Assignment – statistical analysis

By David Elks

Note: The following report is an RStudio Notebook representation of the data analysis required for the assignment. The narrative aims to provide an explanation of the various steps taken, while the code used to create the charts and tables has been included where appropriate.

A key step is loading the Tidyverse library, a project by Hadley Wickham of RStudio to handle data in a ‘tidy’ way. The library includes dplyr, readr and ggplot2. Lubridate is loaded to make handling dates simpler.

library(tidyverse)

library(lubridate

1. **Preliminary checks on the dataset including examining data-types, looking for missing values or values that are out-of-range.**

What is the size of the dataset?

dim(df)

## [1] 9124268 24

Examine the summary, structure head and tail of the document. This reveals that the Uncertainty column is full of NA data while the Qualifier column is sparsely populated. The columns do match up with the supplied meta data on the www.epa.gov.uk website.

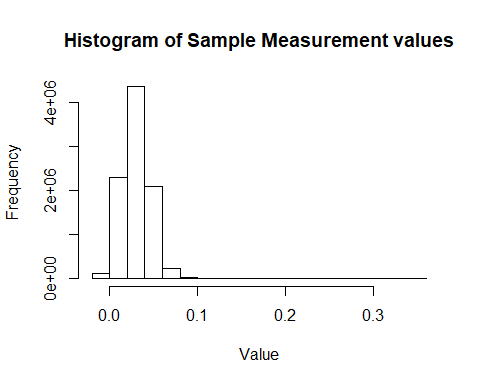
summary(df)

…

## Units.of.Measure MDL Uncertainty   
## Parts per million:9124268 Min. :0.0006 Mode:logical   
## 1st Qu.:0.0050 NA's:9124268   
## Median :0.0050   
## Mean :0.1383   
## 3rd Qu.:0.0050   
## Max. :8.0000   
##   
## Qualifier Method.Type Method.Code   
## :8927125 FEM :9108726 Min. : 19.00   
## SX : 100350 Non-FRM: 15542 1st Qu.: 47.00   
## 3 : 34533 Median : 87.00   
## 2 : 22694 Mean : 72.35   
## QX : 16021 3rd Qu.: 87.00   
## IT : 7405 Max. :901.00   
## (Other): 16140

One concern with the measurement data is that there are 2,949 negative values for ozone measurement (see histogram below), and need to be removed. They only represent 0.03 per cent of the dataset. The histogram below shows the distribution of the data (and the negative-valued data on the far left).

hist(df$Sample.Measurement, main = "Histogram of Sample Measurement values",  
 xlab = "Value", ylab="Frequency")



There is a total of 2,949 measurements with sample values of less than zero.

df %>%   
 filter(Sample.Measurement < 0) %>%   
 count()

## # A tibble: 1 × 1  
## n  
## <int>  
## 1 2949

As these are invalid – it is not possible to have a negative amount of ozone – we remove these negative measurements from the dataset.

df <- df %>%   
 filter(Sample.Measurement >= 0)

Let's look at the structure of the data.

str(df)

## 'data.frame': 9121319 obs. of 24 variables:  
## $ State.Code : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ County.Code : int 3 3 3 3 3 3 3 3 3 3 ...  
## $ Site.Num : int 10 10 10 10 10 10 10 10 10 10 ...  
## $ Parameter.Code : int 44201 44201 44201 44201 44201 44201 44201 44201 44201 44201 ...  
## $ POC : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ Latitude : num 30.5 30.5 30.5 30.5 30.5 ...  
## $ Longitude : num -87.9 -87.9 -87.9 -87.9 -87.9 ...  
## $ Datum : Factor w/ 2 levels "NAD83","WGS84": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Parameter.Name : Factor w/ 1 level "Ozone": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Date.Local : Factor w/ 366 levels "2016-01-01","2016-01-02",..: 61 61 61 61 61 61 61 61 61 62 ...  
## $ Time.Local : Factor w/ 24 levels "00:00","01:00",..: 16 17 18 19 20 21 22 23 24 2 ...  
## $ Date.GMT : Factor w/ 367 levels "2016-01-01","2016-01-02",..: 61 61 61 62 62 62 62 62 62 62 ...  
## $ Time.GMT : Factor w/ 24 levels "00:00","01:00",..: 22 23 24 1 2 3 4 5 6 8 ...  
## $ Sample.Measurement : num 0.041 0.041 0.042 0.041 0.038 0.038 0.036 0.035 0.029 0.026 ...  
## $ Units.of.Measure : Factor w/ 1 level "Parts per million": 1 1 1 1 1 1 1 1 1 1 ...  
## $ MDL : num 0.005 0.005 0.005 0.005 0.005 0.005 0.005 0.005 0.005 0.005 ...  
## $ Uncertainty : logi NA NA NA NA NA NA ...  
## $ Qualifier : Factor w/ 22 levels "","1","2","3",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ Method.Type : Factor w/ 2 levels "FEM","Non-FRM": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Method.Code : int 47 47 47 47 47 47 47 47 47 47 ...  
## $ Method.Name : Factor w/ 8 levels "Instrumental - Chemiluminescence API Model 265E and T265",..: 3 3 3 3 3 3 3 3 3 3 ...  
## $ State.Name : Factor w/ 52 levels "Alabama","Alaska",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ County.Name : Factor w/ 625 levels "Abbeville","Ada",..: 30 30 30 30 30 30 30 30 30 30 ...  
## $ Date.of.Last.Change: Factor w/ 309 levels "2016-02-01","2016-02-04",..: 84 84 84 84 84 84 84 84 84 84 ...

