[redacted]

Brian Hogan

[redacted]

[redacted]

```r
#Step:1 Load Data-----------------------------------------------Brian HOgan-------
str(airquality) #153x6: "Ozone","Solar.R","Wind","Temp","Month","Day
```

```
## '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 ...
```

```r
apply(apply(airquality,2,is.na),2,sum)   #identify NAs
```

```
##   Ozone Solar.R    Wind    Temp   Month     Day
##      37       7       0       0       0       0
```

```r
#Step 2: Clean Data-----------------------------------------------------------
# ==> wanted to use rnorm(ozone.mean(& sd) but had issue neg #s  >:O(
ozone <-airquality[,1]  #substitute NAs with Ozone & Solar.R means
f.ozone <- function(x)
  {   x[is.na(x)] <-  round(mean(na.omit(airquality$Ozone)),1)
      # rnorm(100,mean(na.omit(my.airquality$Ozone)),sd(na.omit(my.airquality$Ozone)))
      return(x) }
solar <-airquality[,2]
f.solar <- function(x)
  {   x[is.na(x)] <-  round(mean(na.omit(airquality$Solar.R)),1)
      return(x) }
ozone <-f.ozone(ozone)
solar <-f.solar(solar)
wind <-round(airquality$Wind) #round wind
hw6.data <-data.frame(ozone,solar,wind,airquality$Temp,airquality$Month,airquality$Day)
colnames(hw6.data) <- colnames(airquality)
remove(ozone,solar,wind)
#Step 3.1: Understand data distributions----------------------------------------
col.names <- colnames(hw6.data)
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.2
```

```r
library(reshape2)
```

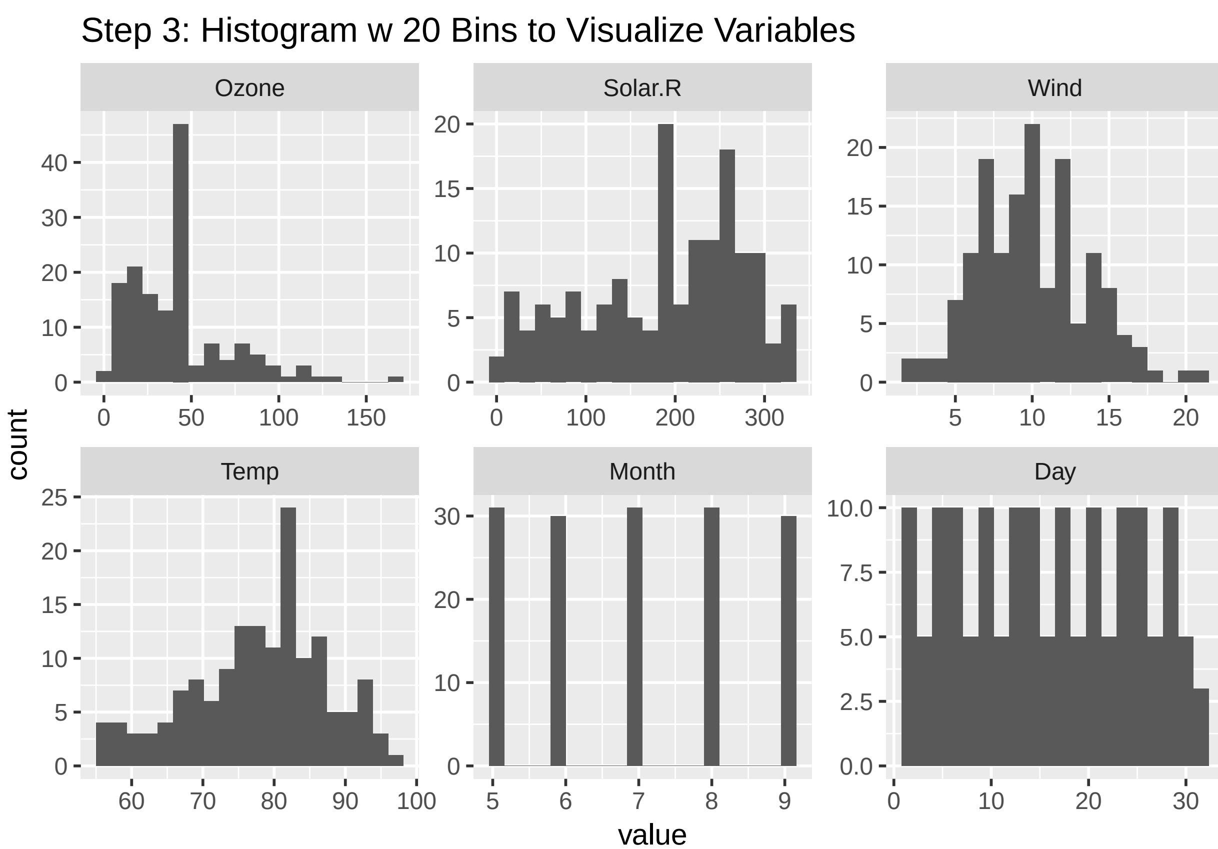
```
## Warning: package 'reshape2' was built under R version 3.5.2
```

```r
head(hw6.data,3)
```

