final\_project

Jessica Woods

2023-05-24

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.1 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.2 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

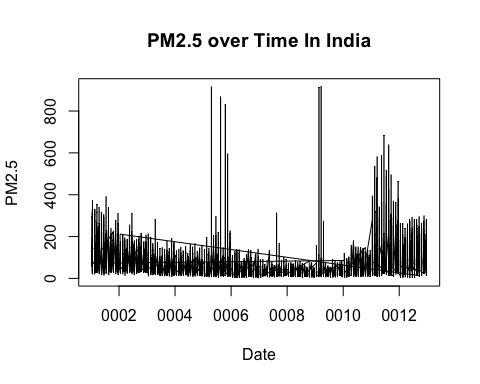
library(ggplot2)  
library(HistData)  
library(dplyr)

## R Markdown

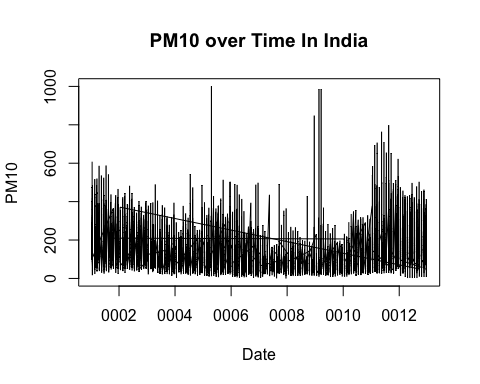
This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

#import data to clean, which is from India csv file collected from Kaggle  
# This data includes monthly reports about the air quality  
india\_air\_data<-read.csv("city\_day.csv")  
# this data is five years of data 2015-2020  
# Don't care about AQI, it is a generalization that may or may not be accurate, we don't want anything in the data that tells you what to think-  
  
clean\_data\_india\_2.5<-na.omit(india\_air\_data$PM2.5)  
clean\_data\_india\_10.0<-na.omit(india\_air\_data$PM10)  
  
# Specify the columns to remove NA values from  
columns <- c("PM2.5", "PM10")  
# Create a logical vector indicating complete cases in specified columns  
complete\_rows <- complete.cases(india\_air\_data[, columns])  
clean\_data <- subset(india\_air\_data, complete\_rows)  
#check to see what the outliers are for further analysis  
# Arrange data in ascending order by date  
date\_ordered <- clean\_data[order(clean\_data$Date), ]  
  
date\_ordered$Date <- as.Date(date\_ordered$Date)  
y <- date\_ordered$PM2.5  
x <- date\_ordered$Date  
y\_11 <- date\_ordered$PM10  
  
# so I want to visually look for possible outliers in the data for further cleaning   
india\_pm2.5\_graph<-plot(x, y, type = "l", xlab = "Date", ylab = "PM2.5", main = "PM2.5 over Time In India")



india\_pm10\_graph<-plot(x, y\_11, type = "l", xlab = "Date", ylab = "PM10", main = "PM10 over Time In India")



## Including Plots

Lets do China next:

# combine and look at china as a whole like I did India before we break them   
China\_data <- data.frame()  
csv\_files<-c("PRSA\_Data\_Dongsi\_20130301-20170228.csv","PRSA\_Data\_Aotizhongxin\_20130301-20170228.csv", "PRSA\_Data\_Changping\_20130301-20170228.csv", "PRSA\_Data\_Dingling\_20130301-20170228.csv", "PRSA\_Data\_Guanyuan\_20130301-20170228.csv", "PRSA\_Data\_Gucheng\_20130301-20170228.csv", "PRSA\_Data\_Huairou\_20130301-20170228.csv", "PRSA\_Data\_Nongzhanguan\_20130301-20170228.csv", "PRSA\_Data\_Shunyi\_20130301-20170228.csv", "PRSA\_Data\_Tiantan\_20130301-20170228.csv","PRSA\_Data\_Wanliu\_20130301-20170228.csv", "PRSA\_Data\_Wanshouxigong\_20130301-20170228.csv")

This cleaning will take a bit more work as the data from Kaggle is a mess and we want the China data in a similar format as the India data to create a scale, the data should intersect at 2015, where China starts early at 2013-2017 and India goes from 2015 -2020. The two year intersection is important analyse. Both of these countries are always tormented by environmentalist, so I want to see for myself what it looks like. This analysis is not to prove a point or a argument, its to learn about the factors involved that can be disputed as nonsense.

