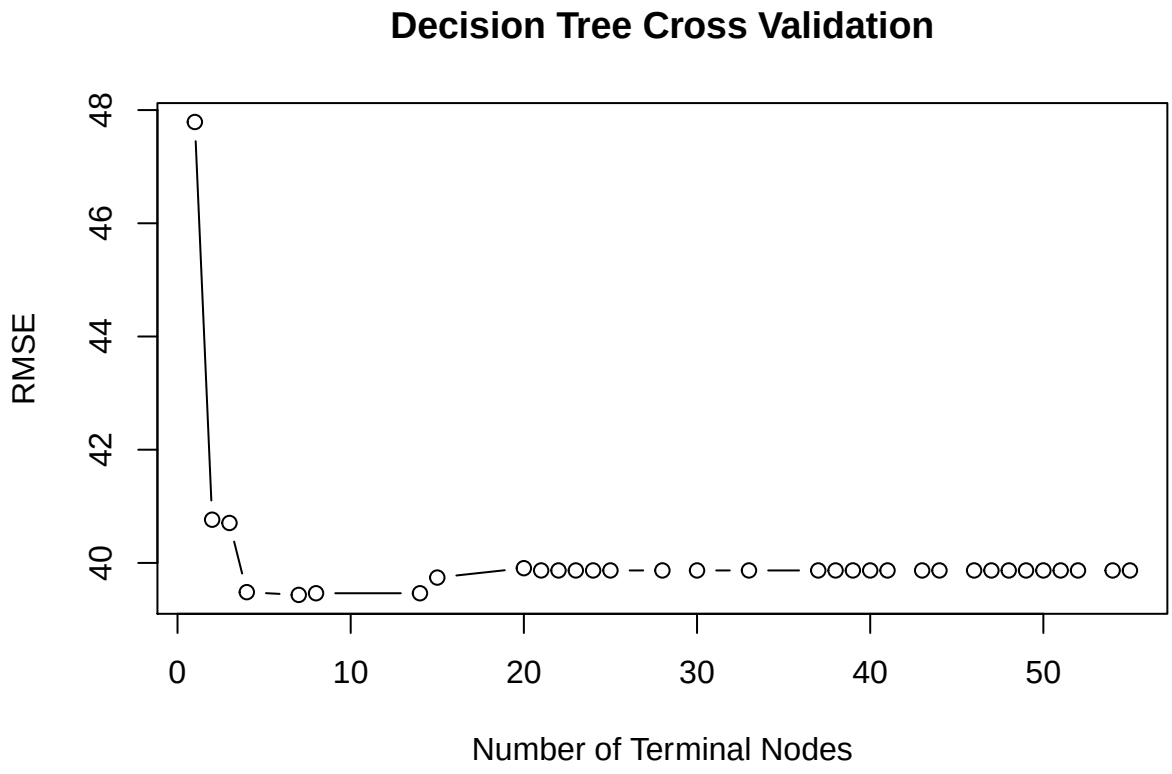
```
plot(tree_Energy_CO_Ratio)
text(tree_Energy_CO_Ratio, pretty = 0)
```




```
tree_pred <- predict(tree_Energy_CO_Ratio, test)
RMSE(test$Energy_CO_Ratio, tree_pred)
```

```
## [1] 48.42792
```