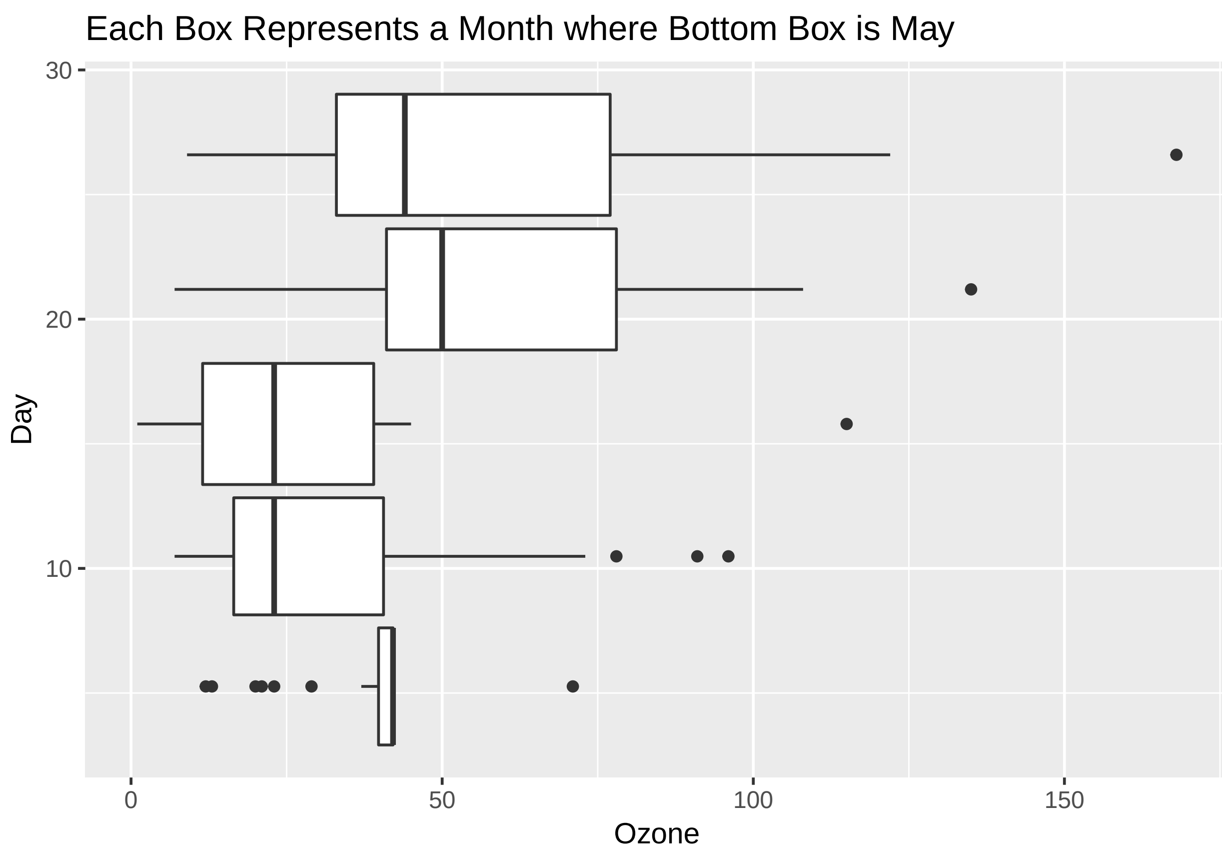
```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190    7   67     5   1
## 2    36     118    8   72     5   2
## 3    12     149   13   74     5   3
```

```r
ggplot(data = melt(hw6.data[,c(1:6)]), mapping = aes(x = value)) +
  facet_wrap(~variable, scales = "free") + geom_histogram(bins = 20) +
  ggtitle("Step 3: Histogram w 20 Bins to Visualize Variables")
```

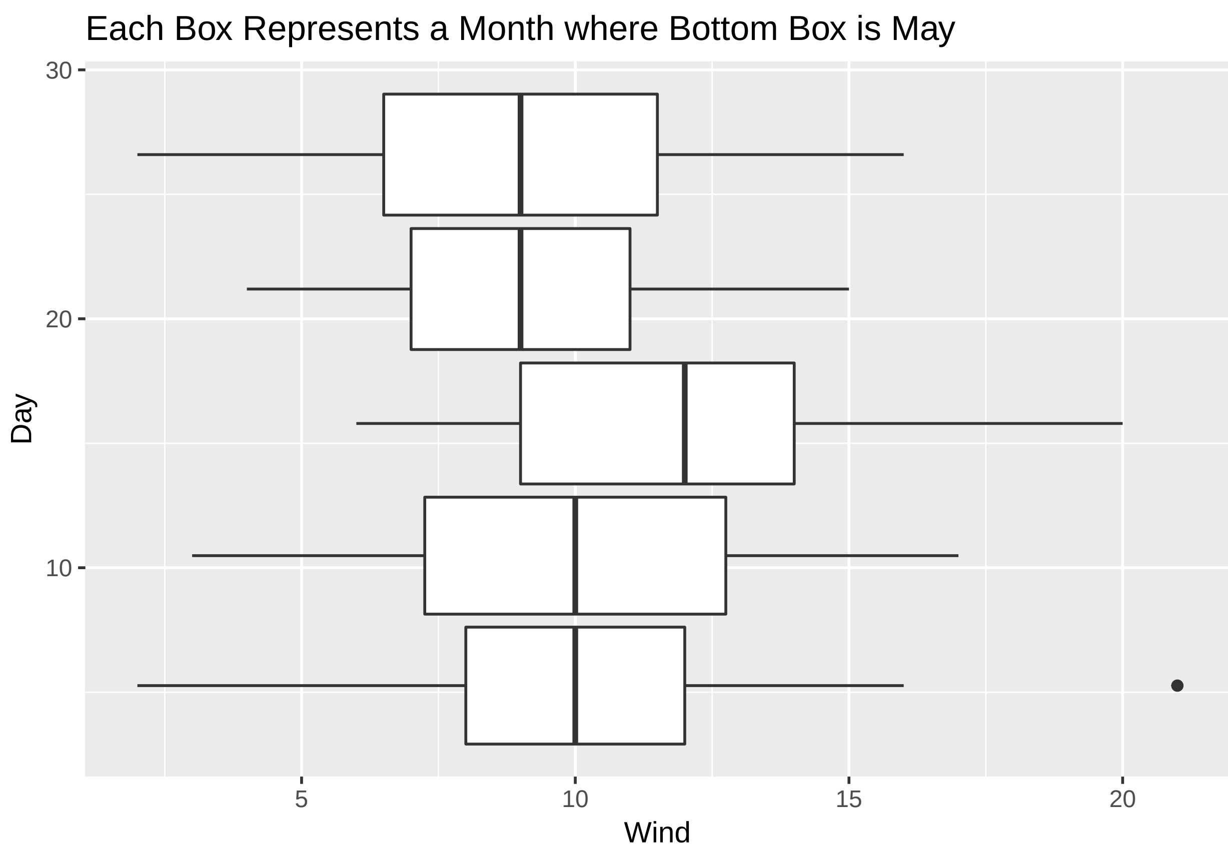
```
## No id variables; using all as measure variables
```




```r
ggplot(hw6.data,aes(x=Day,y=Ozone,group=Month))+geom_boxplot() +
  coord_flip() +ggtitle("Each Box Represents a Month where Bottom Box is May")
```



