PM-GroupProject

Dennis Goldenberg, Solomon Gao, Suleman Sadiq, Alana Berson

2024-04-23

```
library(readxl)
library(ggplot2)
suppressWarnings(library(zoo))
library(nlme)
suppressWarnings(library(forecast))
library(rpart)
suppressWarnings(library(rpart.plot))
library(tree)
```

```
## Warning: package 'tree' was built under R version 4.3.3
```

```
suppressWarnings(library(randomForest))
```

1. Data Exploration and Preprocessing

Reading in USA fatality data:

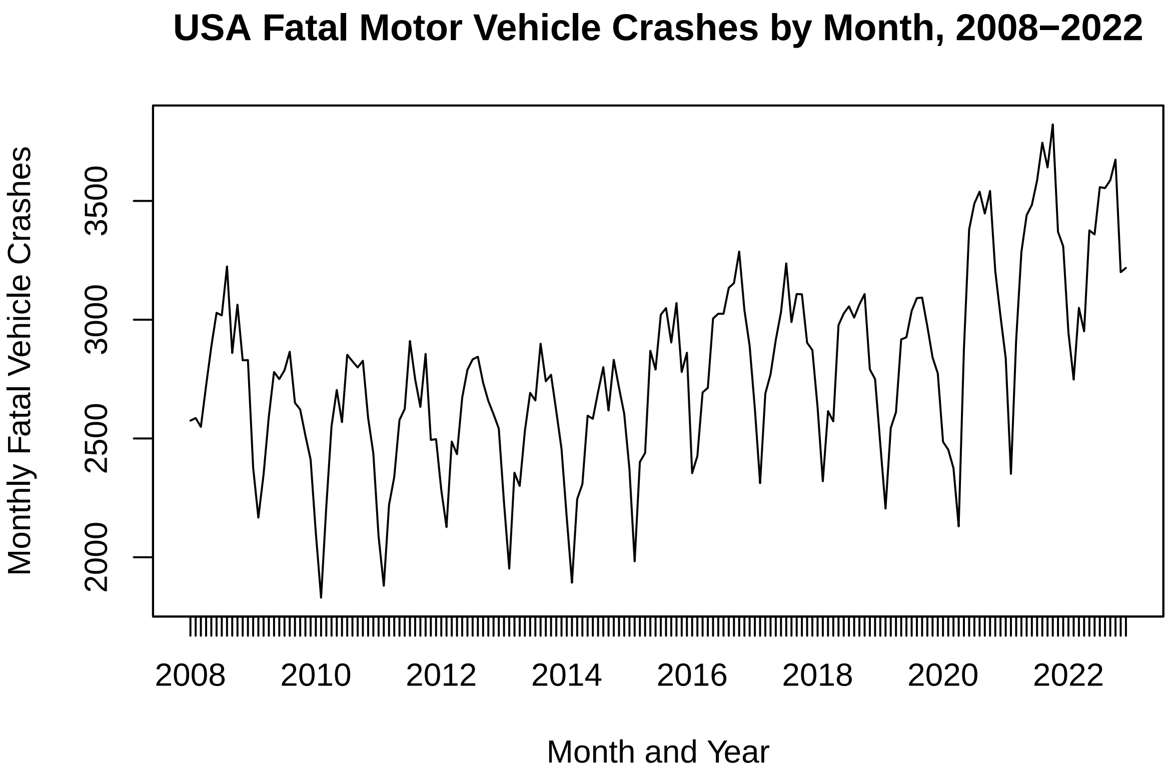
```
USAFat <- as.vector(t(as.matrix(read_excel("data/USA-FatalCrashes.xlsx",
  range = "B8:M22", col_names = FALSE,.name_repair = "unique_quiet"))))
```

Reading in USA population data (in hundreds of thousands):

```
USAPOP <- read_excel("data/USAPOP.xlsx")
colnames(USAPOP) <- c("Year", "Alabama","Alaska","Arizona","Arkansas",
 "California","Colorado","Connecticut","Delaware","D.C.","Florida","Georgia",
 "Hawaii","Idaho","Illinois","Indiana","Iowa","Kansas","Kentucky","Louisiana",
 "Maine","Maryland","Massachusetts","Michigan","Minnesota","Mississippi",
 "Missouri","Montana", "Nebraska","Nevada", "New Hampshire","New Jersey",
 "New Mexico","New York","North Carolina", "North Dakota","Ohio", "Oklahoma",
 "Oregon", "Pennsylvania", "Rhode Island","South Carolina", "South Dakota",
 "Tennessee","Texas", "Utah", "Vermont", "Virginia", "Washington",
 "West Virginia","Wisconsin", "Wyoming")
USAPOP$Year <- (2008:2023)*1000
USAPOP <- USAPOP/1000
USAPOP$Year <- as.integer(USAPOP$Year)
```

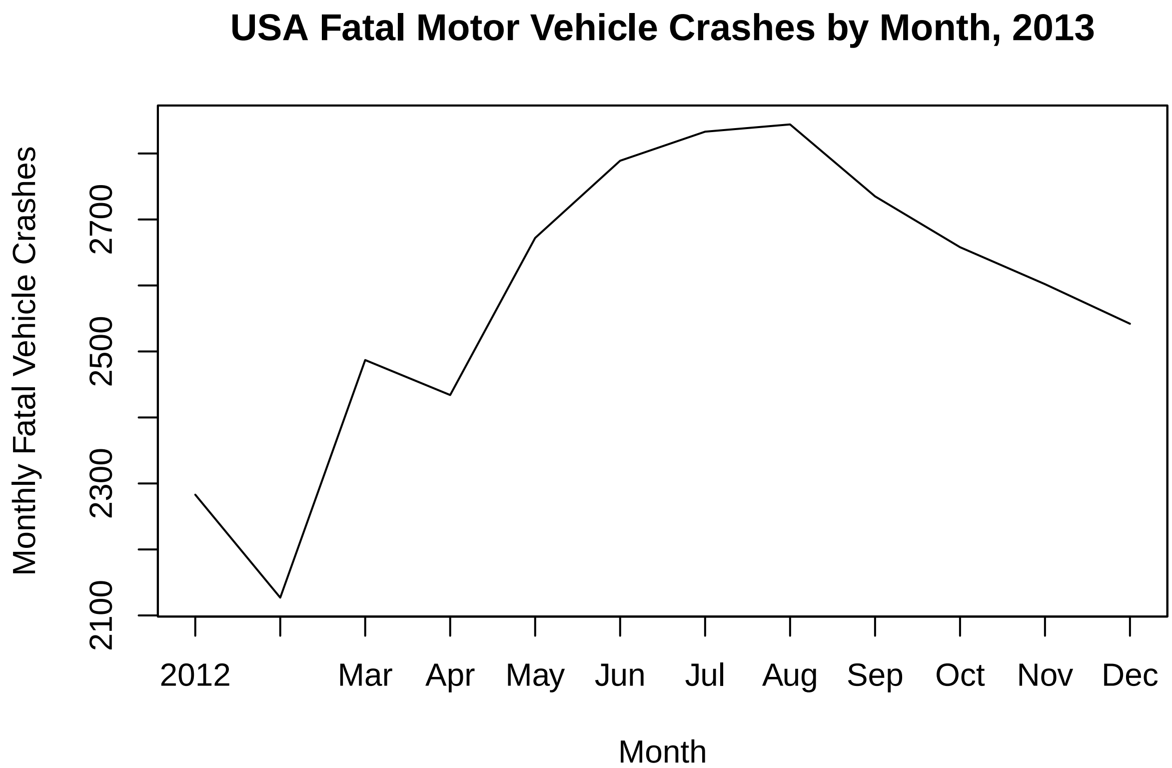
a. Seasonality of Data

```
USADate <- as.yearmon("2008-01") + seq((1/12),(180/12), by = (1/12)) - (1/12)
USAData <- zoo(USAFat, USADate)
plot(USAData, main = "USA Fatal Motor Vehicle Crashes by Month, 2008-2022",
     xlab = "Month and Year", ylab = "Monthly Fatal Vehicle Crashes")
```