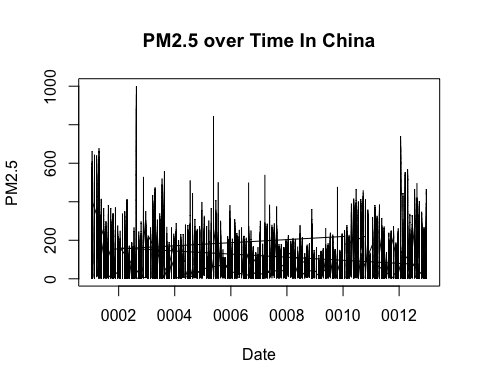
# Iterate over each CSV file  
for (file in csv\_files) {  
 data <- read.csv(file)  
 China\_data <- rbind(China\_data, data)  
}  
# fix the date  
China\_data$Date <- paste(China\_data$month, China\_data$day, China\_data$year, sep = "/")

Lets clean up the dates we have too much data…Do not try to rerun the top chunk after you run this one because it won’t work as you need those …hmmm- lets change variables for safe reasons

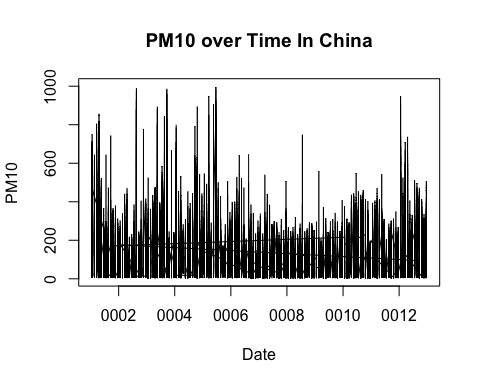
#remove some columns from this  
China\_data\_clean = China\_data[ -c(1:5) ]  
clean\_data\_China\_2.5<-na.omit(China\_data\_clean$PM2.5)  
clean\_data\_China\_10.0<-na.omit(China\_data\_clean$PM10)

Now lets take a look at China’s air pollution over time like we did India as a whole.

date\_ordered\_china <- China\_data\_clean[order(China\_data\_clean$Date), ]  
  
date\_ordered\_china$Date <- as.Date(date\_ordered\_china$Date)  
y\_2 <- date\_ordered\_china$PM2.5  
x1 <- date\_ordered\_china$Date  
y\_10 <- date\_ordered\_china$PM10  
  
# so I want to visually look for possible outliers in the data for further cleaning   
graph\_china\_pm2.5<-plot(x1, y\_2, type = "l", xlab = "Date", ylab = "PM2.5", main = "PM2.5 over Time In China")



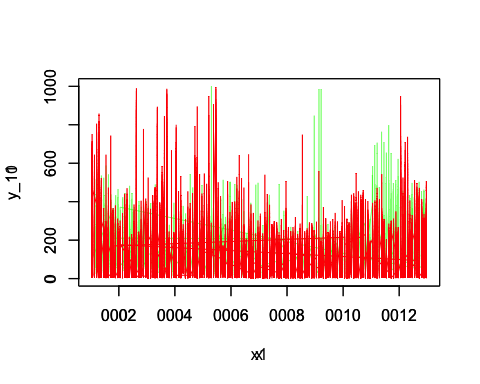
graph\_china\_pm10<-plot(x1, y\_10, type = "l", xlab = "Date", ylab = "PM10", main = "PM10 over Time In China")

