STAT 443 Project2

Subin Cho

3/19/2021

```
# packages
library(dplyr, warn.conflicts = FALSE)
library(ggplot2)
library(skimr)
library(car, warn.conflicts = FALSE)
```

```
## Loading required package: carData
```

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(lars)
```

```
## Loaded lars 1.2
```

```
library(MASS, warn.conflicts = FALSE)
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.0-2
```

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
# read data: Gas Turbine CO Emission
gt_initial = read.csv("gt_2012.csv")
```

The dataset contains 7628 instances of 11 sensor measures aggregated over one hour (by means of average or sum) from a gas turbine in 2012 located in Turkey's north western region for the purpose of studying flue gas emissions, namely CO and NOx (NO + NO2).

- `AT`: Ambient temperature (C)
- `AP`: Ambient pressure (mbar)
- `AH`: Ambient humidity (%)
- `AFDP`: Air filter difference pressure (mbar)
- `GTEP`: Gas turbine exhaust pressure (mbar)

- `TIT`: Turbine inlet temperature (C)
- `TAT`: Turbine after temperature (C)
- `CDP`: Compressor discharge pressure (mbar)
- `TEY`: Turbine energy yield (MWH)
- `CO`: Carbon monoxide (mg/m3)
- `NOX`: Nitrogen oxides (mg/m3)

```
# remove NOX
gt = subset(gt_initial, select = -NOX)
colnames(gt)
```

```
##  [1] "AT"   "AP"   "AH"   "AFDP" "GTEP" "TIT"  "TAT"  "TEY"  "CDP"  "CO"
```

```
# Check NAs
sum(is.na(gt))
```

```
## [1] 0
```

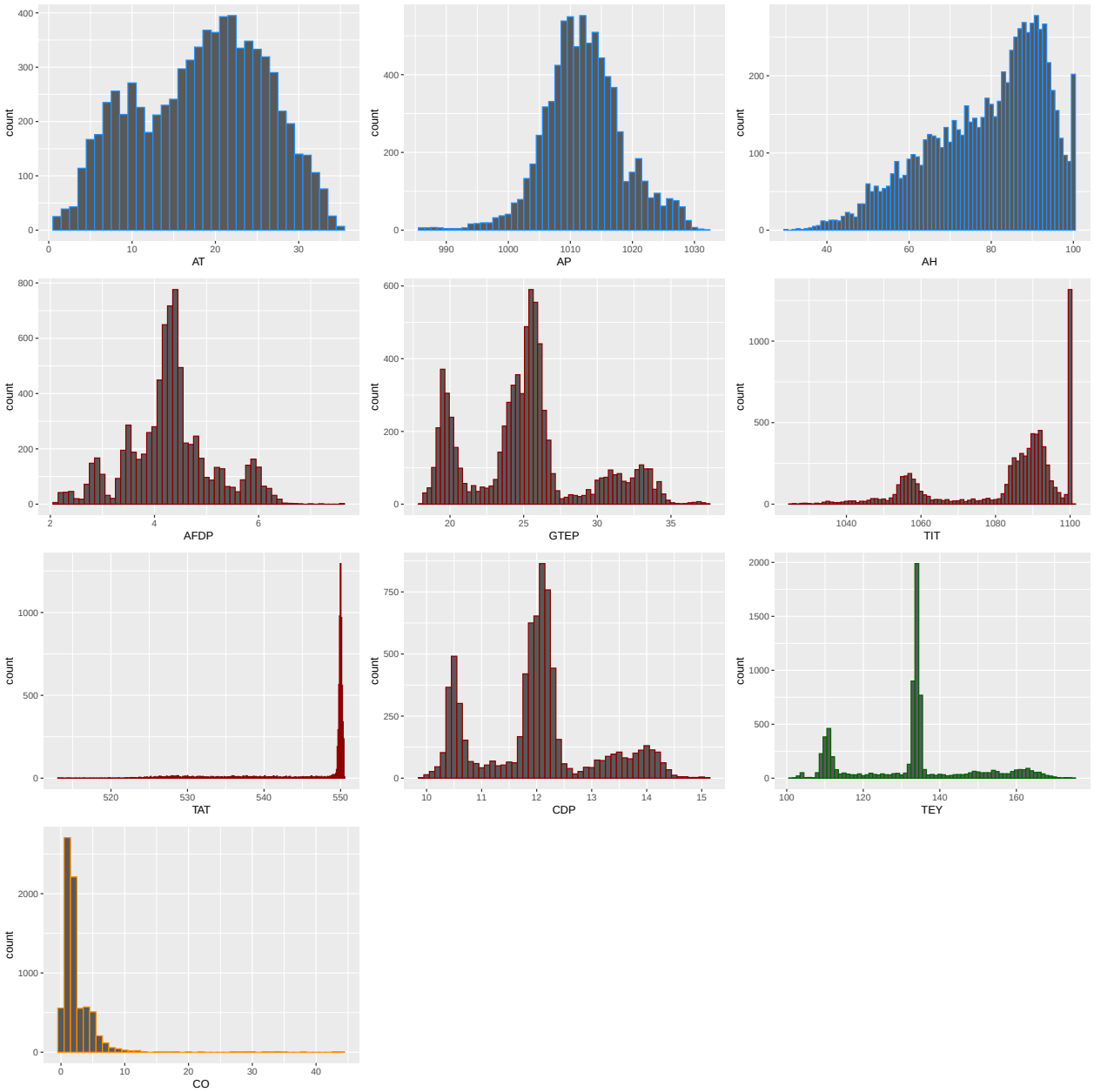
```
# check data
rownames(gt) = NULL
head(gt, n = 10)
```

```
##        AT     AP     AH   AFDP   GTEP    TIT    TAT    TEY    CDP     CO
## 1  6.8594 1007.9 96.799 3.5000 19.663 1059.2 550.00 114.70 10.605 3.1547
## 2  6.7850 1008.4 97.118 3.4998 19.728 1059.3 550.00 114.72 10.598 3.2363
## 3  6.8977 1008.8 95.939 3.4824 19.779 1059.4 549.87 114.71 10.601 3.2012
## 4  7.0569 1009.2 95.249 3.4805 19.792 1059.6 549.99 114.72 10.606 3.1923
## 5  7.3978 1009.7 95.150 3.4976 19.765 1059.7 549.98 114.72 10.612 3.2484
## 6  7.6998 1010.7 92.708 3.5236 19.683 1059.8 549.97 114.72 10.626 3.4467
## 7  7.7901 1011.6 91.983 3.5298 19.659 1060.0 549.87 114.71 10.644 3.4874
## 8  7.7139 1012.7 91.348 3.5088 19.673 1059.8 549.92 114.71 10.656 3.6043
## 9  7.7975 1013.8 90.196 3.5141 19.634 1060.1 550.09 114.72 10.644 3.3943
## 10 8.0820 1015.0 88.597 4.0612 23.406 1083.0 550.21 131.70 11.679 1.9081
```