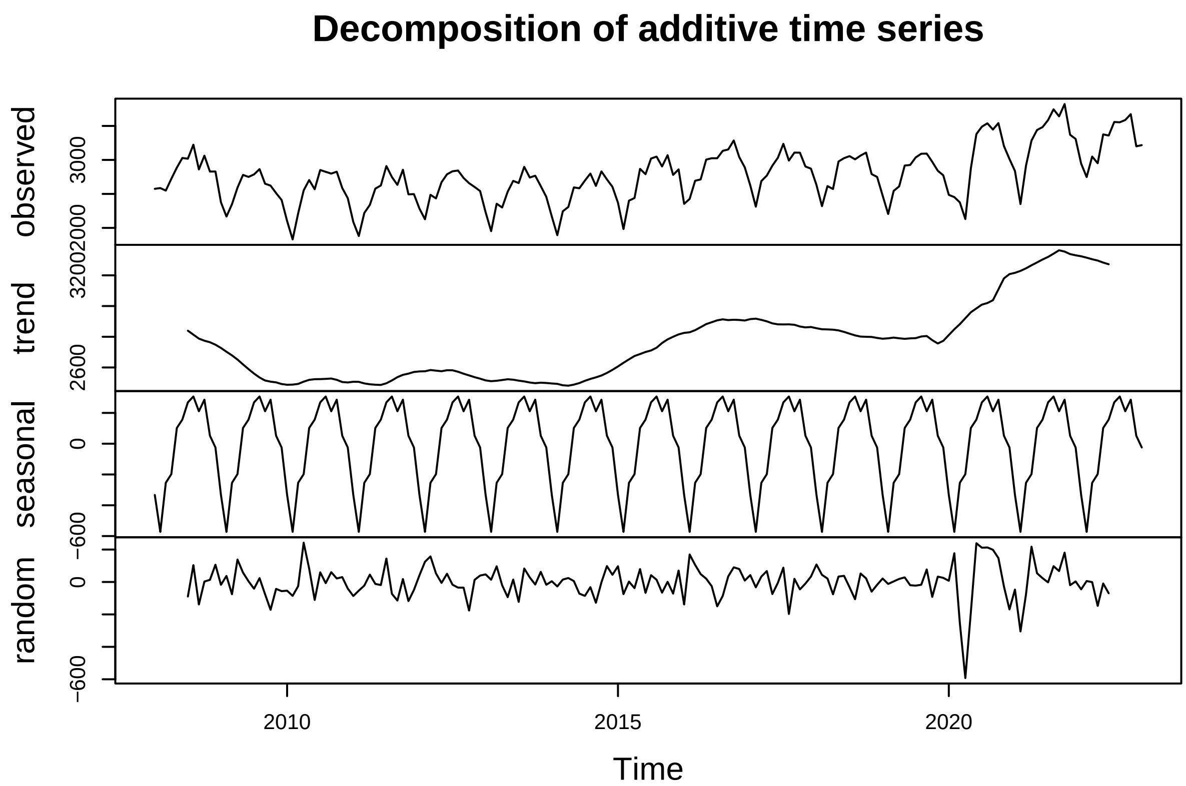
Examining a typical year:

```
plot(USAData[49:60], main = "USA Fatal Motor Vehicle Crashes by Month, 2013",
     xlab = "Month", ylab = "Monthly Fatal Vehicle Crashes")
```



Looking at the Seasonality:

```r
tsUSA <- ts(USAFat, start = 2008, freq = 12)
plot(decompose(tsUSA, type = "add"))
```



b. Collecting fatality data on states that legalized pre-2022

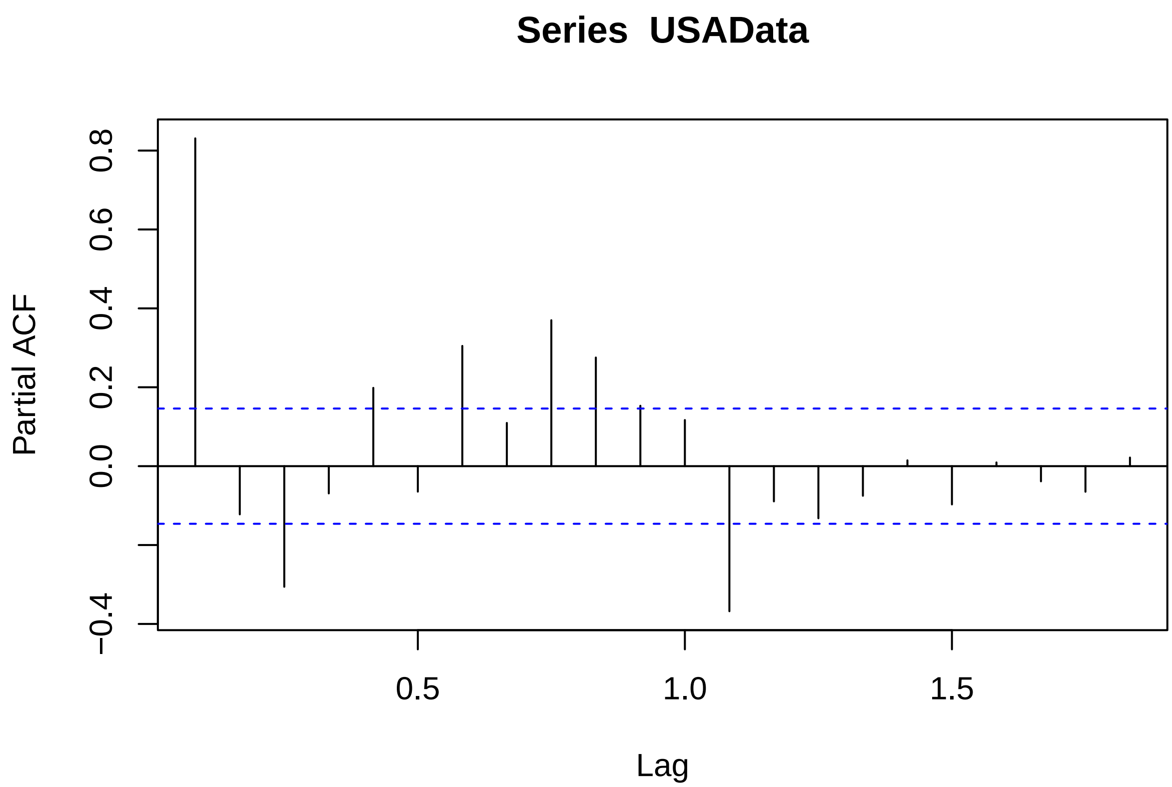
```r
legStates <- c("Colorado", "Washington", "D.C.","Oregon", "Alaska","California",
               "Massachusetts", "Nevada", "Maine", "Vermont", "Michigan",
               "Illinois", "Arizona", "Montana", "New Jersey", "New York",
               "New Mexico", "Virginia", "Connecticut")
legEffDate <- as.yearmon("2012-11") +
  (c(1,2,29,33,39,49,50,51,52,69,74,87,98,99,101,102,105,105,105)/12) - (1/12)
```