 transparent colors I found online which is cool and needed-

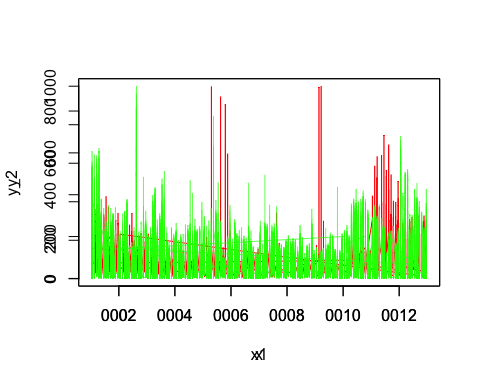
## Transparent colors  
## Mark Gardener 2015  
## www.dataanalytics.org.uk  
  
t\_col <- function(color, percent = 50, name = NULL) {  
 # color = color name  
 # percent = % transparency  
 # name = an optional name for the color  
  
## Get RGB values for named color  
rgb.val <- col2rgb(color)  
  
## Make new color using input color as base and alpha set by transparency  
t.col <- rgb(rgb.val[1], rgb.val[2], rgb.val[3],  
 max = 255,  
 alpha = (100 - percent) \* 255 / 100,  
 names = name)  
  
## Save the color  
invisible(t.col)  
}  
## END

Compare graphs to see the differences this is PM10 were the graphs are paired together over a 5 year span of the collected data, even though they are different dates, because I classified the date, it can easily pair them together with par() function. Interesting part of this is, is that the economic structures that affect the air are different in each country. India is a communal country more so than industrial, as China is largely industrial. Red = India Green = China

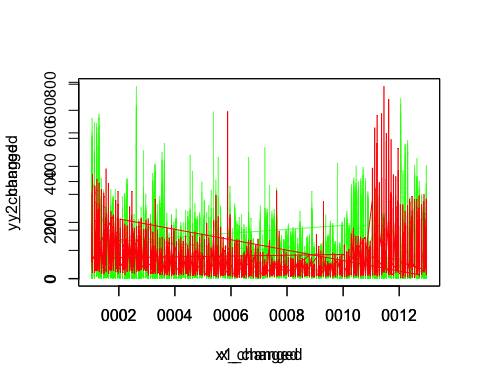
mycol <- t\_col("green", perc = 40, name = "lt.green")  
plot(x,y\_11,type="l",col=mycol)  
par(new=TRUE)  
  
#China data compared to India's data over 5 year span.  
plot(x1,y\_10,type="l",col="red")

 Next we will look at PM2.5 particles that are more from seated into pollution.

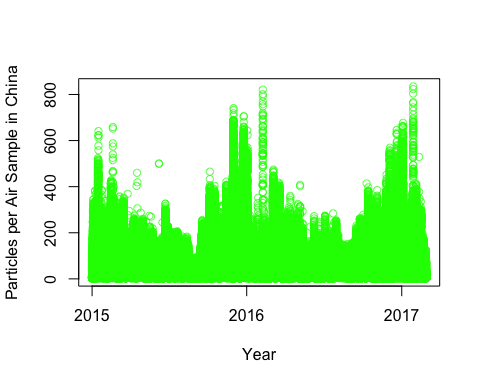
y\_range = c(0:1000)  
  
plot(x,y,type="l", col="red")  
par(new=TRUE)  
# china data  
mycol <- t\_col("green", perc = 30, name = "lt.green")  
plot(x1, y\_2,type= "l", col=mycol)



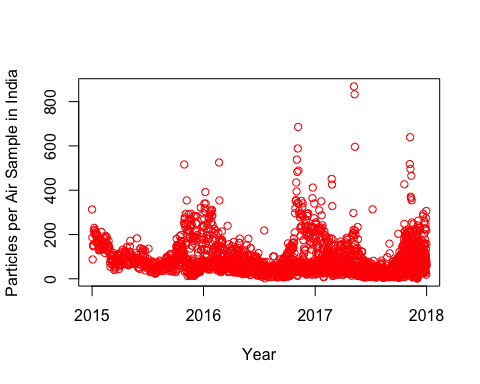
# I have to clean China's data even more to compare as China goes to 1000 in PM2.5  
# it needs to be on some equal grounds to fully compare the graphs as you can see the first graph is way off and the second one is more equally compared.  
  