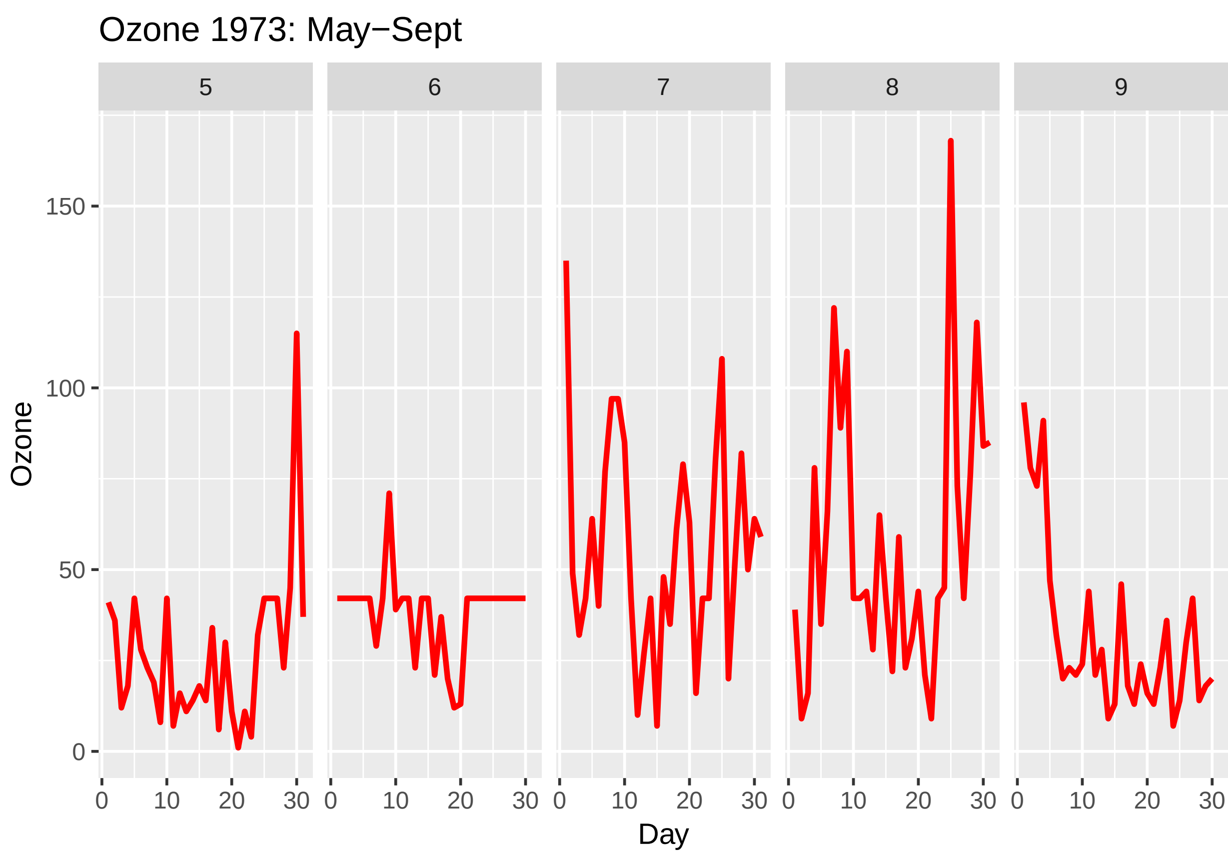

```r
ggplot(hw6.data,aes(x=Day, y=Wind, group=Month))+geom_boxplot()+
  coord_flip()+ggtitle("Each Box Represents a Month where Bottom Box is May")
```