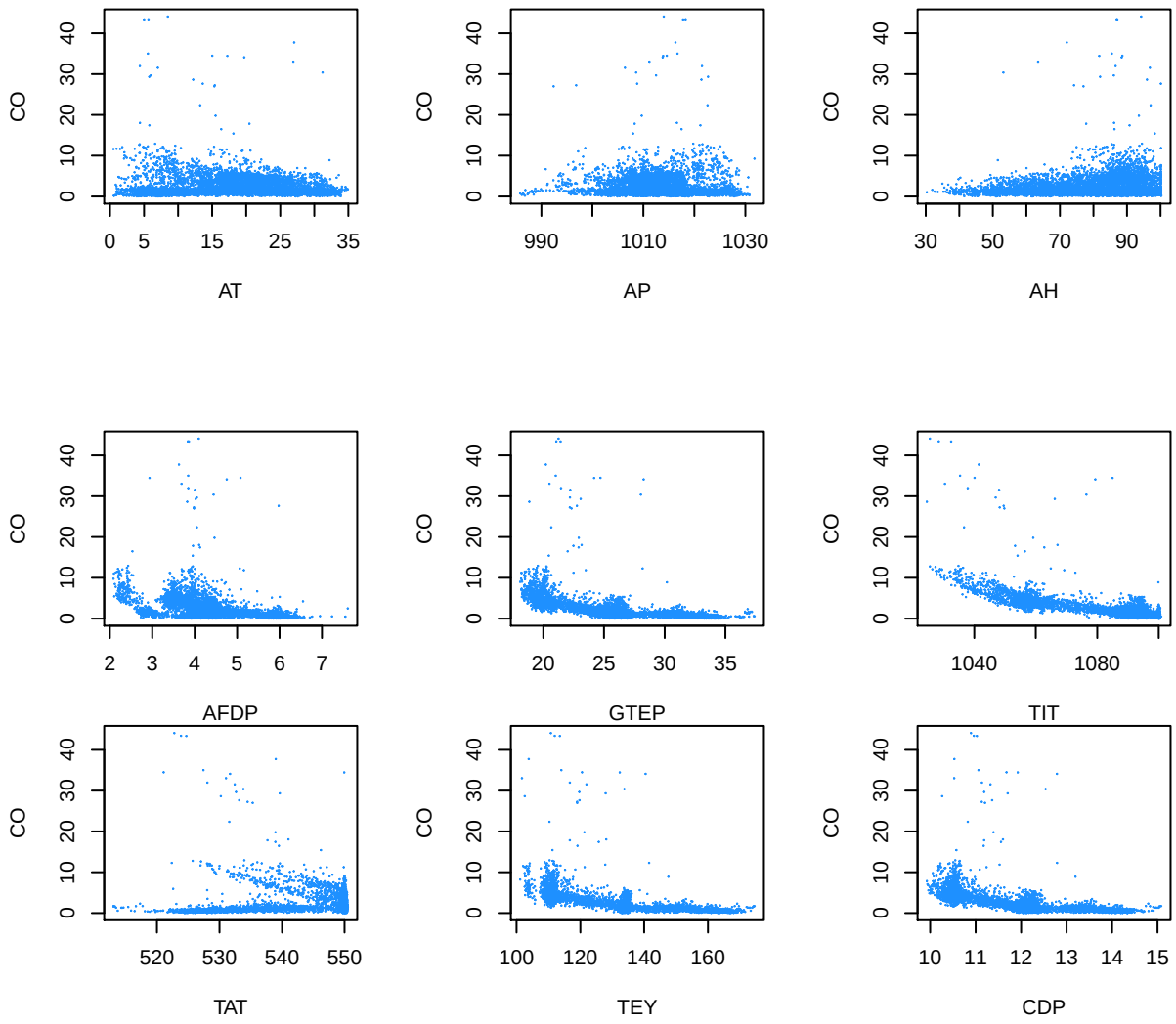
```
# AT, AP, AH
p_AT = gt %>%
  ggplot(aes(x = AT)) +
  geom_histogram(binwidth = 1, color = 'dodgerblue') +
  theme(legend.position = "none")

p_AP = gt %>%
  ggplot(aes(x = AP)) +
  geom_histogram(binwidth = 1, color = 'dodgerblue') +
  theme(legend.position = "none")

p_AH = gt %>%
  ggplot(aes(x = AH)) +
  geom_histogram(binwidth = 1, color = 'dodgerblue') +
  theme(legend.position = "none")

# AFDP, GTEP, TIT, TAT, CDP
p_AFDP = gt %>%
  ggplot(aes(x = AFDP)) +
  geom_histogram(binwidth = 0.1, color = 'darkred') +
```

```
  theme(legend.position = "none")

p_GTEP = gt %>%
  ggplot(aes(x = GTEP)) +
  geom_histogram(binwidth = 0.3, color = 'darkred') +
  theme(legend.position = "none")

p_TIT = gt %>%
  ggplot(aes(x = TIT)) +
  geom_histogram(binwidth = 1, color = 'darkred') +
  theme(legend.position = "none")

p_TAT = gt %>%
  ggplot(aes(x = TAT)) +
  geom_histogram(binwidth = 0.1, color = 'darkred') +
  theme(legend.position = "none")

p_CDP = gt %>%
  ggplot(aes(x = CDP)) +
  geom_histogram(binwidth = 0.1, color = 'darkred') +
  theme(legend.position = "none")

# TEY
p_TEY = gt %>%
  ggplot(aes(x = TEY)) +
  geom_histogram(binwidth = 1, color = 'darkgreen') +
  theme(legend.position = "none")

# CO
p_CO = gt %>%
  ggplot(aes(x = CO)) +
  geom_histogram(binwidth = 1, color = 'darkorange') +
  theme(legend.position = "none")
```

```
gridExtra::grid.arrange(p_AT, p_AP, p_AH,
                        p_AFDP, p_GTEP, p_TIT, p_TAT, p_CDP,
                        p_TEY,
                        p_CO, ncol = 3)
```




```
# EDA
par(mfrow = c(2, 3))
plot(CO ~ . , data = gt, cex = 0.05, col = "dodgerblue")
```




```
# null vs full model
mod_null = lm(CO ~ 1, data = gt)
mod_full = lm(CO ~ ., data = gt)
anova(mod_null, mod_full)
```

```
## Analysis of Variance Table
##
## Model 1: CO ~ 1
## Model 2: CO ~ AT + AP + AH + AFDP + GTEP + TIT + TAT + TEY + CDP
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1   7627 46687
## 2   7618 18621  9     28066 1275.8 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## full model is preferred
```

```
# Removing AFDP
coef(summary(mod_full))[, 4][which.max(coef(summary(mod_full))[, 4])]
```

```
##      AFDP
## 0.1873597
```

```r
mod_1 = lm(CO ~ . - AFDP, data = gt)
summary(mod_1)
```

```
##
## Call:
## lm(formula = CO ~ . - AFDP, data = gt)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.130 -0.593 -0.066  0.394 32.403
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 163.249019   5.656467  28.861  < 2e-16 ***
## AT            0.058315   0.009455   6.168 7.28e-10 ***
## AP           -0.034578   0.003821  -9.049  < 2e-16 ***
## AH            0.016042   0.001604  10.000  < 2e-16 ***
## GTEP         -1.929487   0.088607 -21.776  < 2e-16 ***
## TIT           0.316610   0.026263  12.055  < 2e-16 ***
## TAT          -0.747154   0.037294 -20.034  < 2e-16 ***
## TEY          -0.224337   0.024482  -9.164  < 2e-16 ***
## CDP           1.284358   0.312703   4.107 4.05e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.564 on 7619 degrees of freedom
## Multiple R-squared:  0.6011, Adjusted R-squared:  0.6006
## F-statistic:  1435 on 8 and 7619 DF,  p-value: < 2.2e-16
```

```r
which(coef(summary(mod_1))[, 4] > 0.05)
```

```
## named integer(0)
```

```r
# new data without AFDP
gt_2 = subset(gt, select = -AFDP)
mod_1 = lm(CO ~ ., data = gt_2)
```

```r
# AIC
extractAIC(mod_1) ## AIC for the full model
```

```
## [1]    9.000 6827.344
```

```r
# BIC
extractAIC(mod_1, k = log(nrow(gt_2)))
```

