Alonso\_Week 6 Homework Assignment

Martin Alonso

8/23/2018

IST687 Introduction to Data Science: Week 6 Homework.

Visualization and air quality analysis.

### Part 1: The Code

#Load the packages that will be used in this exercise.   
require(ggplot2)  
require(dplyr)  
require(tidyr)  
require(scales)  
  
#Step 1: Loading the air quality data.  
air <- airquality  
  
#Let's check the data. The dataset has 153 rows and 6 columns.  
dim(air)  
str(air)  
head(air)  
  
#Step 2: Cleaning the data.  
#There are NAs in the data. How many are there in total?  
summary(air)  
  
#The Ozone column has 37 NAs (24%), while the Solar.R column has 7 NAs (4%)  
#Because the Ozone column presents two outliers, median replacement seems to be preferred. However, the Solar.R column will use mean replacement, since the data is more evenly spread and does not present outliers.  
air$Ozone[is.na(air$Ozone)] <- median(air$Ozone, na.rm = T)  
air$Solar.R[is.na(air$Solar.R)] <- mean(air$Solar.R, na.rm = T)  
  
#Making sure that there are no NAs in the dataset.  
summary(air)  
  
#To make the dataset easier to work with in the following steps, I'll adopt Hadley Wickham's TidyR principle.  
air\_gathered <- gather(air, key = Field, value = Measurement, c(-Month, -Day))   
  
#Step 3: Understand the data distribution.  
#Create the following visualizations:  
#3.1 - Histograms for each variable.  
ggplot(air\_gathered, aes(Measurement, fill = Field)) +   
 geom\_histogram() +   
 facet\_wrap(~Field, ncol = 2, scales = c('free')) +  
 theme(legend.position = 'hide')  
  
#3.2 - Boxplot for Ozone.  
ggplot(air\_gathered %>% filter(Field == 'Ozone'), aes(y = Measurement, group = 1)) +  
 geom\_boxplot()  
  
#3.2 - Boxplot for Wind.  
ggplot(air\_gathered %>% filter(Field == 'Wind'), aes(y = Measurement, group = 1)) +  
 geom\_boxplot()  
  
#Explore how the data changes over time.   
#The data is from 1973, so a year column needs to be added and then concatenated with Month and Day to create a time series.  
air\_gathered$Year <- 1973  
air\_gathered$Date <- as.Date(paste0(air\_gathered$Year, '-', air\_gathered$Month, '-', air\_gathered$Day))  
  
#We'll keep only the Date, Day, Field, and Measurement columns.  
air\_gathered <- air\_gathered %>% select(Date, Day, Field, Measurement)  
  
#Now to do some time series analysis.  
#First we'll check each field individually...  
ggplot(air\_gathered, aes(x = Date, y = Measurement)) +   
 geom\_line() +   
 facet\_wrap(~Field, scales = 'free\_y') #by using scales = 'free\_y', each Measurement has it's own defined y-axis while maintaining the same Date scale on the x-axis.  
  
#...Now let's check all measurements together.  
ggplot(air\_gathered, aes(x = Date, y = Measurement, color = Field)) +   
 geom\_line()  
  
#Step 4: Look at the data via a heatmap.  
#We'll create a heat map to show how the data varies by day.  
ggplot(air\_gathered, aes(x = Day, y = Field, fill = Measurement, group = Field)) + geom\_tile() + scale\_fill\_gradient(low = 'blue', high = 'red')  
  
#The problem is that the fields are not on the same scale, so we can't see real change across the different fields.  
#So let's normalize all the measurements on a 0 to 100 scale.  
air\_gathered <- air\_gathered %>% group\_by(Field, Day) %>%  
 mutate(Avg\_Measurement = mean(Measurement)) %>%   
 ungroup() %>% group\_by(Field) %>%  
 mutate(Measurement\_Norm = rescale(Measurement, to = c(0, 100)))  
  
ggplot(air\_gathered, aes(x = Day, y = Field, fill = Measurement\_Norm, group = Field)) + geom\_tile() + scale\_fill\_gradient(low = 'blue', high = 'red')  
  
#We can now see more clearly how Temperature, Wind, and Solar.R affect Ozone.  
  