All the columns appear to have been correctly identified by type except for Time.Local and Time.GMT (we will combine these into datetimes), while the Parameter.Code could be transformed into a factor with a single level.

The date columns possess 366 and 367 levels. This is because 2016 was a leap year while the last measurements with the GMT datestamp extended into 2017.

head(df)

## State.Code County.Code Site.Num Parameter.Code POC Latitude Longitude  
## 1 1 3 10 44201 1 30.49748 -87.88026  
## 2 1 3 10 44201 1 30.49748 -87.88026  
## 3 1 3 10 44201 1 30.49748 -87.88026  
## 4 1 3 10 44201 1 30.49748 -87.88026  
## 5 1 3 10 44201 1 30.49748 -87.88026  
## 6 1 3 10 44201 1 30.49748 -87.88026  
## Datum Parameter.Name Date.Local Time.Local Date.GMT Time.GMT  
## 1 NAD83 Ozone 2016-03-01 15:00 2016-03-01 21:00  
## 2 NAD83 Ozone 2016-03-01 16:00 2016-03-01 22:00  
## 3 NAD83 Ozone 2016-03-01 17:00 2016-03-01 23:00  
## 4 NAD83 Ozone 2016-03-01 18:00 2016-03-02 00:00  
## 5 NAD83 Ozone 2016-03-01 19:00 2016-03-02 01:00  
## 6 NAD83 Ozone 2016-03-01 20:00 2016-03-02 02:00  
## Sample.Measurement Units.of.Measure MDL Uncertainty Qualifier  
## 1 0.041 Parts per million 0.005 NA   
## 2 0.041 Parts per million 0.005 NA   
## 3 0.042 Parts per million 0.005 NA   
## 4 0.041 Parts per million 0.005 NA   
## 5 0.038 Parts per million 0.005 NA   
## 6 0.038 Parts per million 0.005 NA   
## Method.Type Method.Code Method.Name State.Name  
## 1 FEM 47 INSTRUMENTAL - ULTRA VIOLET Alabama  
## 2 FEM 47 INSTRUMENTAL - ULTRA VIOLET Alabama  
## 3 FEM 47 INSTRUMENTAL - ULTRA VIOLET Alabama  
## 4 FEM 47 INSTRUMENTAL - ULTRA VIOLET Alabama  
## 5 FEM 47 INSTRUMENTAL - ULTRA VIOLET Alabama  
## 6 FEM 47 INSTRUMENTAL - ULTRA VIOLET Alabama  
## County.Name Date.of.Last.Change  
## 1 Baldwin 2016-06-20  
## 2 Baldwin 2016-06-20  
## 3 Baldwin 2016-06-20  
## 4 Baldwin 2016-06-20  
## 5 Baldwin 2016-06-20  
## 6 Baldwin 2016-06-20

tail(df)