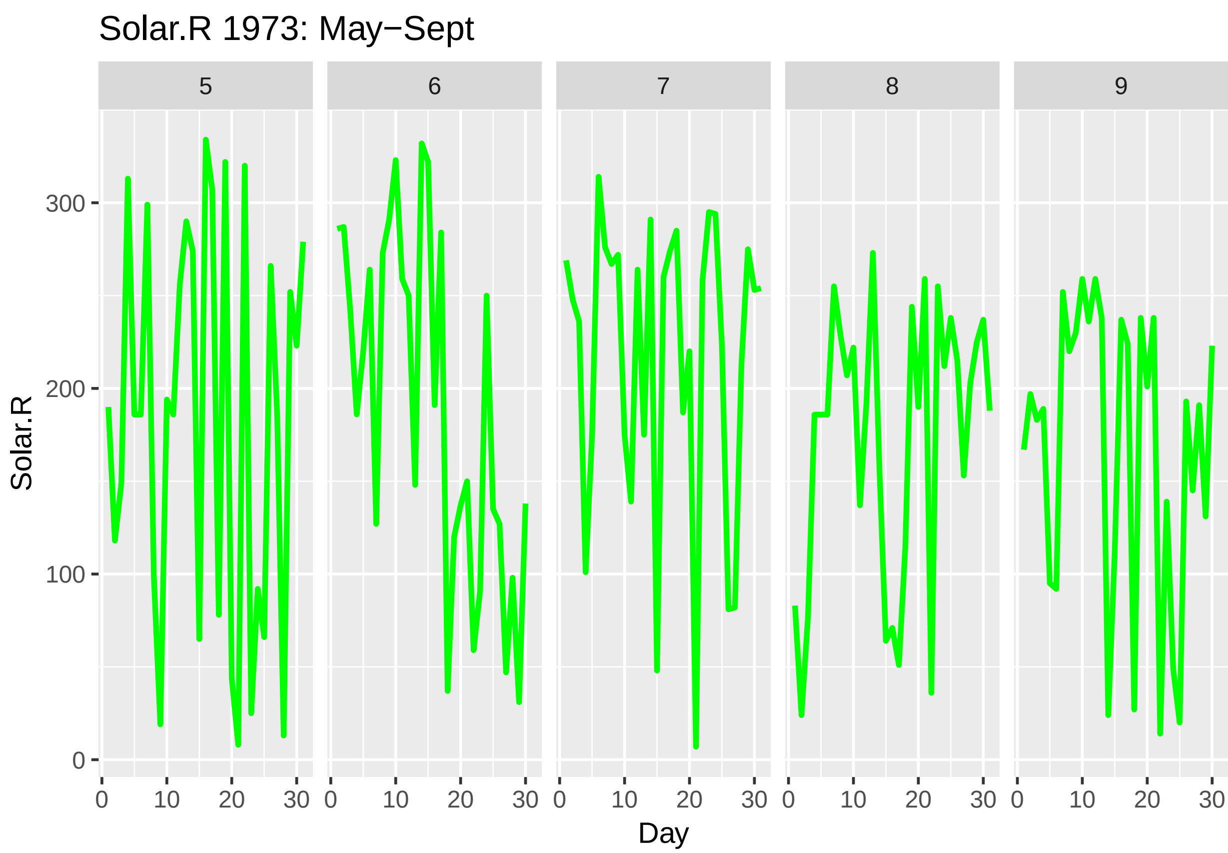

```
#Step 3.2: Explore how the data changes over time---------------------------
list1 <- c(1:153)
Year <- rep(1973, length(list1))    #add in date as requested !
i <-1
mth.name <- 1
mth.name <-NULL
while (i<=153) {       #making facet labels by MOnth
  if (hw6.data[i,"Month"]==5) {mth.name <-c(mth.name,"May")}
  if (hw6.data[i,"Month"]==6) {mth.name <-c(mth.name,"June")}
  if (hw6.data[i,"Month"]==7) {mth.name <-c(mth.name,"July")}
  if (hw6.data[i,"Month"]==8) {mth.name <-c(mth.name,"Aug")}
  if (hw6.data[i,"Month"]==9) {mth.name <-c(mth.name,"Sept")}
  i <-i+1  }
df3.2 <- data.frame(hw6.data,Year,mth.name)
yr.m.day <- paste(df3.2$Year,df3.2$Month, df3.2$Day, sep=".") #DATE !
df3.2 <- data.frame(hw6.data,Year,mth.name,yr.m.day)
head(df3.2)
```

```
##   Ozone Solar.R Wind Temp Month Day Year mth.name yr.m.day
## 1  41.0   190.0    7   67     5   1 1973      May 1973.5.1
## 2  36.0   118.0    8   72     5   2 1973      May 1973.5.2
## 3  12.0   149.0   13   74     5   3 1973      May 1973.5.3
## 4  18.0   313.0   12   62     5   4 1973      May 1973.5.4
## 5  42.1   185.9   14   56     5   5 1973      May 1973.5.5
## 6  28.0   185.9   15   66     5   6 1973      May 1973.5.6
```

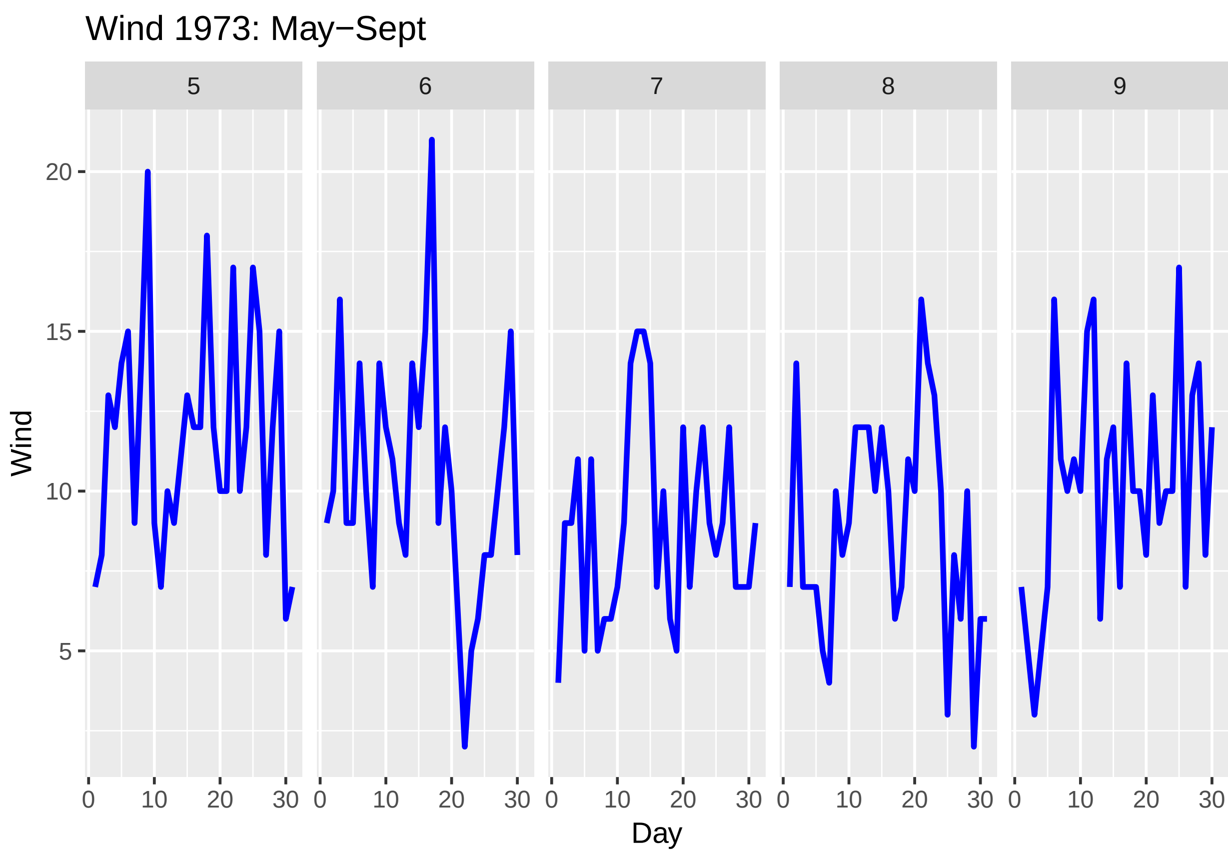
```r
remove(list1,Year,mth.name, i,yr.m.day)  #clean memory
ggplot(df3.2, aes(x=Day, y=Ozone))+ geom_line(color="red", size=1)+
  ggtitle("Ozone 1973: May-Sept") + facet_grid(. ~ Month)
```