#Step 5: Create a scatter plot.   
#We'll create a scatter plot with Wind on the x-axis, Temperature on the y-axis, dot size representing Ozone, and dot color representing Solar.R.   
#Because of this, we will revert to using the air dataset rather than air\_gathered.  
ggplot(air, aes(Wind, Temp)) + geom\_point(aes(size = Ozone, color = Solar.R))  
  
#Step 6: Final analysis.  
#1. Do you see any patterns after exploring the data?  
# Yes, from the last plot, we can see that Wind and Temperature are inversely correlated and, as Temperature rises and Wind decreases, Ozone and Solar.R increase as well, though these two also tend to decrease as Wind increases, regardless of Temperature.  
  
#2. What was the most useful visualization?  
# The most useful one was the heatmap which was the first to indicate that there was a relationship between all measurements. However, the scatter plot was the one to truly show the relationship between the four variables.

### Part 2: Running the Code

#Load the packages that will be used in this exercise.   
require(ggplot2)

## Loading required package: ggplot2

require(dplyr)

## Loading required package: dplyr

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

require(tidyr)

## Loading required package: tidyr

require(scales)

## Loading required package: scales

#Step 1: Loading the air quality data.  
air <- airquality  
  
#Let's check the data. The dataset has 153 rows and 6 columns.  
dim(air)

## [1] 153 6

str(air)

## 'data.frame': 153 obs. of 6 variables:  
## $ Ozone : int 41 36 12 18 NA 28 23 19 8 NA ...  
## $ Solar.R: int 190 118 149 313 NA NA 299 99 19 194 ...  
## $ Wind : num 7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...  
## $ Temp : int 67 72 74 62 56 66 65 59 61 69 ...  
## $ Month : int 5 5 5 5 5 5 5 5 5 5 ...  
## $ Day : int 1 2 3 4 5 6 7 8 9 10 ...

head(air)

## Ozone Solar.R Wind Temp Month Day  
## 1 41 190 7.4 67 5 1  
## 2 36 118 8.0 72 5 2  
## 3 12 149 12.6 74 5 3  
## 4 18 313 11.5 62 5 4  
## 5 NA NA 14.3 56 5 5  
## 6 28 NA 14.9 66 5 6

#Step 2: Cleaning the data.  
#There are NAs in the data. How many are there in total?  
summary(air)

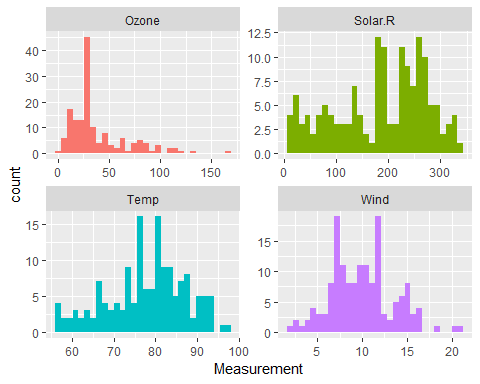
## Ozone Solar.R Wind Temp   
## Min. : 1.00 Min. : 7.0 Min. : 1.700 Min. :56.00   
## 1st Qu.: 18.00 1st Qu.:115.8 1st Qu.: 7.400 1st Qu.:72.00   
## Median : 31.50 Median :205.0 Median : 9.700 Median :79.00   
## Mean : 42.13 Mean :185.9 Mean : 9.958 Mean :77.88   
## 3rd Qu.: 63.25 3rd Qu.:258.8 3rd Qu.:11.500 3rd Qu.:85.00   
## Max. :168.00 Max. :334.0 Max. :20.700 Max. :97.00   
## NA's :37 NA's :7   
## Month Day   
## Min. :5.000 Min. : 1.0   
## 1st Qu.:6.000 1st Qu.: 8.0   
## Median :7.000 Median :16.0   
## Mean :6.993 Mean :15.8   
## 3rd Qu.:8.000 3rd Qu.:23.0   
## Max. :9.000 Max. :31.0   
##

#The Ozone column has 37 NAs (24%), while the Solar.R column has 7 NAs (4%)  
#Because the Ozone column presents two outliers, median replacement seems to be preferred. However, the Solar.R column will use mean replacement, since the data is more evenly spread and does not present outliers.  
air$Ozone[is.na(air$Ozone)] <- median(air$Ozone, na.rm = T)  
air$Solar.R[is.na(air$Solar.R)] <- mean(air$Solar.R, na.rm = T)  
  
#Making sure that there are no NAs in the dataset.  
summary(air)