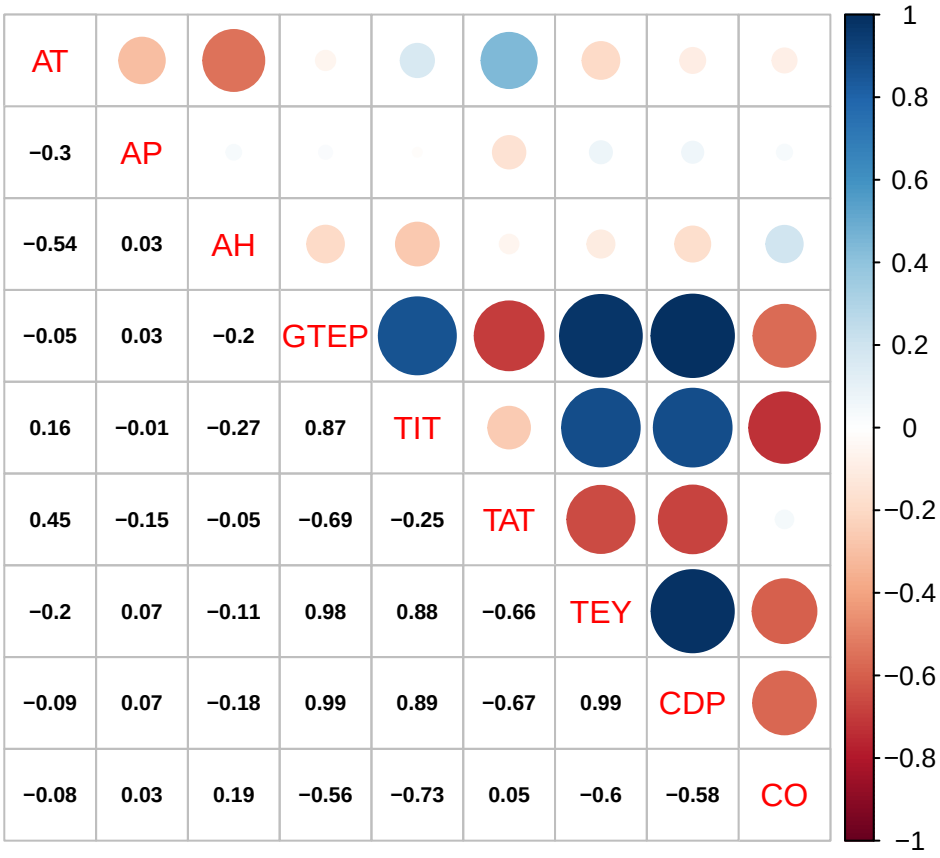
```
## [1]    9.000 6889.801
```

```r
# stepwise
step(mod_1, direction = "backward", trace = 0)
```

```
##
## Call:
## lm(formula = CO ~ AT + AP + AH + GTEP + TIT + TAT + TEY + CDP,
##     data = gt_2)
##
## Coefficients:
## (Intercept)           AT           AP           AH         GTEP          TIT
```

```
##    163.24902      0.05832     -0.03458      0.01604     -1.92949      0.31661
##          TAT          TEY          CDP
##     -0.74715     -0.22434      1.28436
```

```
# Check multicollinearity issues
corrplot.mixed(cor(gt_2), lower.col = "black", number.cex = 0.7)
```




```
round(cor(gt_2), 2)
```