c. Calculating State by State Population Data

2. Modeling

a. Autocorrelation function

```r
pacf(USAData)
```




```
arima(USAData, order = c(1,0,0), method = "ML")
```

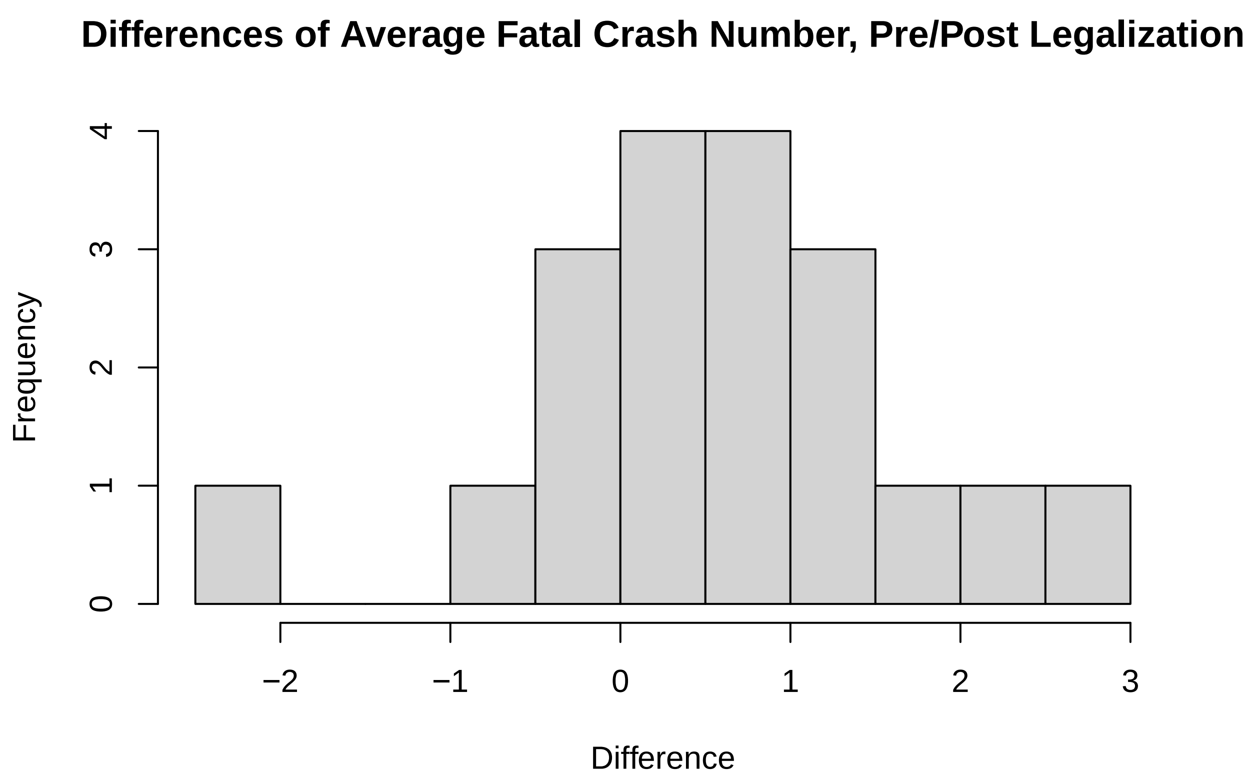
```
##
## Call:
## arima(x = USAData, order = c(1, 0, 0), method = "ML")
##
## Coefficients:
##          ar1  intercept
##       0.8334  2787.4896
## s.e.  0.0406    93.9691
##
## sigma^2 estimated as 46541:  log likelihood = -1223.33,  aic = 2452.66
```

b. Difference of Means on Panel Data

I do a difference of means test to see if there is a change in the average number of fatal crashes per 100,000 in the year pre-legalization vs. the year when legalization went into effect. I linearly interpolate between population estimates to get the population estimate to divide by (i.e. if legalization went into effect at March 2015, I take 2/12 of the population of 2015 and add 10/12 of the population of 2014 for the pre-legalization population):

```
effDatefracs <- as.numeric(legEffDate) - as.integer(legEffDate)
yearsLeg <- as.integer(legEffDate)
avgs_pre <- c()
avgs_post <- c()
for(i in 1:length(legStates)){
  stateInfo <- as.vector(t(as.matrix(read_excel(
```

```r
    paste("data/",legStates[i],".xlsx",sep = ""),range = "B9:M23",
    col_names = FALSE,.name_repair = "unique_quiet"))))

  #linear interpolation
  pop_est_pre <- effDatefracs[i] * USAPOP[yearsLeg[i] - 2007,legStates[i]] +
    (1 - effDatefracs[i]) * USAPOP[yearsLeg[i] - 2008,legStates[i]]
  pop_est_post <- (1-effDatefracs[i])*USAPOP[yearsLeg[i] - 2007,legStates[i]]+
    effDatefracs[i] * USAPOP[yearsLeg[i] - 2006,legStates[i]]

  #find point in state info, calculate averages
  index_Leg <- (as.numeric(legEffDate[i]) - 2008)*12 + 1
  pre_per_HT <- mean(stateInfo[(index_Leg - 12):(index_Leg - 1)])/pop_est_pre
  post_per_HT <-  mean(stateInfo[(index_Leg):(index_Leg + 12)])/pop_est_post
  avgs_pre <- append(avgs_pre,pre_per_HT)
  avgs_post <- append(avgs_post, post_per_HT)
}
hist(avgs_post - avgs_pre, breaks = 10,
     main = "Differences of Average Fatal Crash Number, Pre/Post Legalization",
     xlab = "Difference")
```