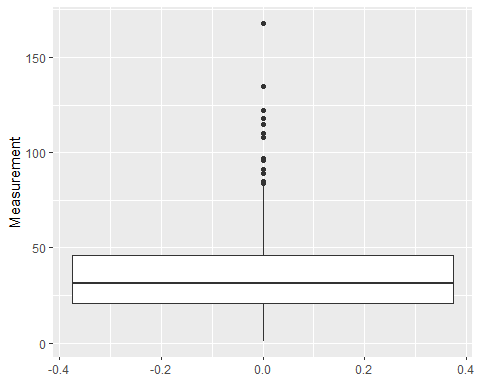
## Ozone Solar.R Wind Temp   
## Min. : 1.00 Min. : 7.0 Min. : 1.700 Min. :56.00   
## 1st Qu.: 21.00 1st Qu.:120.0 1st Qu.: 7.400 1st Qu.:72.00   
## Median : 31.50 Median :194.0 Median : 9.700 Median :79.00   
## Mean : 39.56 Mean :185.9 Mean : 9.958 Mean :77.88   
## 3rd Qu.: 46.00 3rd Qu.:256.0 3rd Qu.:11.500 3rd Qu.:85.00   
## Max. :168.00 Max. :334.0 Max. :20.700 Max. :97.00   
## Month Day   
## Min. :5.000 Min. : 1.0   
## 1st Qu.:6.000 1st Qu.: 8.0   
## Median :7.000 Median :16.0   
## Mean :6.993 Mean :15.8   
## 3rd Qu.:8.000 3rd Qu.:23.0   
## Max. :9.000 Max. :31.0

#To make the dataset easier to work with in the following steps, I'll adopt Hadley Wickham's TidyR principle.  
air\_gathered <- gather(air, key = Field, value = Measurement, c(-Month, -Day))   
  
#Step 3: Understand the data distribution.  
#Create the following visualizations:  
#3.1 - Histograms for each variable.  
ggplot(air\_gathered, aes(Measurement, fill = Field)) +   
 geom\_histogram() +   
 facet\_wrap(~Field, ncol = 2, scales = c('free')) +  
 theme(legend.position = 'hide')

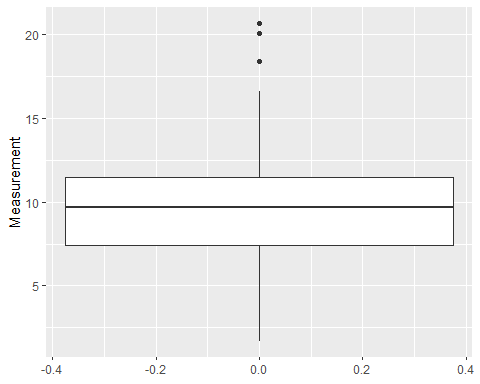
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



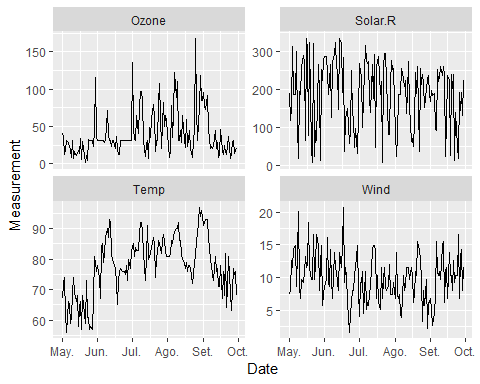
#3.2 - Boxplot for Ozone.  
ggplot(air\_gathered %>% filter(Field == 'Ozone'), aes(y = Measurement, group = 1)) +  
 geom\_boxplot()



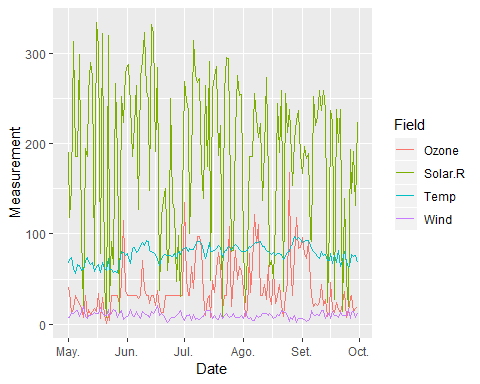
#3.2 - Boxplot for Wind.  
ggplot(air\_gathered %>% filter(Field == 'Wind'), aes(y = Measurement, group = 1)) +  
 geom\_boxplot()