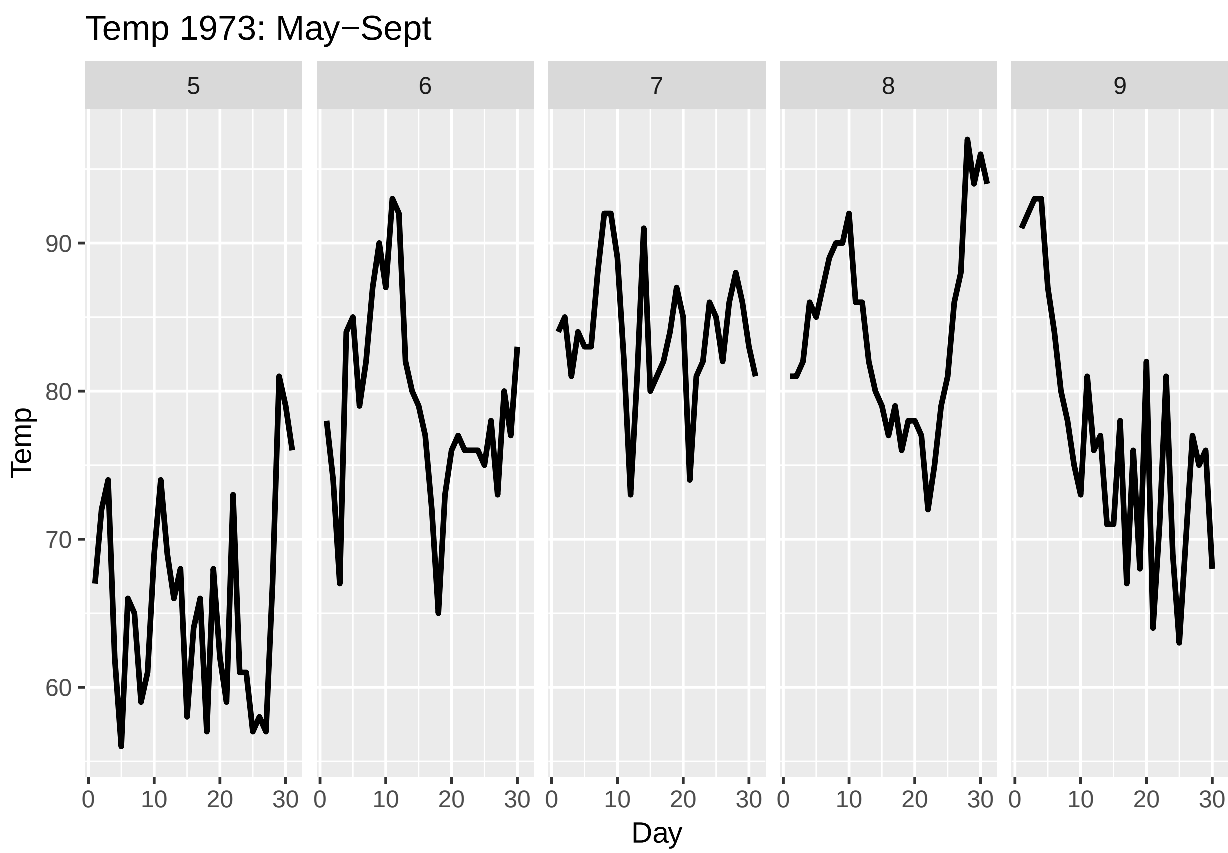

```r
ggplot(df3.2, aes(x=Day, y=Solar.R))+ geom_line(color="green", size=1)+
  ggtitle("Solar.R 1973: May-Sept") + facet_grid(. ~ Month)
```