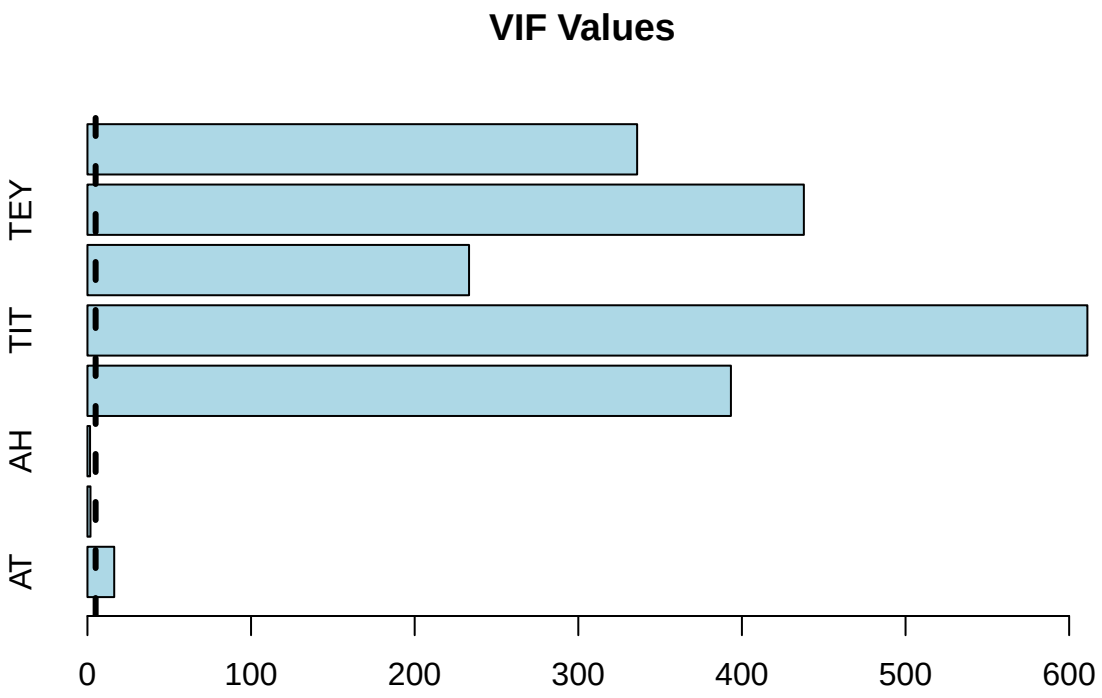
```
##          AT    AP    AH  GTEP   TIT   TAT   TEY   CDP    CO
## AT     1.00 -0.30 -0.54 -0.05  0.16  0.45 -0.20 -0.09 -0.08
## AP    -0.30  1.00  0.03  0.03 -0.01 -0.15  0.07  0.07  0.03
## AH    -0.54  0.03  1.00 -0.20 -0.27 -0.05 -0.11 -0.18  0.19
## GTEP -0.05  0.03 -0.20  1.00  0.87 -0.69  0.98  0.99 -0.56
## TIT    0.16 -0.01 -0.27  0.87  1.00 -0.25  0.88  0.89 -0.73
## TAT    0.45 -0.15 -0.05 -0.69 -0.25  1.00 -0.66 -0.67  0.05
## TEY   -0.20  0.07 -0.11  0.98  0.88 -0.66  1.00  0.99 -0.60
## CDP   -0.09  0.07 -0.18  0.99  0.89 -0.67  0.99  1.00 -0.58
## CO    -0.08  0.03  0.19 -0.56 -0.73  0.05 -0.60 -0.58  1.00
```

```
vif(mod_1) # AT, GTEP, TIT, TAT, TEY, CDP are problematic
```

```
##          AT          AP          AH        GTEP         TIT         TAT         TEY
##   16.369849    1.856871    1.597472 393.275831 611.146829 233.233400 437.861278
##         CDP
## 335.961711
```