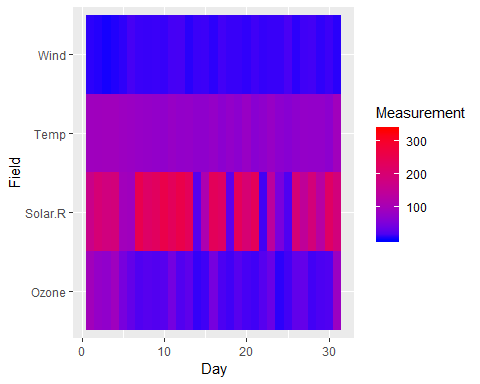
#Explore how the data changes over time.   
#The data is from 1973, so a year column needs to be added and then concatenated with Month and Day to create a time series.  
air\_gathered$Year <- 1973  
air\_gathered$Date <- as.Date(paste0(air\_gathered$Year, '-', air\_gathered$Month, '-', air\_gathered$Day))  
  
#We'll keep only the Date, Day, Field, and Measurement columns.  
air\_gathered <- air\_gathered %>% select(Date, Day, Field, Measurement)  
  
#Now to do some time series analysis.  
#First we'll check each field individually...  
ggplot(air\_gathered, aes(x = Date, y = Measurement)) +   
 geom\_line() +   
 facet\_wrap(~Field, scales = 'free\_y') #by using scales = 'free\_y', each Measurement has it's own defined y-axis while maintaining the same Date scale on the x-axis.



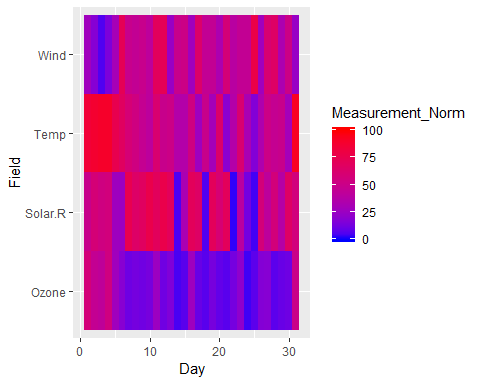
#...Now let's check all measurements together.  
ggplot(air\_gathered, aes(x = Date, y = Measurement, color = Field)) +   
 geom\_line()



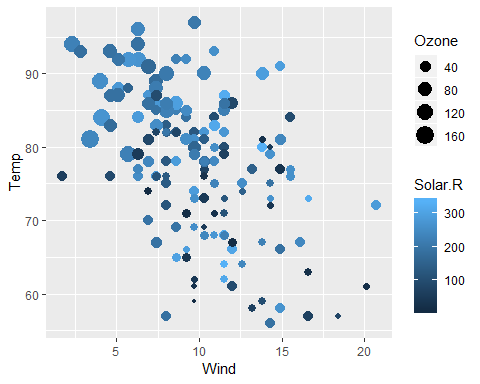
#Step 4: Look at the data via a heatmap.  
#We'll create a heat map to show how the data varies by day.  
ggplot(air\_gathered, aes(x = Day, y = Field, fill = Measurement, group = Field)) + geom\_tile() + scale\_fill\_gradient(low = 'blue', high = 'red')



#The problem is that the fields are not on the same scale, so we can't see real change across the different fields.  
#So let's normalize all the measurements on a 0 to 100 scale.  
air\_gathered <- air\_gathered %>% group\_by(Field, Day) %>%  
 mutate(Avg\_Measurement = mean(Measurement)) %>%   
 ungroup() %>% group\_by(Field) %>%  
 mutate(Measurement\_Norm = rescale(Measurement, to = c(0, 100)))  
  
ggplot(air\_gathered, aes(x = Day, y = Field, fill = Measurement\_Norm, group = Field)) + geom\_tile() + scale\_fill\_gradient(low = 'blue', high = 'red')



#We can now see more clearly how Temperature, Wind, and Solar.R affect Ozone.  
  
#Step 5: Create a scatter plot.   
#We'll create a scatter plot with Wind on the x-axis, Temperature on the y-axis, dot size representing Ozone, and dot color representing Solar.R.   
#Because of this, we will revert to using the air dataset rather than air\_gathered.  
ggplot(air, aes(Wind, Temp)) + geom\_point(aes(size = Ozone, color = Solar.R))



#Step 6: Final analysis.  
#1. Do you see any patterns after exploring the data?  
# Yes, from the last plot, we can see that Wind and Temperature are inversely correlated and, as Temperature rises and Wind decreases, Ozone and Solar.R increase as well, though these two also tend to decrease as Wind increases, regardless of Temperature.  
  
#2. What was the most useful visualization?  
# The most useful one was the heatmap which was the first to indicate that there was a relationship between all measurements. However, the scatter plot was the one to truly show the relationship between the four variables.

### Sources

#### <https://vita.had.co.nz/papers/tidy-data.pdf>