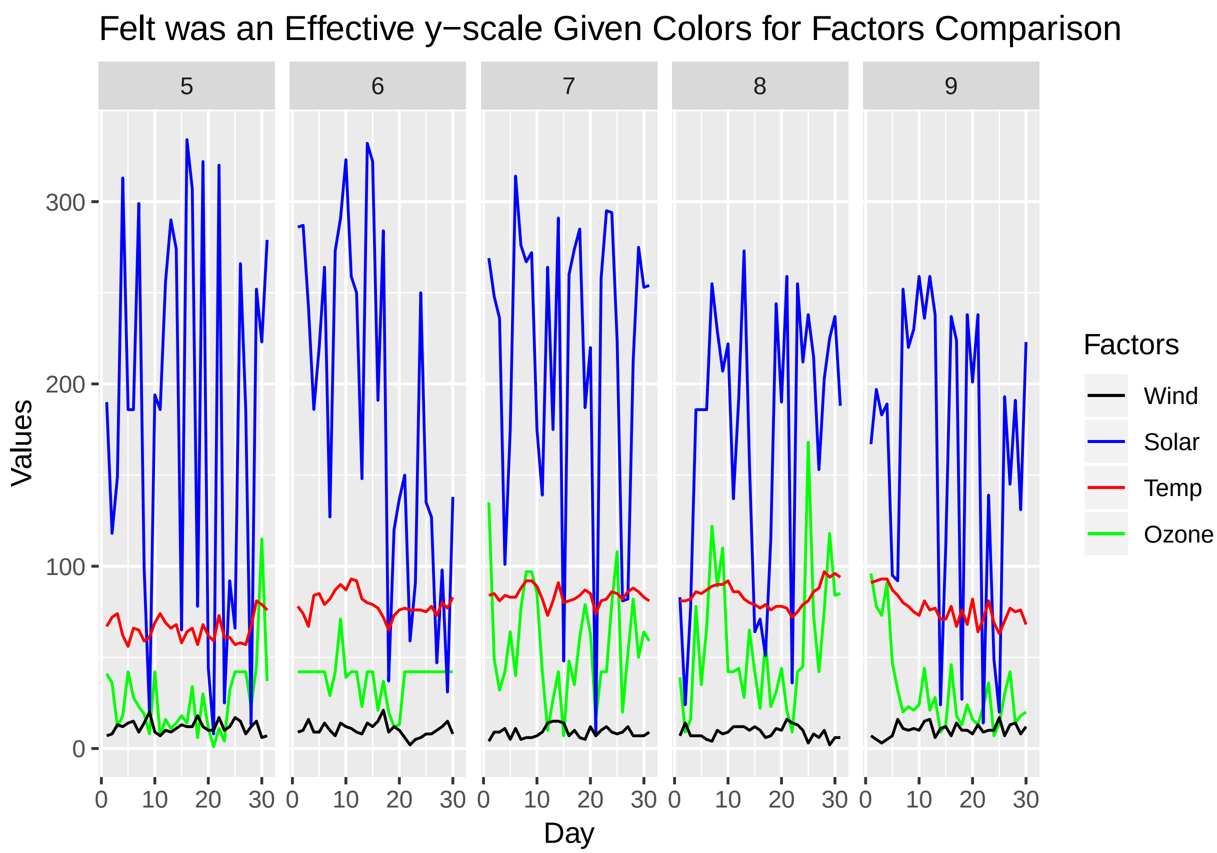

```r
ggplot(df3.2, aes(x=Day, y=Wind))+ geom_line(color="blue", size=1)+
  ggtitle("Wind 1973: May-Sept")+ facet_grid(. ~ Month)
```




```r
ggplot(df3.2, aes(x=Day, y=Temp))+ geom_line(color="black", size=1)+
  ggtitle("Temp 1973: May-Sept")+ facet_grid(. ~ Month)
```




```
#3.2.2-----Multiline graph------------------------------------------------------
ggplot(df3.2, aes(x=value)) +
  geom_line( aes(y=Ozone, x=Day, color="red")) +      #col=Ozone
  geom_line( aes(y=Solar.R, x=Day, color="blue")) +
  geom_line( aes(y=Wind, x=Day, color="black")) +
  geom_line( aes(y=Temp, x=Day, color="green"))  +
    facet_grid(. ~ Month) + ylab("Values") + xlab("Day") +
    ggtitle("1973 Weather May-Sept") +
    scale_color_manual(values = c("black","blue","red","green" ),
    labels=c("Wind","Solar","Temp","Ozone")) + labs(color="Factors") +
    ggtitle("Felt was an Effective y-scale Given Colors for Factors Comparison")
```