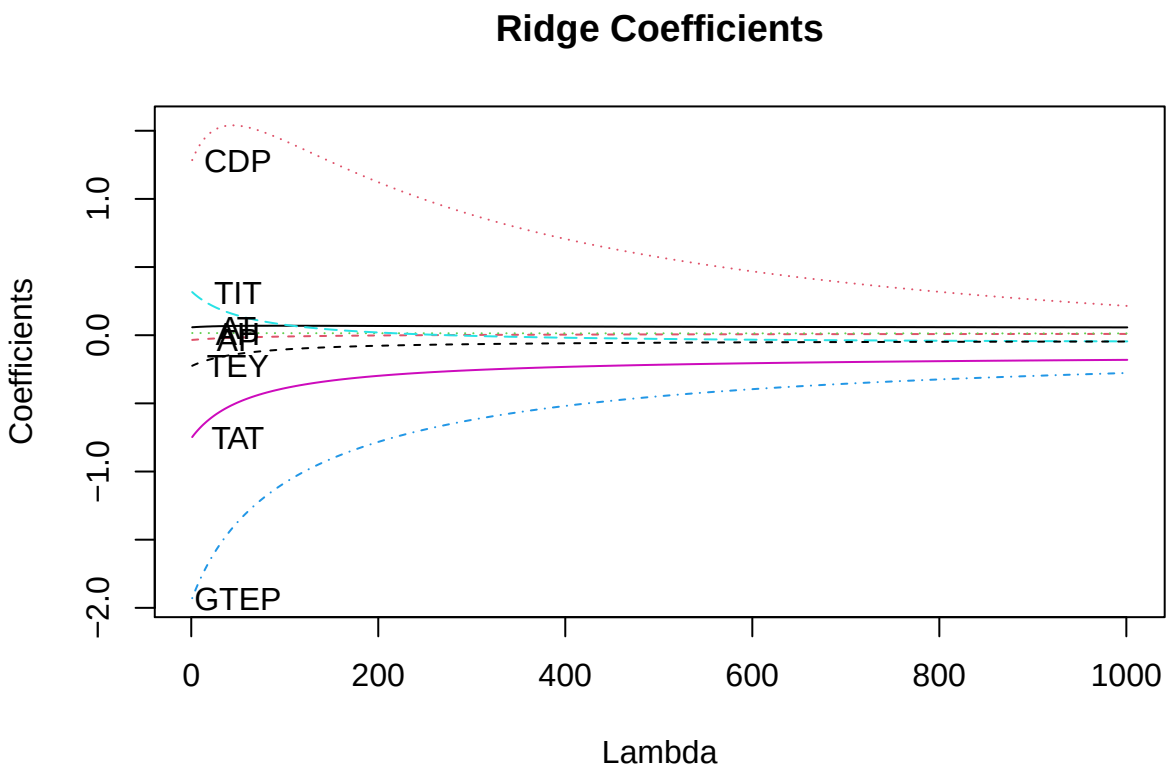
```
# create horizontal bar chart to display each VIF value
barplot(vif(mod_1), main = "VIF Values", horiz = TRUE, col = "lightblue")
```

```r
# add vertical line at 5
abline(v = 5, lwd = 3, lty = 2)
```