I generate the data-frame with data, and run a paired sample t-test:

```r
pairedData <- as.data.frame(cbind(legStates,avgs_pre,avgs_post))
colnames(pairedData) <- c("State", "avg_pre_Leg","avg_post_Leg")
pairedData$avg_pre_Leg <-round(as.numeric(pairedData$avg_pre_Leg),6)
pairedData$avg_post_Leg <- round(as.numeric(pairedData$avg_post_Leg),6)
```

```r
testMeans <- t.test(pairedData$avg_post_Leg, pairedData$avg_pre_Leg,
       paired = TRUE, alternative = "greater")
```

The test statistic is $t = \frac{m}{\frac{s}{\sqrt{n}}}$, where m is the mean difference, s is the sample standard deviation of the difference, and n is the number of observations (in this case, 19). I calculate:

```r
sprintf("Sample Mean Difference: %.5f", testMeans$estimate)
```

```
## [1] "Sample Mean Difference: 0.60166"
```

```r
sprintf("Test statstic: %.5f",testMeans$statistic["t"])
```

```
## [1] "Test statstic: 2.41257"
```

```r
sprintf("Degrees of Freedom: %.0f", testMeans$parameter)
```

```
## [1] "Degrees of Freedom: 18"
```

```r
sprintf("p-value: %.5f", testMeans$p.value)
```

```
## [1] "p-value: 0.01336"
```

```r
print("95% Confidence Interval:")
```

```
## [1] "95% Confidence Interval:"
```

```r
print(testMeans$conf.int[1:2])
```

```
## [1] 0.1692114       Inf
```

So, $\mathbb{P}(T_{18} > 2.41257) = 0.01336 < 0.05$; I reject H_0 at $\alpha = 0.05$.

c. Decision Tree and Random Forest to Predict Crashes

i. Collecting Data

Features to split on: (Note Rhode Island legalized on 05/22/2022 so it should probably be excluded)

- Marijuana Legal? (Pre-2022)
- Billions of Highway Miles-driven per 100,000 (2022)
- Proportion of Population in Urban Areas (2020)
- Speed Limits on Urban Interstates (as of 2024)
-
- Damage in Millions of Dollars per 100,000 by Hazardous Weather (2022)
-

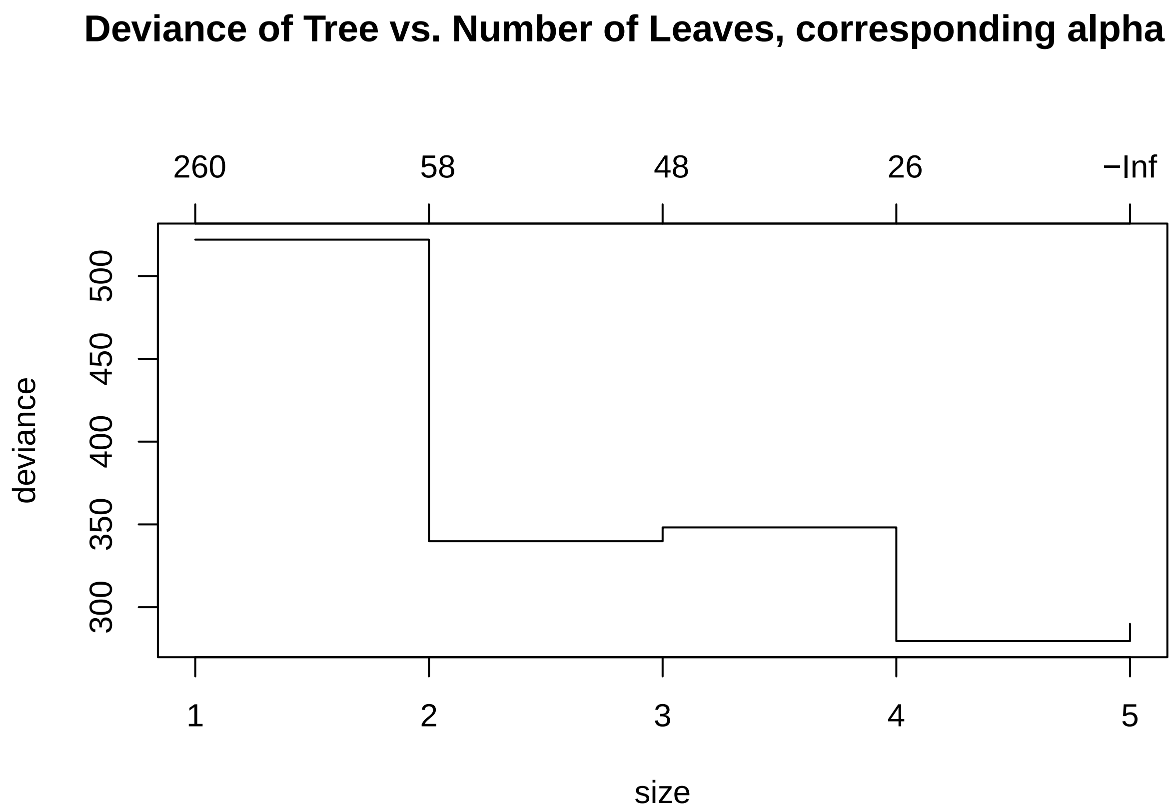
```r
#Create response variable:
avg2022 <- as.vector(t(as.matrix(read_excel("rfdata/2022StateData.xlsx",
    range = "B7:AZ8",col_names = TRUE,.name_repair = "unique_quiet"))))/12
pop2022 <- as.vector(t(as.matrix(USAPOP[2022 - 2007,2:dim(USAPOP)[2]])))
resp <- avg2022/pop2022

#Getting Percent Urban Population
urbRaw <- read_excel("rfdata/UrbanRural.xlsx", sheet = "Data", range = "B1:AZ4")
urb <- as.vector(t(urbRaw[2,]))/as.vector(t(urbRaw[1,]))

#Speed Limits on Urban Interstates
```

```r
spL <- c(70, 55, 65, 65, 65, 65, 55, 55, 55, 65, 70, 60, 75, 55, 55,
        55, 75, 65, 70, 75, 70, 65, 70, 65, 70, 60, 65, 70, 65, 65, 55, 75, 65,
        70, 75, 65, 70, 55, 70, 55, 70, 80, 70, 75, 70, 55, 70, 60, 55, 70, 75)

#Marijuana Legalized?
legMar <- as.integer(c(0,1,1,0,1,1,1,0,1,0,0,0,0,1,0,0,0,0,0,1,0,
            1,1,0,0,0,1,0,1,0,1,1,1,0,0,0,0,1,0,0,0,0,0,0,0,1,1,1,0,0,0))

#% Population Over 70 by State, 2022
over_70 <- as.numeric(read_excel("rfdata/USAAge70Over.xlsx")[1, ])

#Total Damage (in millions of $) from Hazardous Weather Events, 2022
hazard <- c(18.25, 30.97, 26.33, 44.48, 86.33, 1.1, 0.19, 0.26, 0, 17004.67,
            4.26, 1.33, 311.26, 25.70, 15.98, 24.97, 105.49, 8.22, 191.57,
            0.9, 7.84, 32.83, 75.24, 73.41, 183.68, 63.23, 4.27, 30.98,
            70.07, 443.28, 11, 0.17, 183.12, 10.59, 57.63, 20.17, 44.83,
            9.83, 8.08,1.12, 2.08, 499.83, 3.5, 1527.29, 36.76, 69.56,5.69,
            358.53, 11.48, 24.11, 25.17)
hazard <- hazard/pop2022

#Binge Drinking Prevalence among Adults
binge <- as.vector(t(as.matrix(read_excel("rfdata/BingeDrinking.xlsx",
                              sheet = "Data",range="C6:C57"))))/100

#Billions of Highway-Miles Driven per 100,000 people
hMiles <- t(as.matrix(read_excel("rfdata/hMiles.xlsx", range = "C1:C53")))
#remove Puerto Rico
hMiles <- hMiles[-c(40)]
#take billions of Miles per 100,000 people
hperHT <- (hMiles/1000)/pop2022

#Final Data Gathering
finData <- as.data.frame(cbind(resp,legMar,hperHT,urb,spL,over_70,hazard,binge))
corMat <- cor(finData)
finData$legMar <- factor(finData$legMar)
finData$spL <- factor(finData$spL)
finData$hperHT <- round(finData$hperHT, 6)
finData$urb <- round(finData$urb, 6)
finData$over_70 <- round(finData$over_70, 6)
finData$hazard <- round(finData$hazard, 6)
finData$states <- colnames(USAPOP)[2:52]
finData <- finData[,c(1,9,2:8)]
colnames(finData) <- c("resp", "State", "Legal","HW_travel",
                       "Urban", "Speed_Lim","over_70","hazard","binge")
finData <- finData[which(finData$State != "Rhode Island"),]
```