## State.Code County.Code Site.Num Parameter.Code POC Latitude  
## 9121314 72 77 1 44201 1 18.17794  
## 9121315 72 77 1 44201 1 18.17794  
## 9121316 72 77 1 44201 1 18.17794  
## 9121317 72 77 1 44201 1 18.17794  
## 9121318 72 77 1 44201 1 18.17794  
## 9121319 72 77 1 44201 1 18.17794  
## Longitude Datum Parameter.Name Date.Local Time.Local Date.GMT  
## 9121314 -65.91548 WGS84 Ozone 2016-12-31 18:00 2016-12-31  
## 9121315 -65.91548 WGS84 Ozone 2016-12-31 19:00 2016-12-31  
## 9121316 -65.91548 WGS84 Ozone 2016-12-31 20:00 2017-01-01  
## 9121317 -65.91548 WGS84 Ozone 2016-12-31 21:00 2017-01-01  
## 9121318 -65.91548 WGS84 Ozone 2016-12-31 22:00 2017-01-01  
## 9121319 -65.91548 WGS84 Ozone 2016-12-31 23:00 2017-01-01  
## Time.GMT Sample.Measurement Units.of.Measure MDL Uncertainty  
## 9121314 22:00 0.003 Parts per million 0.005 NA  
## 9121315 23:00 0.005 Parts per million 0.005 NA  
## 9121316 00:00 0.003 Parts per million 0.005 NA  
## 9121317 01:00 0.002 Parts per million 0.005 NA  
## 9121318 02:00 0.002 Parts per million 0.005 NA  
## 9121319 03:00 0.002 Parts per million 0.005 NA  
## Qualifier Method.Type Method.Code Method.Name  
## 9121314 FEM 47 INSTRUMENTAL - ULTRA VIOLET  
## 9121315 FEM 47 INSTRUMENTAL - ULTRA VIOLET  
## 9121316 FEM 47 INSTRUMENTAL - ULTRA VIOLET  
## 9121317 FEM 47 INSTRUMENTAL - ULTRA VIOLET  
## 9121318 FEM 47 INSTRUMENTAL - ULTRA VIOLET  
## 9121319 FEM 47 INSTRUMENTAL - ULTRA VIOLET  
## State.Name County.Name Date.of.Last.Change  
## 9121314 Puerto Rico Juncos 2017-01-30  
## 9121315 Puerto Rico Juncos 2017-01-30  
## 9121316 Puerto Rico Juncos 2017-01-30  
## 9121317 Puerto Rico Juncos 2017-01-30  
## 9121318 Puerto Rico Juncos 2017-01-30  
## 9121319 Puerto Rico Juncos 2017-01-30

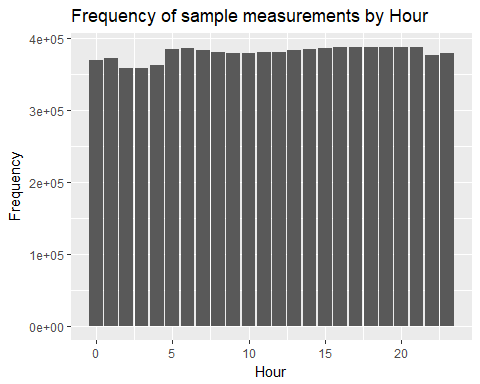
1. **Examine the hourly ozone measurements and check that all hours during the day are represented.**

The Lubridate library is used to create a datetime column 'date\_time' - by combining the Date.Local and Time.Local columns - and then extract the hour as an Hour column.

df$date\_time = ymd\_hm(paste(df$Date.Local, df$Time.Local))  
df$Hour = hour(df$date\_time)

Select the sample measurement and newly-created Hour column, group by Hour and then plot the number of measurements. This shows the number of measurements by hour - a graph which shows the distribution is broadly uniform.

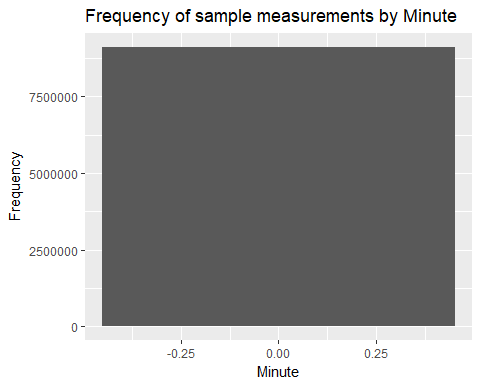
df %>%   
 select(Sample.Measurement, Hour) %>%   
 group\_by(Hour) %>%   
 count() %>%   
 ggplot(., mapping = aes(x=Hour, y =n))+  
 geom\_bar(stat = "identity")+  
 labs(  
 title = "Frequency of sample measurements by Hour",  
 y= "Frequency"  
 )



Check minute for when readings are taken. It appears they are all taken on the hour.