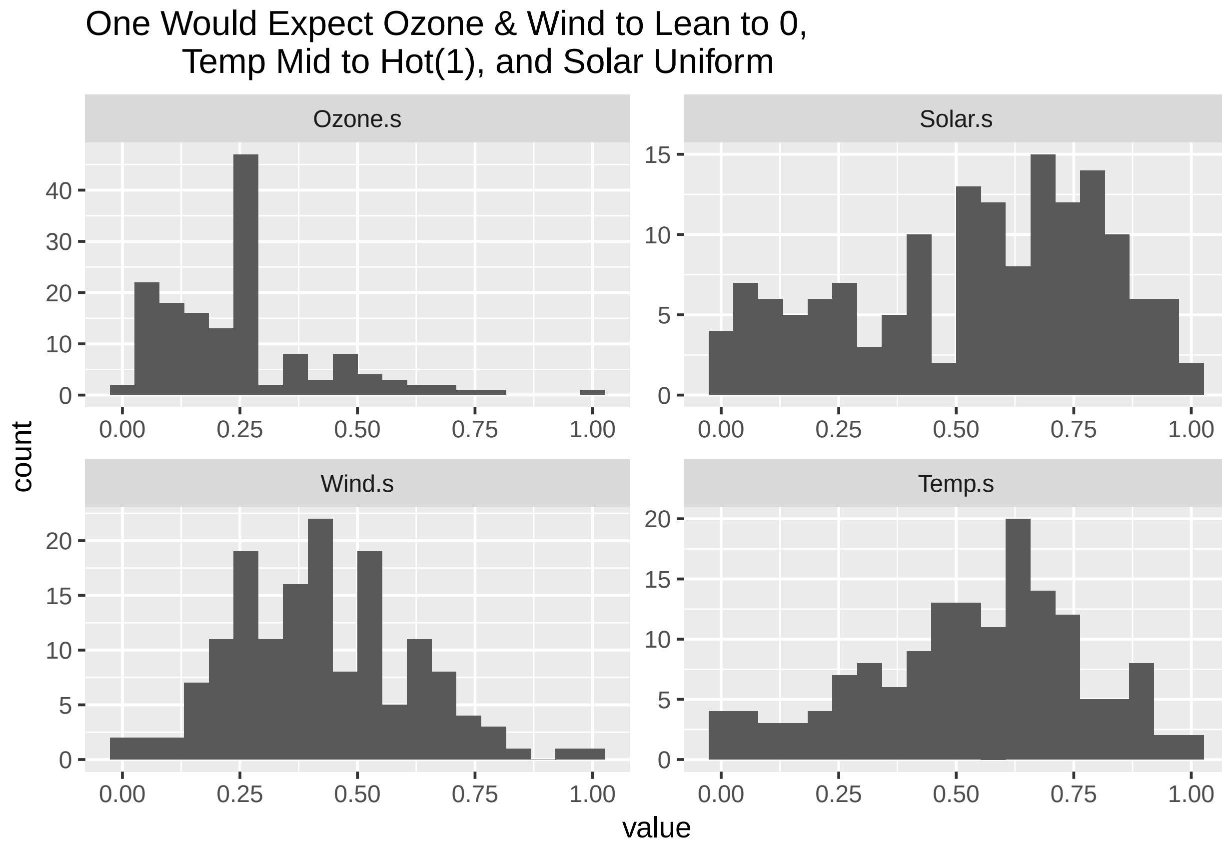



```
#Step 4 Heatmap---using normalization approach between 0-1 as didnt' want negative scale
Ozone.s <- (df3.2[,1]-min(df3.2$Ozone))  /(max(df3.2$Ozone-min(df3.2$Ozone)))
Solar.s <- (df3.2[,2]-min(df3.2$Solar.R))/(max(df3.2$Solar.R-min(df3.2$Solar.R)))
Wind.s <- (df3.2[,3]-min(df3.2$Wind))   /(max(df3.2$Wind-min(df3.2$Wind)))
Temp.s <- (df3.2[,4]-min(df3.2$Temp))   /(max(df3.2$Temp-min(df3.2$Temp)))
df4 <-data.frame(df3.2,Ozone.s,Solar.s,Wind.s,Temp.s)
df4 <-df4[-1:-4]
remove(Ozone.s,Solar.s,Wind.s,Temp.s)
summary(df4[c(-1:-5)])
```

```
##     Ozone.s          Solar.s            Wind.s            Temp.s
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.1198   1st Qu.:0.3456   1st Qu.:0.2632   1st Qu.:0.3902
##  Median :0.2461   Median :0.5719   Median :0.4211   Median :0.5610
##  Mean   :0.2462   Mean   :0.5472   Mean   :0.4221   Mean   :0.5337
##  3rd Qu.:0.2695   3rd Qu.:0.7615   3rd Qu.:0.5263   3rd Qu.:0.7073
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
```

```
#Comparison to learn if need to transform data to interpret better in heatmap
ggplot(data = melt(df4[,c(5:9)]), mapping = aes(x = value)) + geom_histogram(bins=20)+
  facet_wrap(~variable, scales = "free") +
  ggtitle("One Would Expect Ozone & Wind to Lean to 0,
          Temp Mid to Hot(1), and Solar Uniform")
```

```
## Using yr.m.day as id variables
```



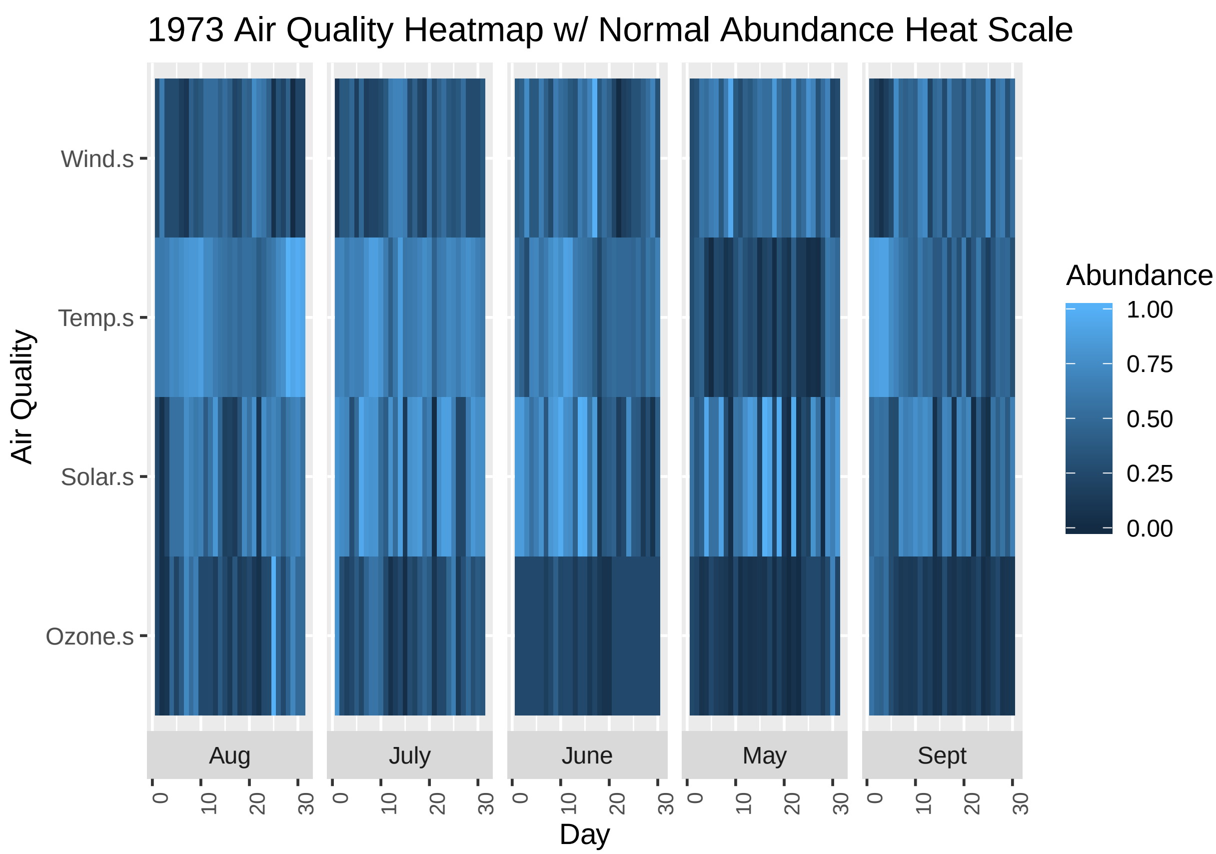

```r
#install.packages("tidyr")      #CLASS SLACK HELP was INVALUABLE !
library("tidyr")
```

```
## Warning: package 'tidyr' was built under R version 3.5.2

##
## Attaching package: 'tidyr'

## The following object is masked from 'package:reshape2':
##
##     smiths
```