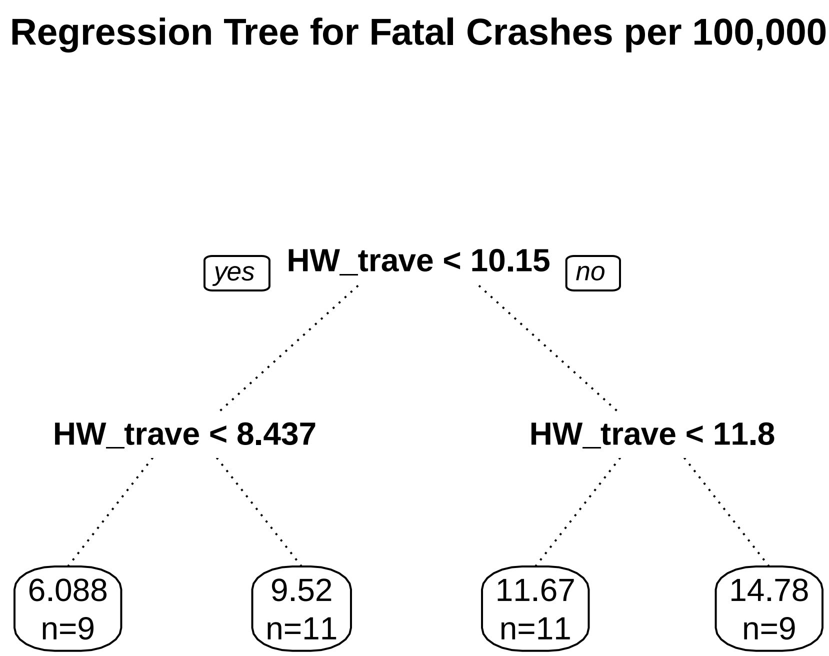
ii. 1 tree (for example)

```r
set.seed(1)
train_index <- sample(1:nrow(finData),.8 * nrow(finData),replace = FALSE)
test_index <- setdiff(1:50, train_index)
rtree <- tree(resp ~ ., data = finData[train_index,-c(2)])
cv <- cv.tree(rtree, K = 5, FUN = prune.tree)
```

```r
par(oma = c(0,0,2,0))
plot(cv)
title(main = "Deviance of Tree vs. Number of Leaves, corresponding alpha",
outer = TRUE)
```




```r
rtree <- rpart(resp ~., data = finData[train_index,-c(2)], method = "anova",
               control = rpart.control(cp = 1/20))
prp(rtree, main = "Regression Tree for Fatal Crashes per 100,000",
    roundint = FALSE, extra = 1, digits = 4, branch.lty = 3)
```