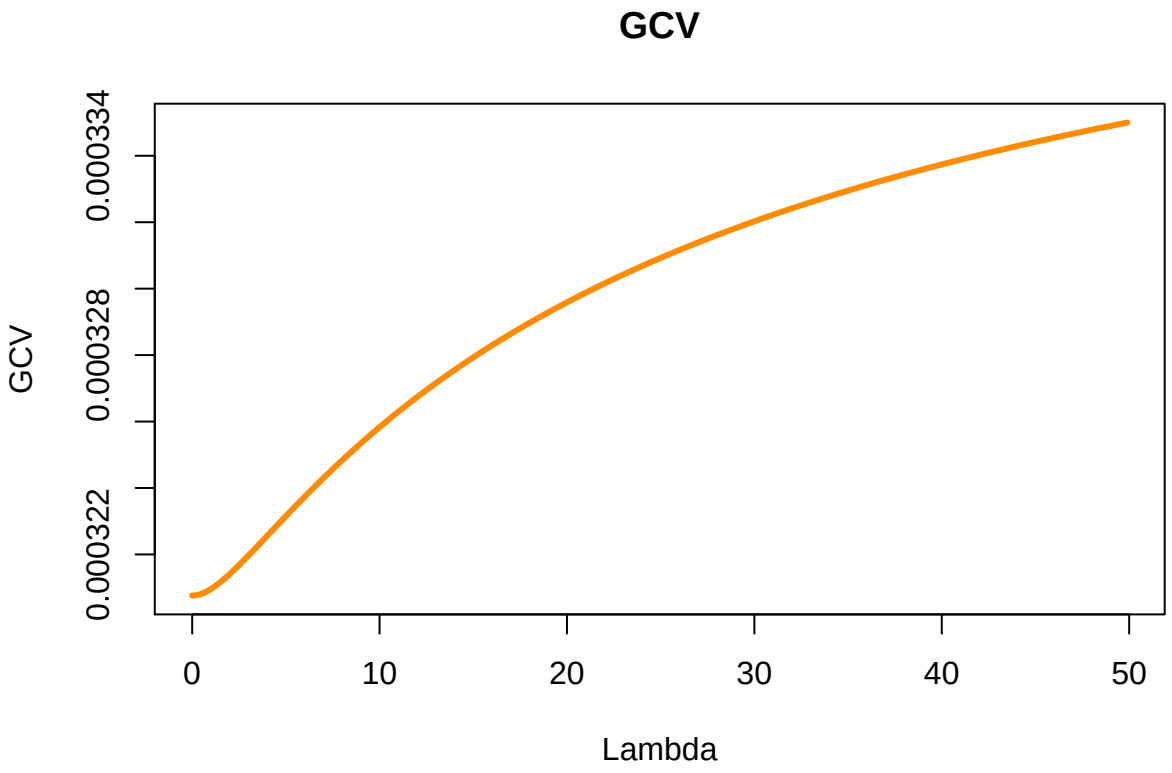



```r
# Multicolinearity
## Principal Component Regression, Ridge Regression, Stepwise Regression
```

```r
# Ridge Regression
fit = lm.ridge(CO ~ ., gt_2, lambda = seq(0, 100, by = 0.1))

#par(mfrow = c(1,2))
matplot(coef(fit)[, -1], type = "l", xlab = "Lambda", ylab = "Coefficients")
text(rep(50, 8), coef(fit)[1,-1], colnames(gt_2)[1:8])
title("Ridge Coefficients")
```




```
# use GCV to select the best lambda
plot(fit$lambda[1:500], fit$GCV[1:500], type = "l", col = "darkorange",
         ylab = "GCV", xlab = "Lambda", lwd = 3)
title("GCV")
```



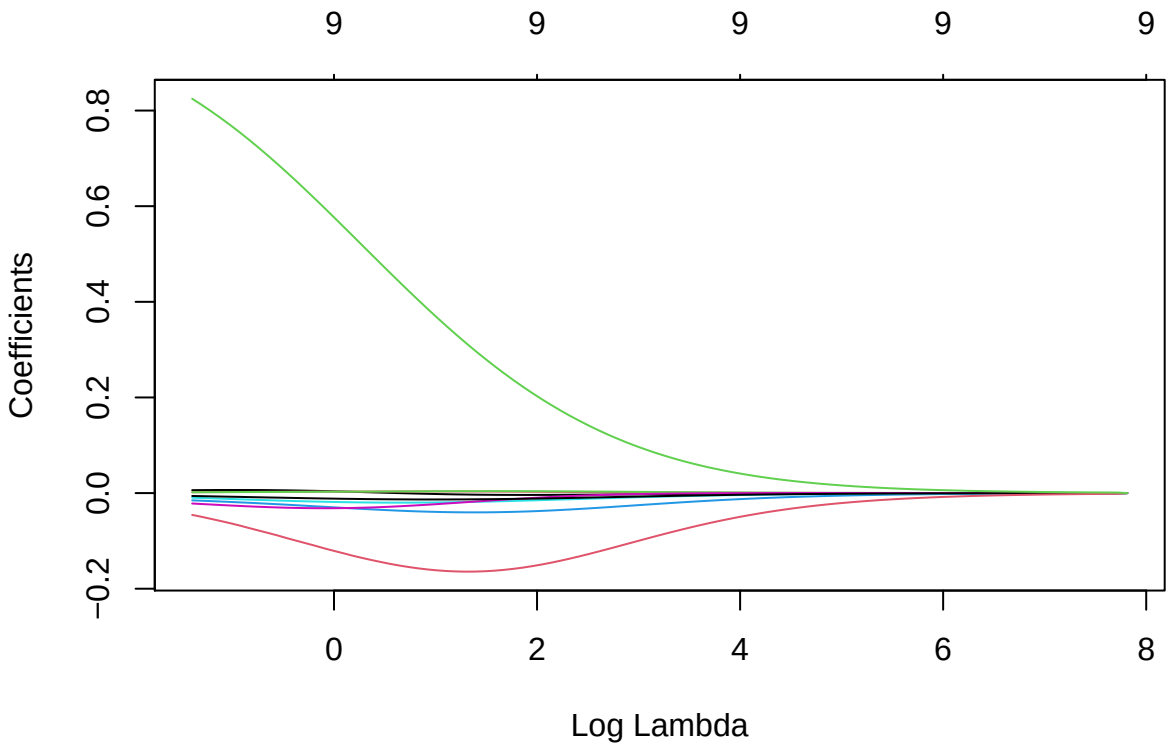
```r
fit$lambda[which.min(fit$GCV)]
```

```
## [1] 0
```

```r
# Ridge Regression 2
set.seed(3)
fit2 = cv.glmnet(data.matrix(gt_2), gt_2$CO,
                 nfolds = 10, alpha = 0)
coef(fit2, s = "lambda.min")
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                          1
## (Intercept) 22.395640050
## AT           0.006094845
## AP           0.001835954
## AH           0.001718463
## GTEP        -0.015256747
## TIT         -0.009640791
## TAT         -0.021798310
## TEY         -0.006098142
## CDP         -0.045617911
## CO           0.824705880
```

```r
plot(fit2$glmnet.fit, "lambda")
```