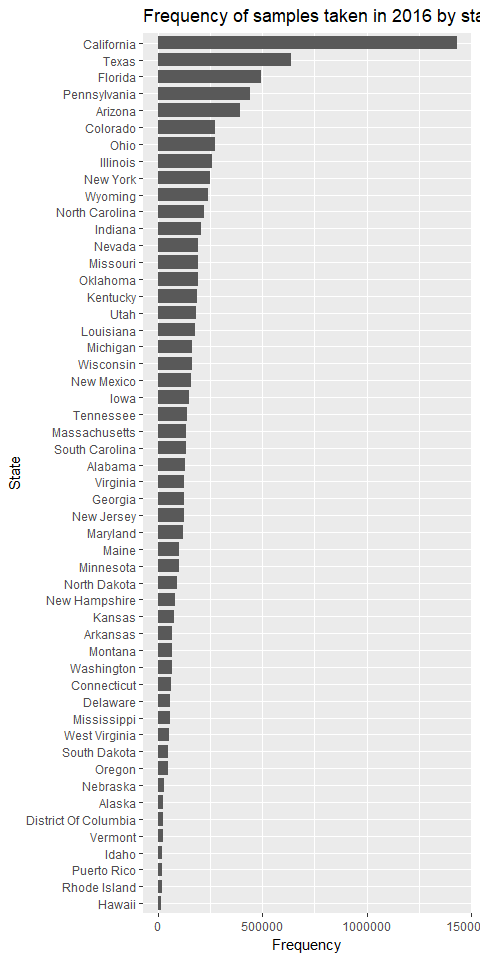
df$Minute = minute(df$date\_time)

df %>%   
 select(Sample.Measurement, Minute) %>%   
 group\_by(Minute) %>%   
 count() %>%   
 ggplot(., mapping = aes(x=Minute, y =n))+  
 geom\_bar(stat = "identity")+  
 labs(  
 title = "Frequency of sample measurements by Minute",  
 y= "Frequency"  
 )



We repeat the same examination of count of sample measurements, this this against state. It shows the state of California dominates in terms of the samples collected, with as many samples taken as the next highest two states - Texas and Florida - combined.

df %>%   
 select(Sample.Measurement, State.Name) %>%   
 group\_by(State.Name) %>%   
 count() %>%   
 arrange(n) %>%   
 ggplot(., mapping = aes(x=reorder(State.Name, n), y =n))+  
 geom\_bar(stat = "identity", width=.8, position = position\_dodge(width = .25))+  
 labs(  
 title = "Frequency of samples taken in 2016 by state",  
 y= "Frequency",  
 x = "State"  
 )+  
 coord\_flip()



Check the number of states listed in the data. The number is 52 against the 50 stated in the question. However, this includes Puerto Rico and Washington DC.

df %>%   
 select(State.Name) %>%   
 unique %>%   
 nrow

## [1] 52

1. **Validate your dataset with one external data source. This can be achieved by studying the recorded data and comparing it with the publicly available national air quality standards. For ozone measurements, the most current standard set in 2015 should be noted.**

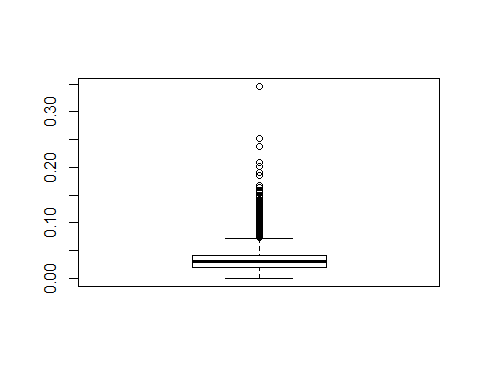
The standard according to the latest National Ambient Air Quality Standards (<https://www.epa.gov/criteria-air-pollutants/naaqs-table>) is that the maximum eight-hour concentration, averaged over three years, should not exceed 0.070 ppm.