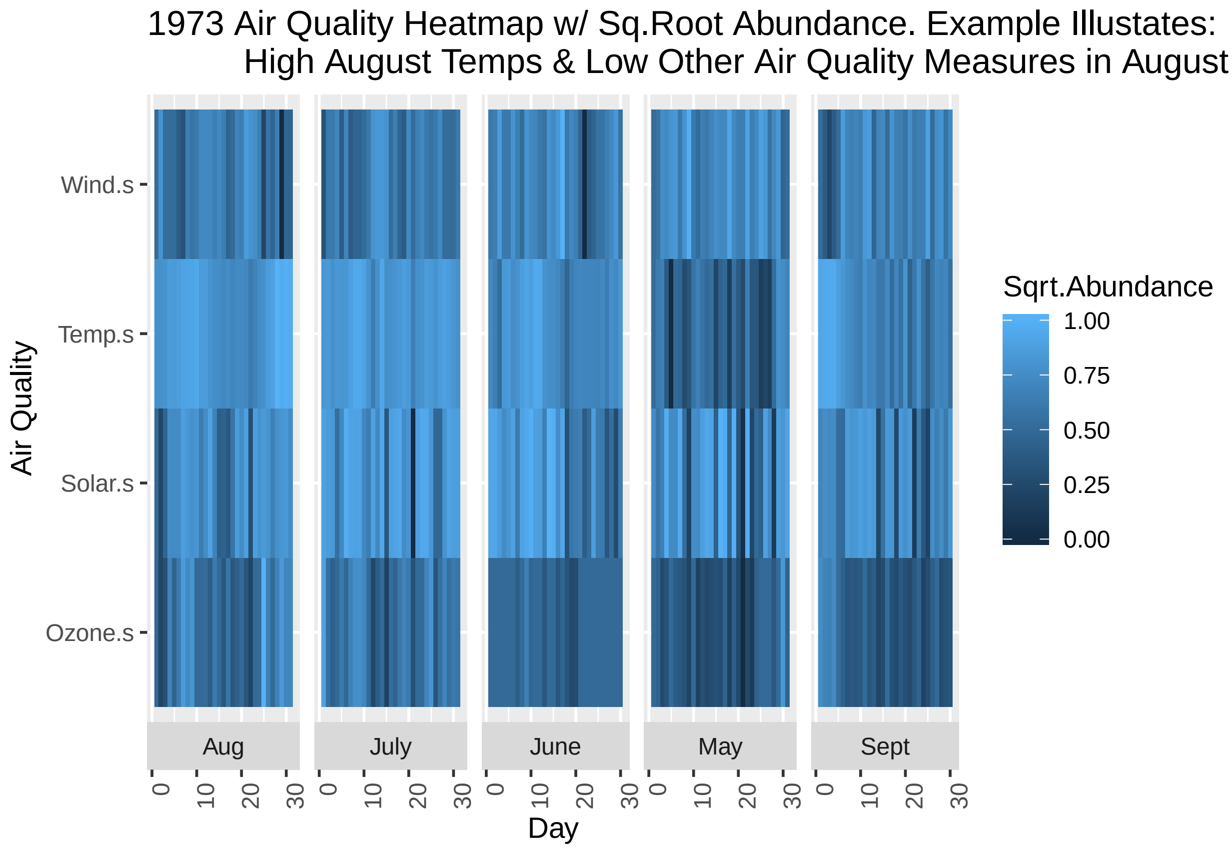
```r
#Abundance = grouping of all air quality factor values 0-1
#Felt Square Root Transformation did help with graph read but could vary by person
df5 <- gather(data = df4, key = Class, value= Abundance,-c(1:5))
df5$Sqrt.Abundance <- sqrt(df5$Abundance)
head(df5,2)
```

```
##   Month Day Year mth.name yr.m.day   Class Abundance Sqrt.Abundance
## 1     5   1 1973      May 1973.5.1 Ozone.s 0.2395210      0.4894088
## 2     5   2 1973      May 1973.5.2 Ozone.s 0.2095808      0.4578000
```

```r
heat.reg <-ggplot(data=df5, mapping= aes(x=Day,y=Class,fill=Abundance))+
  geom_tile() + xlab(label="Day") + ylab(label="Air Quality") +
  ggtitle("1973 Air Quality Heatmap w/ Normal Abundance Heat Scale") +
  theme(axis.text.x = element_text(size=8,angle = 90, hjust = 1)) +
  facet_grid(~ mth.name, switch = "x", scales="free_x", space="free_x")
heat.reg #non transformed data
```




```r
heat.sqrt <-ggplot(data=df5, mapping= aes(x=Day,y=Class,fill=Sqrt.Abundance))+
  geom_tile() + xlab(label="Day") + ylab(label="Air Quality") +
  ggtitle("1973 Air Quality Heatmap w/ Sq.Root Abundance. Example Illustates:
          High August Temps & Low Other Air Quality Measures in August") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  facet_grid(~ mth.name, switch = "x", scales="free_x", space="free_x")
heat.sqrt  #transform data to accentuate the values
```



END HOMEWORK 6