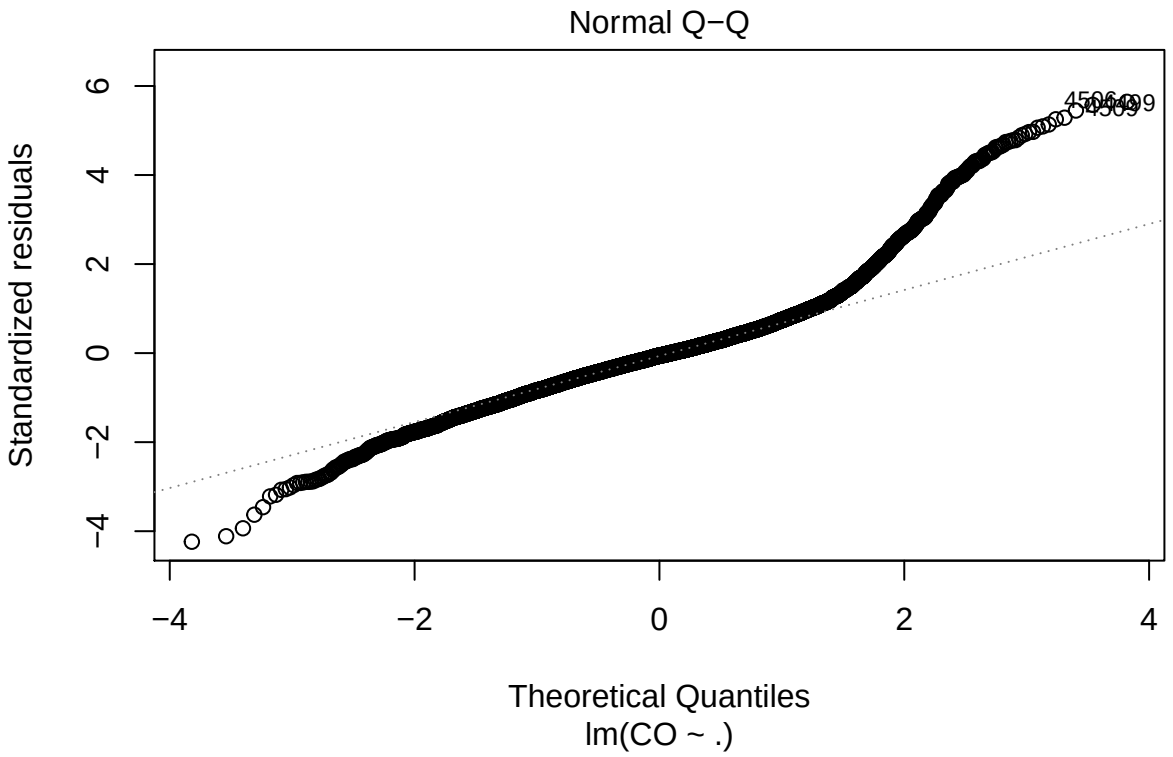



```r
# Higher Order Terms?
```

```r
# Removing influential observations
cd = cooks.distance(mod_1)
#cd[which(cd > 4 / length(cd))]
length(cd[which(cd > 4 / length(cd))])
```

```
## [1] 151
```

```
mod_1_cd = lm(CO ~ ., data = gt_2, subset = cd <= 4 / length(cd))
plot(mod_1_cd, which = 2)
```