If we compare this against distribution of the data for 2016, the median for 2016 was 0.031 ppm. However, the maximum value is almost five times larger than the maximum limit.Under normal circumstances, it would be worth going back into the data to establish the veracity of the recordings made. As these are likely to be erroneous, we take the decision to filter out readings above the top whisker which can be found using boxplot.stats (df$Sample.Measurement) as 0.070.

summary(df$Sample.Measurement)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00000 0.02000 0.03100 0.03044 0.04100 0.34600

boxplot(df$Sample.Measurement)



boxplot.stats(df$Sample.Measurement)

## $stats  
## [1] 0.000 0.020 0.031 0.041 0.072 (This last value is the upper whisker)

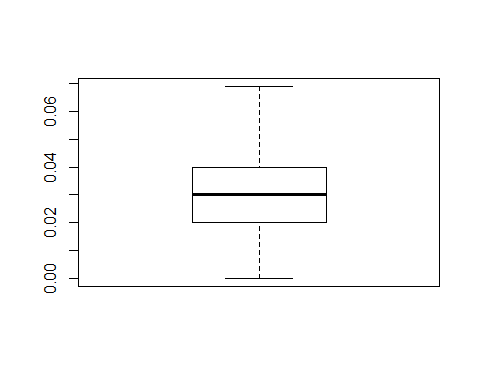
df <- df %>%   
 filter(Sample.Measurement < 0.070)

dim(df)

## [1] 9048447 27

The readjusted boxplot looks like:

boxplot(df$Sample.Measurement)



Another step we can take to check the data is by comparing the distribution of the data in 2016 against 2015. First we load in the data...

df\_2015 <- read.csv("hourly\_44201\_2015.csv")

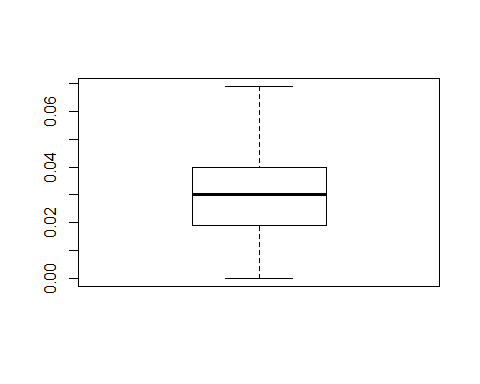
There are 1,618 sample measurements that are less than 0. We will remove them as before.

df\_2015 %>%   
 filter(Sample.Measurement < 0) %>%   
 count()

## # A tibble: 1 × 1  
## n  
## <int>  
## 1 1618

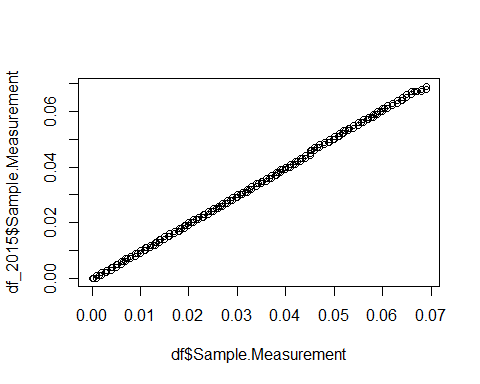
df\_2015 <- df\_2015 %>%   
 filter(Sample.Measurement >= 0 & Sample.Measurement < .07)

boxplot(df\_2015$Sample.Measurement)



In order to check the validity of the data for 2016, we will compute a QQ Plot of the Sample.Measurement data for 2016 and 2015. We should expect to see that the data would follow a broadly staight line if the two distributions are comparable. This is what we see in the following plot:

qqplot(df$Sample.Measurement, df\_2015$Sample.Measurement)



4. **Compile a list of states/counties which are ordered from highest to lowest with respect to their levels of ozone. This can be achieved as a preliminary investigation, by taking the average across the entire year for each state/county and then ranking counties according to this metric.**

**Top 10 highest levels of average Ozone in 2016**

df %>%   
 select(State.Name, County.Name, Sample.Measurement) %>%   
 group\_by(State.Name, County.Name) %>%   
 summarise(Mean.Ozone = mean(Sample.Measurement)) %>%   
 arrange(desc(Mean.Ozone))

## Source: local data frame [786 x 3]  
## Groups: State.Name [52]  
##   
## State.Name County.Name Mean.Ozone  
## <fctr> <fctr> <dbl>  
## 1 Wyoming Albany 0.04760823  
## 2 Colorado Clear Creek 0.04689654  
## 3 California Mariposa 0.04621983  
## 4 North Carolina Jackson 0.04579595  
## 5 North Carolina Yancey 0.04536798  
## 6 Colorado Park 0.04424932  
## 7 Utah San Juan 0.04393979  
## 8 California Nevada 0.04384125  
## 9 Nevada White Pine 0.04359447  
## 10 California El Dorado 0.04238799  
## # ... with 776 more rows