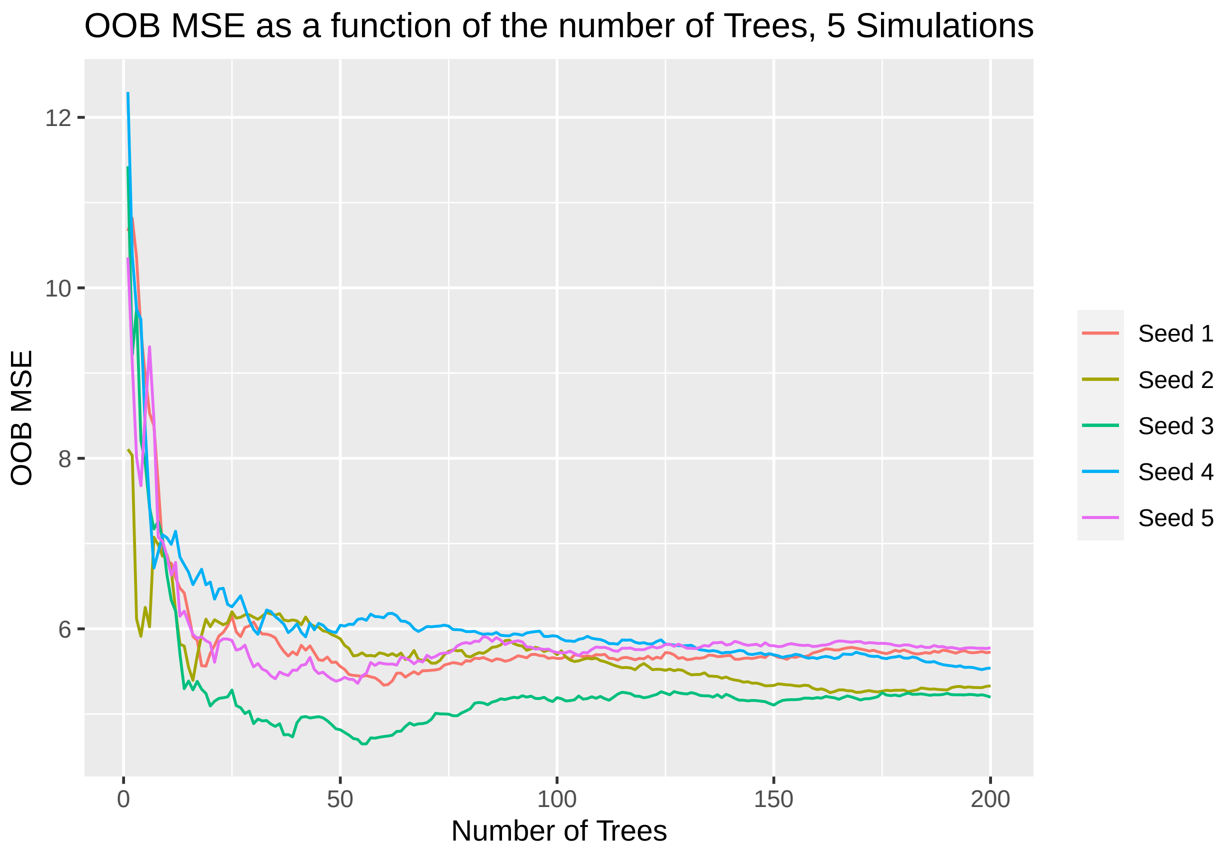

```
oneTree <- tree(resp ~ ., data = finData[train_index, -c(2)], method = "anova")
oneTree <- prune.tree(oneTree, k = 30)
predOneTree <- predict(oneTree, newdata = finData[test_index,3:9])
mseOneTree <- mean((predOneTree - finData$resp[test_index])^2)
pseudoROneTree <- 1 - (mseOneTree * 50)/(var(finData$resp) * 49)
```

iii. Bagging

I generate 5 different iterations of the random Forest for bagging in order to try and optimize for the number of trees:

```
num_tree = 200
mseRBs <- c()
for(j in 1:5){
  set.seed(j)
  rB_raw <- randomForest(resp ~., data = finData[,-c(2)], ntree = num_tree,
      mtry = dim(finData)[2] - 2, importance = TRUE, keep.inbag = TRUE)
  mseRBs <- append(mseRBs, rB_raw$mse)
}
mseRBs <- as.data.frame(matrix(mseRBs, nrow = num_tree, ncol = j))
colnames(mseRBs) <- c(1:5)
plotMSE <- ggplot(data = mseRBs, aes(x = 1:200)) +
  geom_line(aes(y = `1`, color = "Seed 1")) +
  geom_line(aes(y = `2`, color = "Seed 2")) +
  geom_line(aes(y = `3`, color = "Seed 3")) +
  geom_line(aes(y = `4`, color = "Seed 4")) +
  geom_line(aes(y = `5`, color = "Seed 5")) +
```

```r
  xlab("Number of Trees") + ylab("OOB MSE") +
  ggtitle("OOB MSE as a function of the number of Trees, 5 Simulations") +
  theme(legend.title = element_blank(), plot.title = element_text(hjust = 0.5))
plotMSE
```



It seems as though around 50 is where the OOB MSE finishes decreasing. For the sake of bias-variance trade-off, I select 50 as the number of trees, and fit a random forest model:

```r
set.seed(6)
RB <- randomForest(resp ~., data = finData[,-c(2)], ntree = 50,
      mtry = dim(finData)[2] - 2, importance = TRUE, keep.inbag = TRUE)
mseRB <- RB$mse[50]
pseudoRRB <- 1 - (mseRB * 50)/(var(finData$resp) * 49)
RB$importance
```

```
##                %IncMSE IncNodePurity
## Legal       0.04685631      1.713326
## HW_travel 11.59499367    426.179536
## Urban      -0.19137532     24.914778
## Speed_Lim -0.21115978     27.582473
## over_70    -0.33457491     31.267251
## hazard     -0.01103727     28.871072
## binge       0.34190161     34.615816
```

iv. Random Forest

I generate summary statistics:

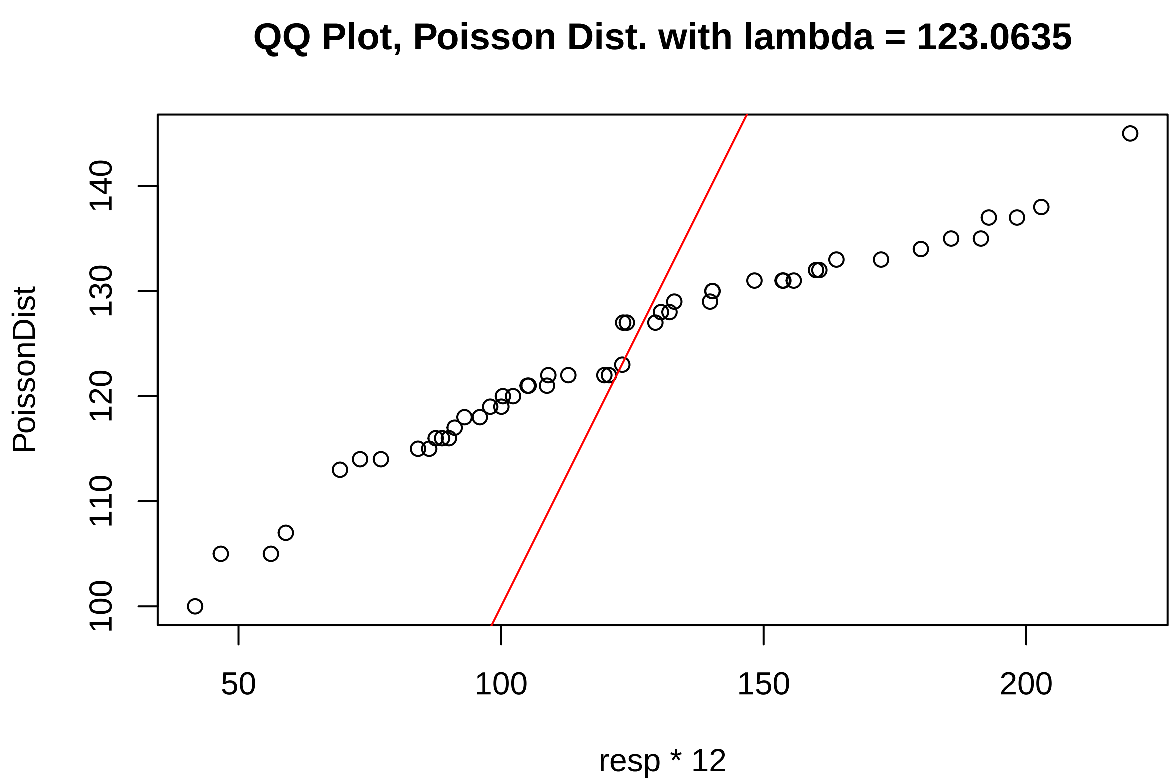
```r
sumstats <- as.data.frame(cbind(c(mseOneTree, mseRB),
                                c(pseudoROneTree, pseudoRRB)))
colnames(sumstats) <- c("MSE", "Pseudo Rsq.")
sumstats$Algorithm <- c("Decision Tree", "Bagged Forest")
sumstats <- sumstats[,c(3,1,2)]
sumstats
```

```
##       Algorithm      MSE Pseudo Rsq.
## 1 Decision Tree 6.605976   0.4375485
## 2 Bagged Forest 5.609170   0.5224193
```