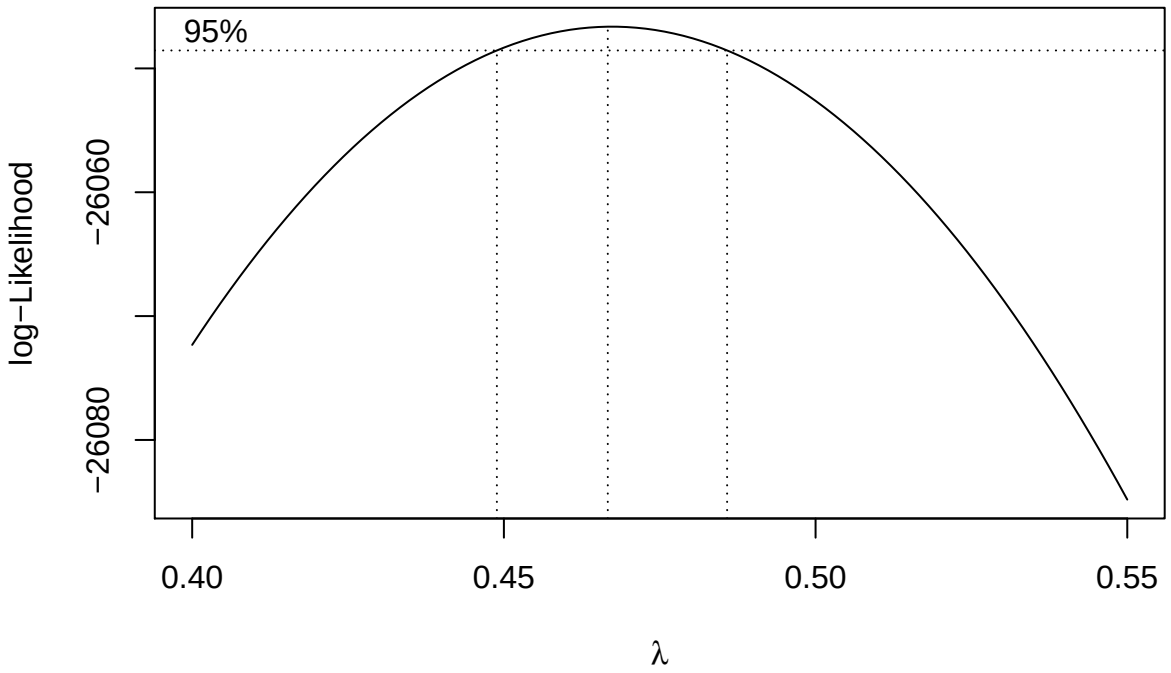

```
# calculate LOOCV RMSE
calc_loocv_rmse = function(model){
  sqrt(mean((resid(model) / (1 - hatvalues(model)))^2))
}

calc_loocv_rmse(mod_1_cd)
```

```
## [1] 0.7688012
```

```
# Breusch-Pagan Test
bptest(mod_1_cd)
```

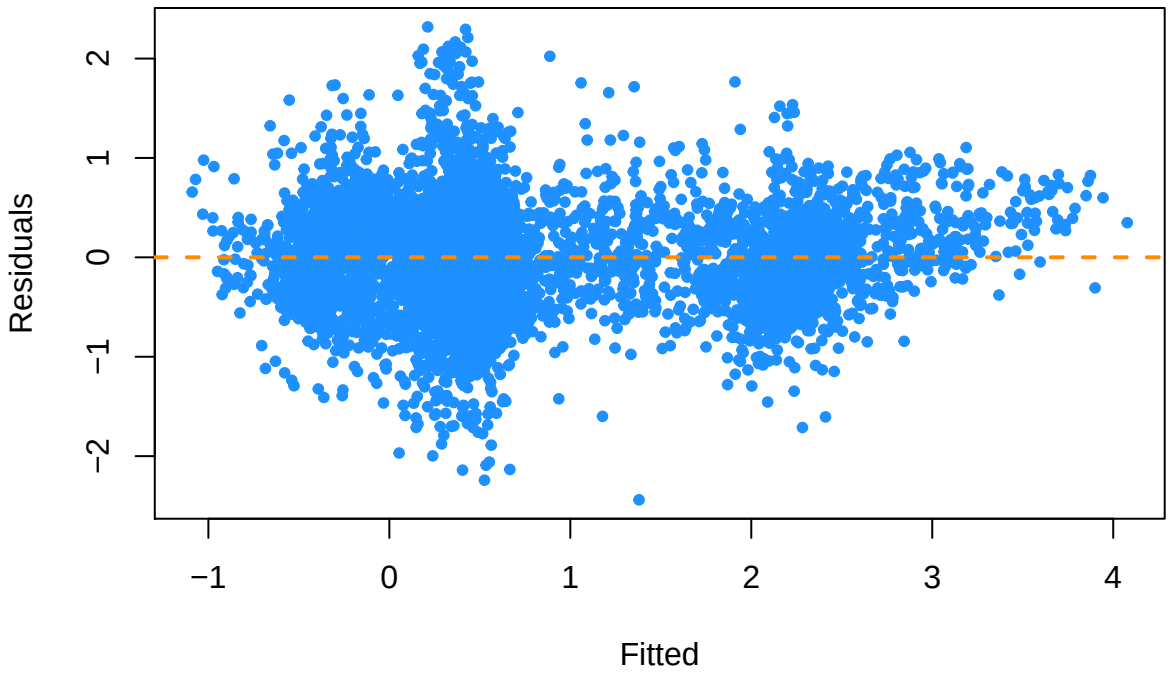
```
##
##  studentized Breusch-Pagan test
##
## data:  mod_1_cd
## BP = 599.81, df = 8, p-value < 2.2e-16
```

```
boxcox(mod_1_cd, plotit = TRUE, lambda = seq(0.4, 0.55, by = 0.01))
```




```
lambda = 0.47
mod_1_cox = lm((((CO ^ lambda) - 1) / lambda) ~ ., data = gt_2,
               subset = cd <= 4 / length(cd))
```

```
plot(fitted(mod_1_cox), resid(mod_1_cox), col = "dodgerblue",
     pch = 20, cex = 1, xlab = "Fitted", ylab = "Residuals")
abline(h = 0, lty = 2, col = "darkorange", lwd = 2)
```




```
#??????
```