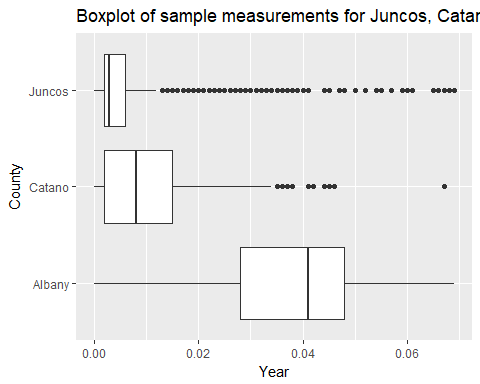
**Top 10 lowest levels of average Ozone in 2016**

df %>%   
 select(State.Name, County.Name, Sample.Measurement) %>%   
 group\_by(State.Name, County.Name) %>%   
 summarise(Mean.Ozone = mean(Sample.Measurement)) %>%   
 arrange(Mean.Ozone)

## Source: local data frame [786 x 3]  
## Groups: State.Name [52]  
##   
## State.Name County.Name Mean.Ozone  
## <fctr> <fctr> <dbl>  
## 1 Puerto Rico Juncos 0.006348360  
## 2 Puerto Rico Catano 0.009334708  
## 3 Puerto Rico Bayamon 0.012548015  
## 4 Alaska Fairbanks North Star 0.014479977  
## 5 Oregon Washington 0.017407850  
## 6 South Carolina Horry 0.019322158  
## 7 California Siskiyou 0.019478493  
## 8 New Jersey Warren 0.019881084  
## 9 Washington Whatcom 0.020032764  
## 10 Montana Missoula 0.020333066  
## # ... with 776 more rows

We can also plot histograms to give a better idea of the distribution of the data within the top and bottom counties within the data.

df %>%   
 filter(County.Name == "Juncos" | County.Name == "Albany" | County.Name == "Catano") %>%   
 select(County.Name, Sample.Measurement) %>%   
 ggplot(., mapping = aes(x = County.Name, y = Sample.Measurement))+  
 geom\_boxplot()+  
 coord\_flip()+  
 labs(  
 title = "Boxplot of sample measurements for Juncos, Catano and Albany",  
 y= "Year",  
 x = "County"  
 )



1. **Check the total number of observations extracted for the two states/counties identified in 4. Explain whether the total computed in each case makes sense**.

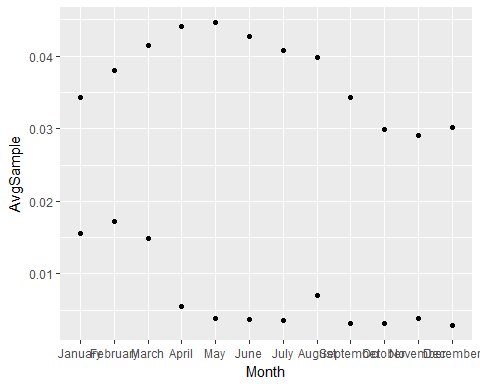
If there is one ozone measurement taken every hour throughout the 366 days of the year, this would account for 8,784 readings. The following code shows there have been almost twice as many readings taken in Albany, while there have been a 1,000 fewer taken in Juncos.

df %>%   
 filter(County.Name == "Juncos" | County.Name == "Albany"| County.Name == "Catano") %>%   
 select(County.Name, Sample.Measurement) %>%   
 group\_by(County.Name) %>%   
 summarise(Max = max(Sample.Measurement),  
 Min = min(Sample.Measurement),  
 Median = median(Sample.Measurement),  
 std\_dev = sd(Sample.Measurement),  
 count = n()  
 )

## # A tibble: 3 × 6  
## County.Name Max Min Median std\_dev count  
## <fctr> <dbl> <dbl> <dbl> <dbl> <int>  
## 1 Albany 0.069 0 0.041 0.013979383 16137  
## 2 Catano 0.067 0 0.008 0.008076756 8246  
## 3 Juncos 0.069 0 0.003 0.007752254 7254

1. **Examine how the ozone varies through the year in these two counties by studying their respective monthly averages. Display your results using graphical plots or visualisations supported by R. The lower path represents Juncos, the upper path represents the Albany data.**

df %>%   
 filter(County.Name == "Juncos" | County.Name == "Albany") %>%   
 mutate(Month = month(date\_time, label=TRUE, abbr=FALSE)) %>%   
 group\_by(County.Name, Month) %>%   
 select(Sample.Measurement) %>%   
 summarise(AvgSample = mean(Sample.Measurement)) %>%   
 ggplot(., mapping=aes(x = Month, y = AvgSample))+  
 geom\_path()+  
 labs(  
 title = "Average monthly ozone measurement - Juncos v Albany",  
 y= "Monthly average ozone measurement",  
 x = "Month"  
 )



1. **Describe any patterns you might observe from the results obtained in 6. Explain/justify them where applicable.**

There is a known correlation between Ozone levels and temperature, according to Bloomer et al (2010), which would fit with the seasonal increase from January to August before dropping down for Albany.