d. Regression Analysis

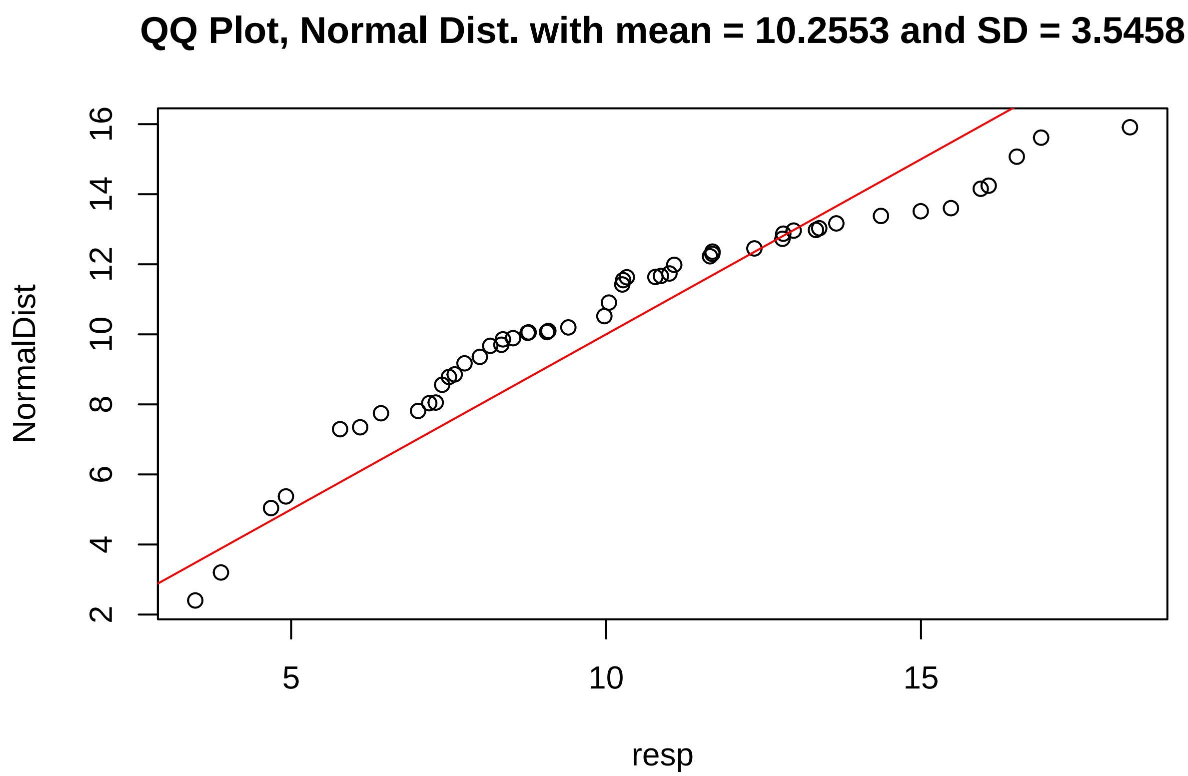
i. Testing distribution of Response

```r
set.seed(1)
lambdaMLE <- mean(resp * 12)
#hist(resp, breaks = 20)
#qqplot(resp, distribution = "poisson")
PoissonDist <- rpois(length(resp), lambdaMLE)
title <- sprintf("QQ Plot, Poisson Dist. with lambda = %.4f",lambdaMLE)
qqplot(resp * 12, PoissonDist, main = title)
abline(0,1,col = 'red')
```

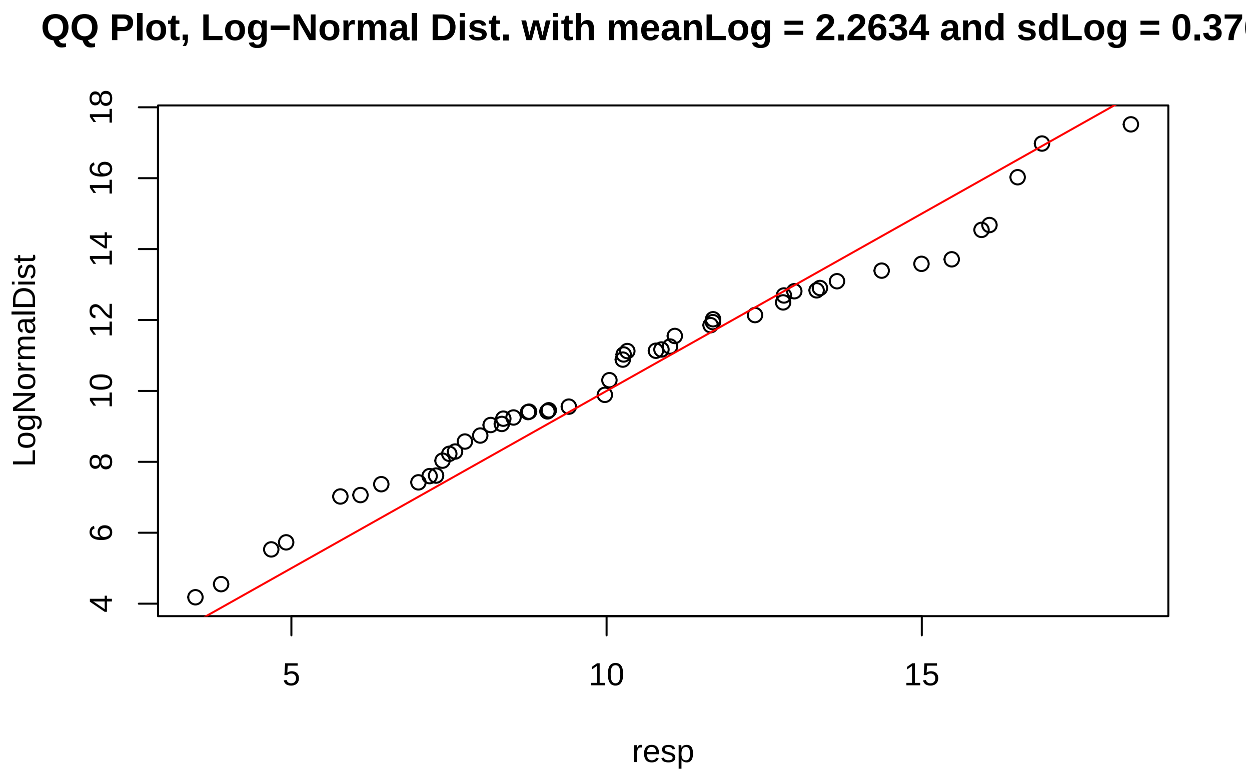



```r
set.seed(1)
meanResp <- mean(resp)
```

```
sdResp <- sd(resp)
NormalDist <- rnorm(length(resp), meanResp, sdResp)
title <- sprintf("QQ Plot, Normal Dist. with mean = %.4f and SD = %.4f", meanResp, sdResp)
qqplot(resp, NormalDist, main = title)
abline(0, 1, col = 'red')
```




```
set.seed(1)
meanLogResp <- mean(log(resp))
sdLogResp <- sd(log(resp))
LogNormalDist <- rlnorm(length(resp), meanLogResp, sdLogResp)
title <- sprintf("QQ Plot, Log-Normal Dist. with meanLog = %.4f and sdLog = %.4f", meanLogResp, sdLogRes
qqplot(resp, LogNormalDist, main = title)
abline(0, 1, col = 'red')
```