date\_ordered\_china <- date\_ordered\_china[which(date\_ordered\_china$PM2.5 <= 800),]  
  
# this should give a better comparison  
date\_ordered\_china$Date <- as.Date(date\_ordered\_china$Date)  
y\_2\_changed <- date\_ordered\_china$PM2.5  
x1\_changed <- date\_ordered\_china$Date  
  
# I had to do the same thing with India...I selected all data in the PM2.5 values below or equal to 800  
date\_ordered\_1 <- clean\_data[order(clean\_data$Date), ]  
  
date\_ordered\_1$Date <- as.Date(date\_ordered\_1$Date)  
  
date\_ordered\_1 <- date\_ordered\_1[which(date\_ordered\_1$PM2.5 <= 800),]  
y\_changed <- date\_ordered\_1$PM2.5  
x\_changed <- date\_ordered\_1$Date  
  
# these are not the pretty graphs, I was simply mashing them together to see if there was a significant difference  
# between China and India before I started analyzing the data.  
plot(x1\_changed, y\_2\_changed,type= "l", col=mycol)  
par(new=TRUE)  
plot(x\_changed, y\_changed, type="l", col="red")

 With these visuals we can easily say industrialism has a greater impact on the air quality than communial of high populations. In the PM2.5 of India, you can surmise that the monsoon season between April and September is more impacted than the monsoon seasons are in China. We can look at the means of China per year and compare it to the means of India per year. Which would conclude that theory.

dates\_as\_date<-as.Date(China\_data\_clean$Date, format = "%m/%d/%Y")  
#my dates were still messed up so I had to change them more in China data-  
dates\_as\_date\_i<-as.Date(india\_air\_data$Date, format = "%m/%d/%y")  
  
China\_data\_clean$Date\_as\_Date <- dates\_as\_date  
india\_air\_data$Date\_as\_Date<- dates\_as\_date\_i  
# Extract the dates and PM2.5 values  
# Filter for the years 2015, 2016, and 2017  
filtered\_data\_china\_2015\_17 <- subset(China\_data\_clean, format(Date\_as\_Date, "%Y") %in% c("2015", "2016", "2017"))  
filtered\_data\_india\_2015\_17 <- subset(india\_air\_data, format(Date\_as\_Date, "%Y") %in% c("2015", "2016", "2017"))  
# remove na's from India's data agian...  
filtered\_data\_india\_2015\_17<-filtered\_data\_india\_2015\_17[complete.cases(filtered\_data\_india\_2015\_17$PM2.5, filtered\_data\_india\_2015\_17$PM10), ]  
filtered\_data\_china\_2015\_17<-filtered\_data\_china\_2015\_17[complete.cases(filtered\_data\_china\_2015\_17$PM2.5, filtered\_data\_china\_2015\_17$PM10), ]  
df\_india <- data.frame(filtered\_data\_india\_2015\_17)  
df\_china <- data.frame(filtered\_data\_china\_2015\_17)  
  
# Plotting PM2.5 for India and China together  
y\_changed <- df\_india$PM2.5  
x\_changed <- df\_india$Date\_as\_Date  
y\_china <- df\_china$PM2.5  
  
x\_china<- df\_china$Date\_as\_Date  
  
plot(x\_china, y\_china, col= mycol, ylab="Particles per Air Sample in China", xlab="Year")



plot(x\_changed, y\_changed, col="red", ylab="Particles per Air Sample in India", xlab = "Year")