The background levels for Juncos appear to be generally much lower except for three months when it's much higher, particularly given the time of year.

This would be a cause for further investigation.

## Part 2

1. **Develop a procedure to assess how stable are the rankings from year to year. Given the 2016 dataset, this can be achieved, as a guide, by randomising the data to check if anything changes. Such a randomised process could approximate the data changing from one year to the next, giving us a sense of how stable the rankings are.**

**First we randomize the data using the sample function**

resample <- df %>%   
 select(State.Name, County.Name, Sample.Measurement)  
resample$randomised <- sample(df$Sample.Measurement)

Having randomised the data, we re-rank the data, first from max to lowest...

**Top 10 highest levels of average Ozone in 2016 (randomized)**

resample %>%   
 select(State.Name, County.Name, randomised) %>%   
 group\_by(State.Name, County.Name) %>%   
 summarise(Mean.Ozone = mean(randomised)) %>%   
 arrange(desc(Mean.Ozone))

## Source: local data frame [786 x 3]  
## Groups: State.Name [52]  
##   
## State.Name County.Name Mean.Ozone  
## <fctr> <fctr> <dbl>  
## 1 Kentucky Hancock 0.03077063  
## 2 Wisconsin Manitowoc 0.03073053  
## 3 Michigan Allegan 0.03072230  
## 4 Idaho Idaho 0.03059904  
## 5 Maryland Frederick 0.03059495  
## 6 Indiana Posey 0.03055752  
## 7 Georgia Douglas 0.03054716  
## 8 Georgia Clarke 0.03053874  
## 9 Minnesota Stearns 0.03050757  
## 10 North Carolina Pitt 0.03050044  
## # ... with 776 more rows

**Bottom 10 highest levels of average Ozone in 2016 (randomized)**

resample %>%   
 select(State.Name, County.Name, randomised) %>%   
 group\_by(State.Name, County.Name) %>%   
 summarise(Mean.Ozone = mean(randomised)) %>%   
 arrange(Mean.Ozone)

## Source: local data frame [786 x 3]  
## Groups: State.Name [52]   
## State.Name County.Name Mean.Ozone  
## <fctr> <fctr> <dbl>  
## 1 Illinois Macoupin 0.02933326  
## 2 North Carolina Alexander 0.02955401  
## 3 Oklahoma Cotton 0.02959122  
## 4 New York New York 0.02961314  
## 5 Iowa Clinton 0.02964354  
## 6 Oregon Marion 0.02965094  
## 7 South Carolina Berkeley 0.02965887  
## 8 Colorado Douglas 0.02966588  
## 9 South Carolina Oconee 0.02967205  
## 10 Indiana Elkhart 0.02968802

1. **Identify and illustrate any patterns that have been displayed following the application of the procedure developed in 1.**

An initial scan of the original top and bottom rankings and those achieved using the random sample show no similar rankings. However, this may be as a result of the decision to take out outliers from the 2016 dataset.

1. **Validate your results in 2, using the hourly ozone measurements collated for earlier years since 2010.**

**2010**

df\_2010 <- read.csv("hourly\_44201\_2010.csv") %>%   
 select(State.Name, County.Name, Sample.Measurement)  
df\_2010 %>%   
 select(State.Name, County.Name, Sample.Measurement) %>%   
 group\_by(State.Name, County.Name) %>%   
 summarise(Mean.Ozone = mean(Sample.Measurement)) %>%   
 arrange(desc(Mean.Ozone)) %>%   
 head()

## Source: local data frame [6 x 3]  
## Groups: State.Name [3]  
##   
## State.Name County.Name Mean.Ozone  
## <fctr> <fctr> <dbl>  
## 1 Colorado Clear Creek 0.05686899  
## 2 Colorado Gunnison 0.05065054  
## 3 Tennessee Sevier 0.04937869  
## 4 Colorado Chaffee 0.04917855  
## 5 Colorado Park 0.04913080  
## 6 Utah San Juan 0.04715162