Actual Linear Modeling

```r
linMod <- lm("log(resp) ~ Speed_Lim",
             data = finData)
summary(linMod)
```

```
##
## Call:
## lm(formula = "log(resp) ~ Speed_Lim", data = finData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.86126 -0.19085 -0.01437  0.30527  0.53962
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.10762    0.10562  19.955   <2e-16 ***
## Speed_Lim60  0.03396    0.22816   0.149   0.8824
## Speed_Lim65  0.12207    0.14351   0.851   0.3996
## Speed_Lim70  0.28785    0.13905   2.070   0.0443 *
## Speed_Lim75  0.34148    0.16936   2.016   0.0499 *
## Speed_Lim80  0.29771    0.36587   0.814   0.4202
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3503 on 44 degrees of freedom
```

```
## Multiple R-squared:  0.1337, Adjusted R-squared:  0.03521
## F-statistic: 1.358 on 5 and 44 DF,  p-value: 0.2586
```

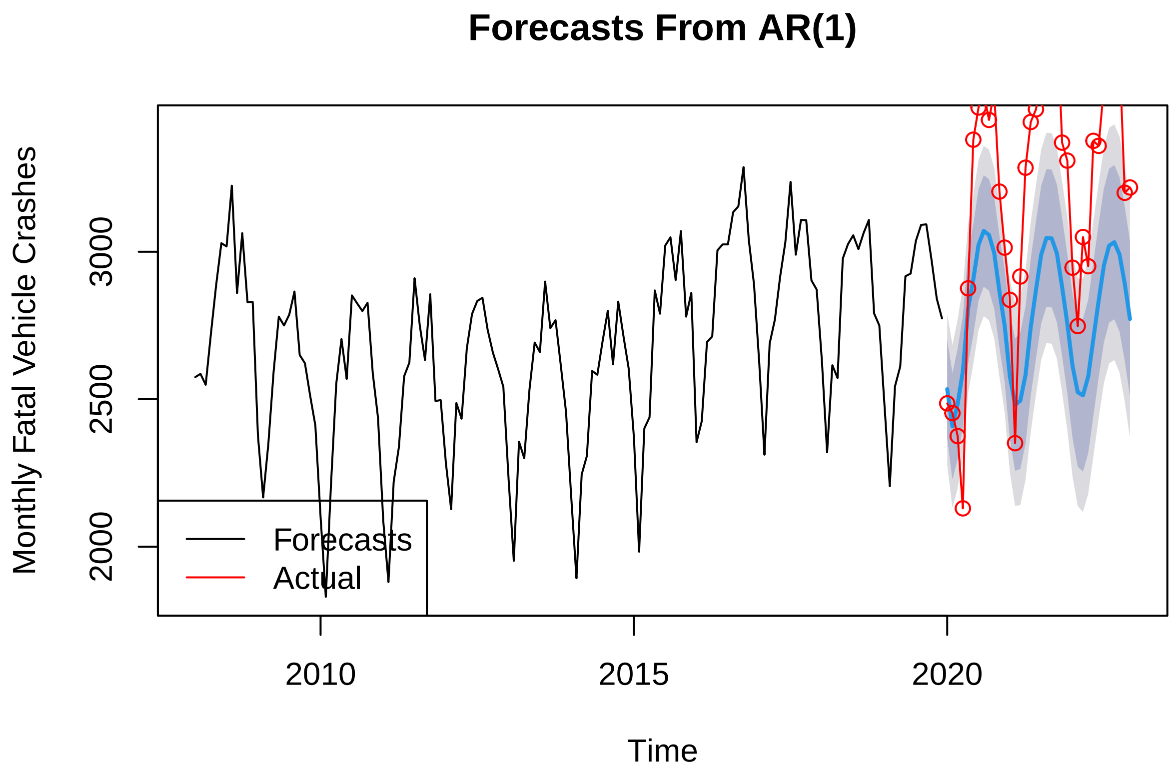
Auto Regression Code below

```r
# Splitting data set into first 80% as train set and last 20% as test set
train_set <- ts(USAFat[1:floor(0.8 * length(USAFat))], start = 2008, freq = 12)
test_set <- ts(USAFat[(floor(0.8 * length(USAFat)) + 1):length(USAFat)], start = 2020, freq = 12)
```

```r
ar1_model <- ar(train_set, method = "mle")
summary(ar1_model)
```

```
##              Length Class  Mode
## order          1    -none- numeric
## ar            12    -none- numeric
## var.pred       1    -none- numeric
## x.mean         1    -none- numeric
## aic           13    -none- numeric
## n.used         1    -none- numeric
## n.obs          1    -none- numeric
## order.max      1    -none- numeric
## partialacf     0    -none- NULL
## resid        144    ts     numeric
## method         1    -none- character
## series         1    -none- character
## frequency      1    -none- numeric
## call           3    -none- call
## asy.var.coef 144    -none- numeric
```

```r
predictions <- forecast(ar1_model, h = length(test_set))
plot(predictions, main = "Forecasts From AR(1)", xlab = "Time", ylab = "Monthly Fatal Vehicle Crashes")
lines(test_set, col = "red", type = "o")
legend("bottomleft", legend = c("Forecasts", "Actual"), col = c("black", "red"), lty = 1)
```


Forecasts From AR(1)
Monthly Fatal Vehicle Crashes
2000
2500
3000
2010
2015
2020
Time
Forecasts
Actual