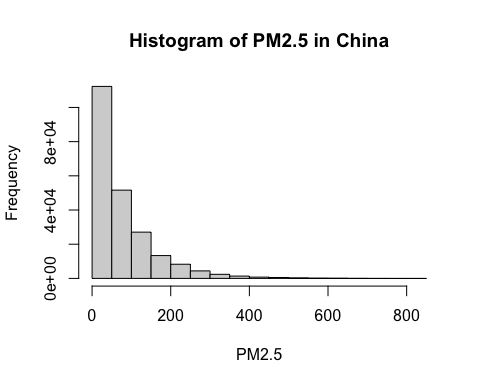
mean(df\_china$PM2.5) #mean value is 76.95093

## [1] 76.95093

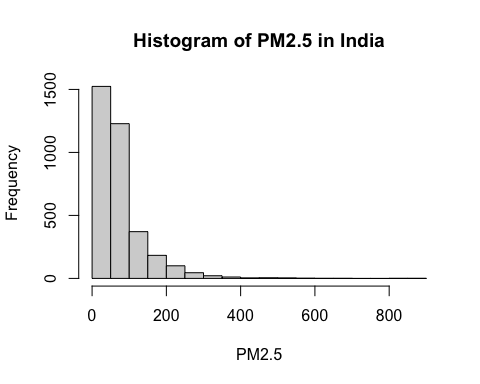
mean(df\_india$PM2.5) #mean value is 74.9217

## [1] 74.9217

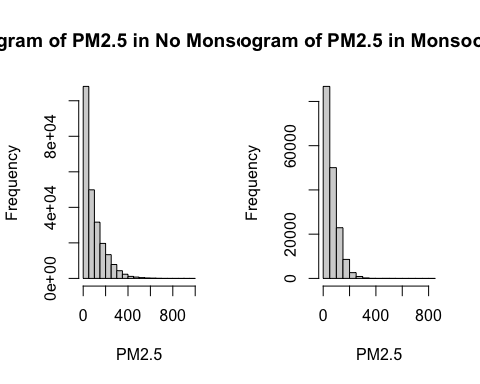
hist(y\_china, main = "Histogram of PM2.5 in China", xlab = "PM2.5", ylab = "Frequency")



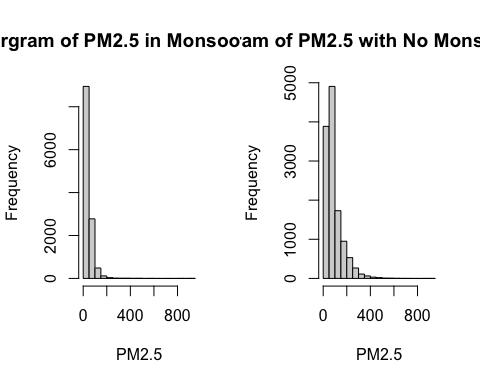
hist(y\_changed, main = "Histogram of PM2.5 in India", xlab = "PM2.5", ylab = "Frequency")

 I am going to study the data and determine if the select data is parametric or not. What is my null hypothesis and alpha level. “H0: There is a significant difference in PM2.5 levels from monsoon season in China and India.”