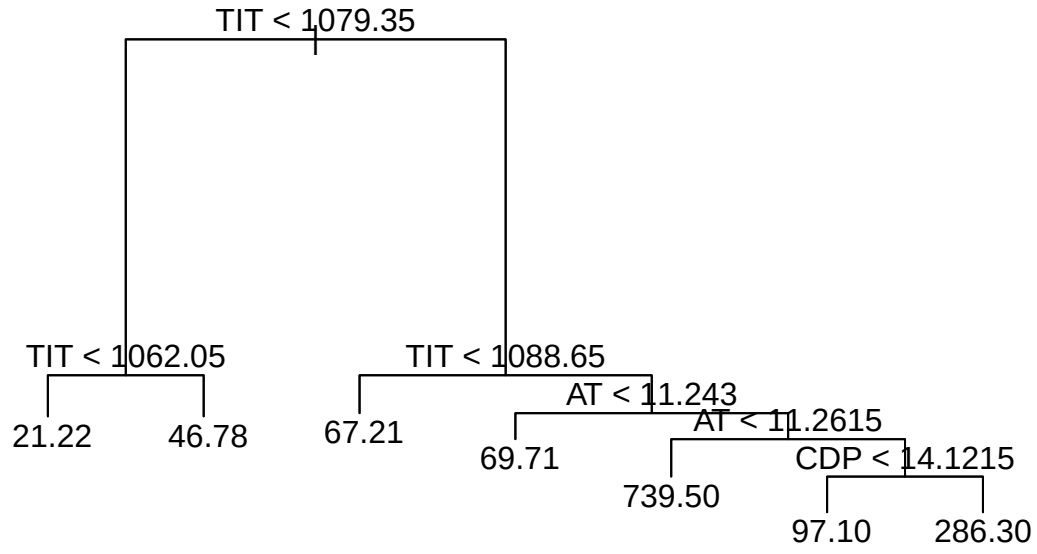
```
cv_info <- cv.tree(tree_Energy_CO_Ratio, FUN = prune.tree)
plot(cv_info$size, sqrt(cv_info$dev / nrow(train)), type = "b", xlab = "Number of Terminal Nodes", ylab
```




```
pruned_tree <- prune.tree(tree_Energy_CO_Ratio, best = 7)
summary(pruned_tree)
```

```
##
## Regression tree:
## snip.tree(tree = tree_Energy_CO_Ratio, nodes = c(4L, 14L, 5L,
## 6L, 62L, 63L))
## Variables actually used in tree construction:
## [1] "TIT" "AT"  "CDP"
## Number of terminal nodes:  7
## Residual mean deviance:  1329 = 7843000 / 5900
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -226.400  -12.500   -3.957    0.000    6.555  639.000
```

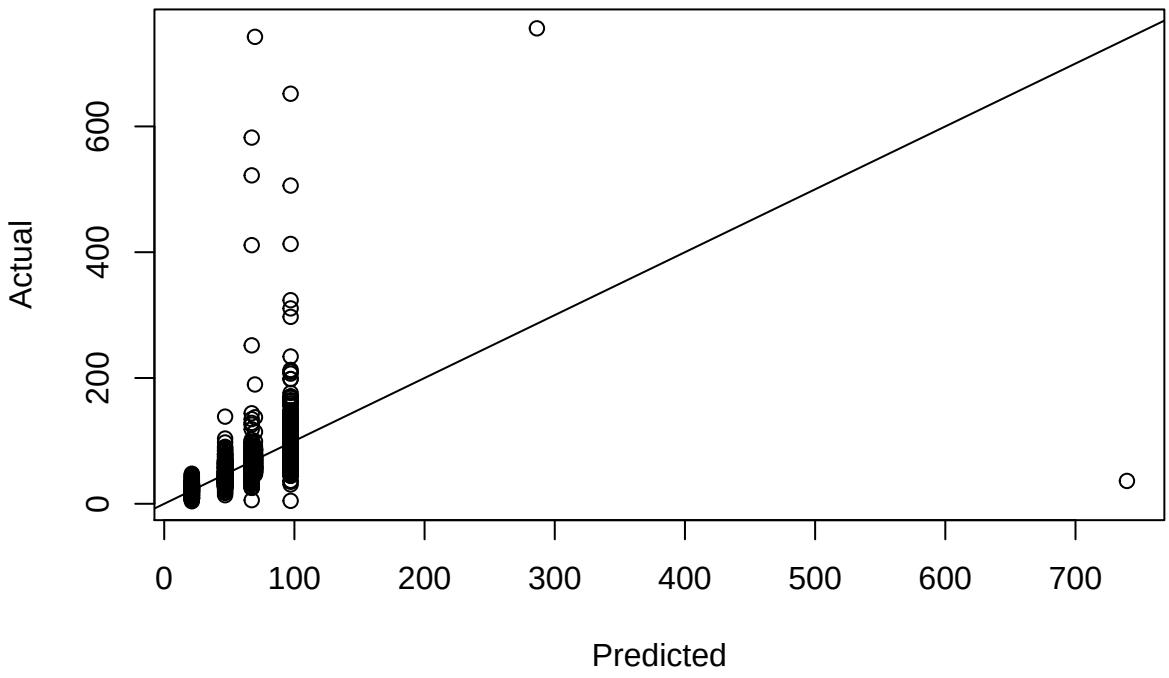
```
plot(pruned_tree)
text(pruned_tree, pretty = 0)
```




```
tree_pred <- predict(pruned_tree, test)
RMSE(test$Energy_CO_Ratio, tree_pred)
```

```
## [1] 45.78954
```

```
plot(tree_pred, test$Energy_CO_Ratio, xlab = "Predicted", ylab = "Actual")
abline(0, 1)
```


Actual
0
200
400
600
0
100
200
300
400
500
600
700
Predicted