**2011**

df\_2011 <- read.csv("hourly\_44201\_2011.csv") %>%   
 select(State.Name, County.Name, Sample.Measurement)  
df\_2011 %>%   
 select(State.Name, County.Name, Sample.Measurement) %>%   
 group\_by(State.Name, County.Name) %>%   
 summarise(Mean.Ozone = mean(Sample.Measurement)) %>%   
 arrange(desc(Mean.Ozone)) %>%   
 head()

## Source: local data frame [6 x 3]  
## Groups: State.Name [5]  
##   
## State.Name County.Name Mean.Ozone  
## <fctr> <fctr> <dbl>  
## 1 Colorado Clear Creek 0.05552918  
## 2 North Carolina Yancey 0.05076491  
## 3 Colorado Chaffee 0.05055212  
## 4 Wyoming Albany 0.04945584  
## 5 Nevada White Pine 0.04885879  
## 6 Tennessee Sevier 0.04842138

**2012**

df\_2012 <- read.csv("hourly\_44201\_2012.csv") %>%   
 select(State.Name, County.Name, Sample.Measurement)  
df\_2012 %>%   
 select(State.Name, County.Name, Sample.Measurement) %>%   
 group\_by(State.Name, County.Name) %>%   
 summarise(Mean.Ozone = mean(Sample.Measurement)) %>%   
 arrange(desc(Mean.Ozone)) %>%   
 head()

## Source: local data frame [6 x 3]  
## Groups: State.Name [4]  
##   
## State.Name County.Name Mean.Ozone  
## <fctr> <fctr> <dbl>  
## 1 Colorado Clear Creek 0.06209439  
## 2 Wyoming Albany 0.05123590  
## 3 Colorado Chaffee 0.04991726  
## 4 Arizona Gila 0.04929632  
## 5 California Mariposa 0.04859101  
## 6 Colorado Park 0.04835179

**2013**

df\_2013 <- read.csv("hourly\_44201\_2013.csv") %>%   
 select(State.Name, County.Name, Sample.Measurement)  
df\_2013 %>%   
 select(State.Name, County.Name, Sample.Measurement) %>%   
 group\_by(State.Name, County.Name) %>%   
 summarise(Mean.Ozone = mean(Sample.Measurement)) %>%   
 arrange(desc(Mean.Ozone)) %>%   
 head()

## Source: local data frame [6 x 3]  
## Groups: State.Name [3]  
##   
## State.Name County.Name Mean.Ozone  
## <fctr> <fctr> <dbl>  
## 1 Colorado Clear Creek 0.05544181  
## 2 Colorado Chaffee 0.05126348  
## 3 Wyoming Albany 0.04866779  
## 4 Colorado Park 0.04810279  
## 5 California Mariposa 0.04739876  
## 6 Colorado Pitkin 0.04686228

**2014**

df\_2014 <- read.csv("hourly\_44201\_2014.csv") %>%   
 select(State.Name, County.Name, Sample.Measurement)  
df\_2014 %>%   
 select(State.Name, County.Name, Sample.Measurement) %>%   
 group\_by(State.Name, County.Name) %>%   
 summarise(Mean.Ozone = mean(Sample.Measurement)) %>%   
 arrange(desc(Mean.Ozone)) %>%   
 head()

## Source: local data frame [6 x 3]  
## Groups: State.Name [3]  
##   
## State.Name County.Name Mean.Ozone  
## <fctr> <fctr> <dbl>  
## 1 Colorado Clear Creek 0.05022088  
## 2 California Mariposa 0.04849011  
## 3 California Nevada 0.04821713  
## 4 Colorado Park 0.04799988  
## 5 Wyoming Albany 0.04740057  
## 6 Colorado Chaffee 0.04732017

**2015**

df\_2015 %>%   
 select(State.Name, County.Name, Sample.Measurement) %>%   
 group\_by(State.Name, County.Name) %>%   
 summarise(Mean.Ozone = mean(Sample.Measurement)) %>%   
 arrange(desc(Mean.Ozone)) %>%   
 head()

## Source: local data frame [6 x 3]  
## Groups: State.Name [4]  
##   
## State.Name County.Name Mean.Ozone  
## <fctr> <fctr> <dbl>  
## 1 Colorado Clear Creek 0.04906275  
## 2 California Nevada 0.04659331  
## 3 Wyoming Albany 0.04642288  
## 4 Nevada Elko 0.04610694  
## 5 Colorado Chaffee 0.04583840  
## 6 California Mariposa 0.04543110