filtered\_data\_china\_May\_Sept <- subset(China\_data\_clean, format(Date\_as\_Date, "%m") %in% c("05", "06", "07","08","09"))  
filtered\_data\_india\_April\_Sept <- subset(india\_air\_data, format(Date\_as\_Date, "%m") %in% c("04", "05", "06", "07","08","09"))  
# I want all the months outside of monsoon weather to compare the data, China has one less month of Monsoon weather  
filtered\_data\_china\_no\_monsoon <- subset(China\_data\_clean, format(Date\_as\_Date, "%m") %in% c("01", "02","03","04","10","11","12"))  
filtered\_data\_india\_no\_monsoon <- subset(india\_air\_data, format(Date\_as\_Date, "%m") %in% c("01", "02","03","10","11","12"))  
#monsoon variables  
monsoon\_y\_china = filtered\_data\_china\_May\_Sept$PM2.5  
monsoon\_x\_china = filtered\_data\_china\_May\_Sept$Date\_as\_Date  
# no monsoon variables  
monsoon\_y\_china <- na.omit(monsoon\_y\_china)  
no\_monsoon\_y\_china = filtered\_data\_china\_no\_monsoon$PM2.5  
# need to rerun because I think somehow it has NA values  
# somehow LOL- na's invaded my data-  
no\_monsoon\_x\_china = filtered\_data\_china\_no\_monsoon$Date\_as\_Date  
no\_monsoon\_y\_china <- na.omit(no\_monsoon\_y\_china)  
# I am putting the histograms next to each other  
par(mfrow = c(1, 2))  
hist(no\_monsoon\_y\_china, main = "Histogram of PM2.5 in No Monsoon China", xlab = "PM2.5", ylab = "Frequency" )  
hist(monsoon\_y\_china,main = "Histogram of PM2.5 in Monsoon China", xlab = "PM2.5", ylab = "Frequency")



#india monsoon variables and no monsoon variables  
monsoon\_y\_india = filtered\_data\_india\_April\_Sept$PM2.5  
monsoon\_y\_india <- na.omit(monsoon\_y\_india)  
  
monsoon\_x\_india = filtered\_data\_india\_April\_Sept$Date\_as\_Date  
no\_monsoon\_y\_india = filtered\_data\_india\_no\_monsoon$PM2.5  
no\_monsoon\_y\_india <- na.omit(no\_monsoon\_y\_india)  
  
no\_monsoon\_x\_india = filtered\_data\_india\_no\_monsoon$Date\_as\_Date  
hist(monsoon\_y\_india, main= "Historgram of PM2.5 in Monsoon of India", xlab = "PM2.5", ylab = "Frequency")  
hist(no\_monsoon\_y\_india, main= "Historgram of PM2.5 with No Monsoon in India", xlab = "PM2.5", ylab = "Frequency")



mean(no\_monsoon\_y\_china) #mean =91.31508

## [1] 91.31508

mean(monsoon\_y\_china) # mean = 63.73192

## [1] 63.73192

mean(no\_monsoon\_y\_india) # mean = 91.00855

## [1] 91.00855

mean(monsoon\_y\_india) # mean = 43.70478

## [1] 43.70478

sd(no\_monsoon\_y\_china) #standard dev = 94.46

## [1] 94.46144

sd(monsoon\_y\_china) # standard dev = 52.50

## [1] 52.50287

sd(no\_monsoon\_y\_india) #standard dev = 72.79

## [1] 72.79203

sd(monsoon\_y\_india) #standard dev = 43.95

## [1] 43.94994

Because we can see in the histogram that this might not be normally distributed I will try the Wilcoxon Rank Test.

# Perform Wilcoxon rank-sum test  
wilcox\_test <- wilcox.test(no\_monsoon\_y\_china, monsoon\_y\_china)  
  
# Print the results  
print(wilcox\_test)

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: no\_monsoon\_y\_china and monsoon\_y\_china  
## W = 2.2487e+10, p-value < 2.2e-16  
## alternative hypothesis: true location shift is not equal to 0

# The P-Value 2.2e-16 supports the hypothesis that a monsoon strongly changes the results of PM2.5 values in China.

Now to do a Wilcoxon Test on the India Monsoon data.

# Perform Wilcoxon rank-sum test  
wilcox\_test\_india <- wilcox.test(no\_monsoon\_y\_india, monsoon\_y\_india)  
  
# Print the results  
print(wilcox\_test\_india)

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: no\_monsoon\_y\_india and monsoon\_y\_india  
## W = 120059221, p-value < 2.2e-16  
## alternative hypothesis: true location shift is not equal to 0

# The P-Value 2.2e-16 supports the hypothesis that a monsoon strongly changes the results of PM2.5 values in India.

Next I will perform a sign test, since we know this non parametric data.

# Perform sign test  
result\_of\_sign\_china <- binom.test(sum(no\_monsoon\_y\_china > monsoon\_y\_china), n = length(no\_monsoon\_y\_china))

## Warning in no\_monsoon\_y\_china > monsoon\_y\_china: longer object length is not a  
## multiple of shorter object length

result\_of\_sign\_india<- binom.test(sum(no\_monsoon\_y\_india > monsoon\_y\_india), n = length(no\_monsoon\_y\_india))

## Warning in no\_monsoon\_y\_india > monsoon\_y\_india: longer object length is not a  
## multiple of shorter object length

# Print the results  
print(result\_of\_sign\_china)

##   
## Exact binomial test  
##   
## data: sum(no\_monsoon\_y\_china > monsoon\_y\_china) and length(no\_monsoon\_y\_china)  
## number of successes = 130293, number of trials = 239922, p-value <  
## 2.2e-16  
## alternative hypothesis: true probability of success is not equal to 0.5  
## 95 percent confidence interval:  
## 0.5410681 0.5450588  
## sample estimates:  
## probability of success   
## 0.543064

print(result\_of\_sign\_india)

##   
## Exact binomial test  
##   
## data: sum(no\_monsoon\_y\_india > monsoon\_y\_india) and length(no\_monsoon\_y\_india)  
## number of successes = 10082, number of trials = 12516, p-value <  
## 2.2e-16  
## alternative hypothesis: true probability of success is not equal to 0.5  
## 95 percent confidence interval:  
## 0.7984843 0.8124312  
## sample estimates:  
## probability of success   
## 0.8055289

To confirm this we will perform a chiX2 test on the data.

#so it blew up because I need to take a sample to make them the same length of data  
sample\_no\_monsoon\_china <- sample(no\_monsoon\_y\_china, 10000)  
sample\_monsoon\_china <-sample(monsoon\_y\_china, 10000)  
sample\_no\_monsoon\_india<-sample(no\_monsoon\_y\_india, 10000)  
sample\_monsoon\_india<- sample(monsoon\_y\_india, 10000)  
# Create a contingency table of the observed frequencies  
table\_china <- table(sample\_no\_monsoon\_china, sample\_monsoon\_china)  
table\_india <- table(sample\_no\_monsoon\_india, sample\_monsoon\_india)  
# Perform the chi-squared test  
chi\_sq\_test\_china <- chisq.test(table\_china)

## Warning in chisq.test(table\_china): Chi-squared approximation may be incorrect

chi\_sq\_test\_india <- chisq.test(table\_india)

## Warning in chisq.test(table\_india): Chi-squared approximation may be incorrect

# Print the results  
print(chi\_sq\_test\_china)

##   
## Pearson's Chi-squared test  
##   
## data: table\_china  
## X-squared = 154093, df = 145985, p-value < 2.2e-16

print(chi\_sq\_test\_india)

##   
## Pearson's Chi-squared test  
##   
## data: table\_india  
## X-squared = 42732124, df = 42738936, p-value = 0.7694

Lets also check if we remove a monsoon month from the data, in India and compare it to China, to see if industrial vr communal has any impact on the air quality in comparison to monsoons.

filtered\_data\_india\_monsoon\_no\_april <-subset(india\_air\_data, format(Date\_as\_Date, "%m") %in% c("05", "06", "07","08","09"))  
# Create a copy of the "10" month data to simulate a similar monsoon season as China  
september\_data <- subset(india\_air\_data, format(Date\_as\_Date, "%m") == "10")  
filtered\_data\_india\_no\_monsoon\_with\_April <- filtered\_data\_india\_no\_monsoon  
# Add the copy to the existing data frame  
filtered\_data\_india\_no\_monsoon\_with\_April <- rbind(filtered\_data\_india\_no\_monsoon\_with\_April, september\_data)  
simulation\_of\_china\_monsoon\_in\_india <- filtered\_data\_india\_monsoon\_no\_april$PM2.5  
#drop NA's they are the worst bugs in this mess  
simulation\_of\_china\_monsoon\_in\_india<- na.omit(simulation\_of\_china\_monsoon\_in\_india)  
simulation\_of\_china\_nomonsoon\_in\_india <-filtered\_data\_india\_no\_monsoon\_with\_April$PM2.5  
simulation\_of\_china\_nomonsoon\_in\_india <- na.omit(simulation\_of\_china\_nomonsoon\_in\_india)  
  
# lets compare the mean and standard deviation of a simulated India with China's monsoon months  
mean(simulation\_of\_china\_nomonsoon\_in\_india) # mean is 88.82

## [1] 88.82384

mean(simulation\_of\_china\_monsoon\_in\_india) # mean is 41.65

## [1] 41.65911

mean(no\_monsoon\_y\_china) #mean =91.31508

## [1] 91.31508

mean(monsoon\_y\_china) # mean = 63.73192

## [1] 63.73192

sd(simulation\_of\_china\_nomonsoon\_in\_india) #standard deviation 71.13

## [1] 71.12907

sd(simulation\_of\_china\_monsoon\_in\_india) #standard deviation 44.97

## [1] 44.97175

sd(no\_monsoon\_y\_china) #standard dev = 94.46

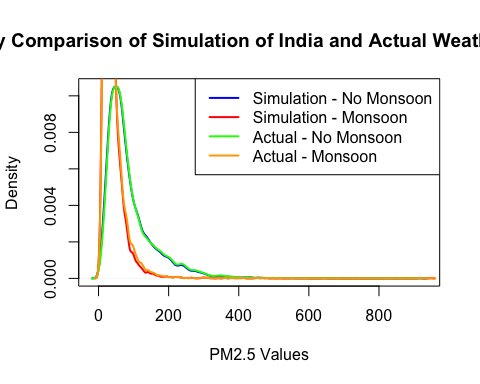
## [1] 94.46144

sd(monsoon\_y\_china) # standard dev = 52.50

## [1] 52.50287

We can plot a density curve also comparing these results.

# Plotting density curves  
plot(density(simulation\_of\_china\_nomonsoon\_in\_india),   
 col = "blue",   
 lwd = 2,   
 main = "Density Comparison of Simulation of India and Actual Weather of India",  
 xlab = "PM2.5 Values",  
 ylab = "Density")  
  
lines(density(simulation\_of\_china\_monsoon\_in\_india),   
 col = "red",   
 lwd = 2)  
  
lines(density(no\_monsoon\_y\_india),  
 col = "green",  
 lwd = 2)  
  
lines(density(monsoon\_y\_india),  
 col = "orange",  
 lwd = 2)  
  
# Adding a legend  
legend("topright",   
 legend = c("Simulation - No Monsoon", "Simulation - Monsoon", "Actual - No Monsoon", "Actual - Monsoon"),  
 col = c("blue", "red", "green", "orange"),  
 lwd = 2)

 Interesting the mean values are the same roughly over the same period of time, so I am going to do further analysis to just make requirements for the class, because I am running out of time. I am going to create a density plot and compare data with.

#install.packages("forecast")  
#library(forecast)  
# Extract the 'PM2.5' column as a numeric vector  
#my\_data <- as.numeric(df\_india$PM2.5)  
#start\_date <- as.Date("2015-01-01")  
# Create a time series object  
#my\_ts <- ts(my\_data, start = start\_date, frequency = 12) # Adjust start date and frequency as needed  
  
# Plot the time series  
#plot(my\_ts, main = "My Time Series Data", xlab = "Date", ylab = "Value")  
  
# Fit an ARIMA model  
#arima\_model <- auto.arima(my\_ts)  
#forecast\_values <- forecast(arima\_model, h = 12) # Generate 12-step-ahead forecasts  
#plot(forecast\_values, main = "Forecasted Values")