

LewisDB_HW6.R

dblewis

Fri Feb 22 10:34:24 2019

```
#####
#Name: Daniel Lewis
#Description: Homework Assignment 5
#Date: 2/22/19 (JST)
#input: airquality data set
#output: 4 histograms, 5 line charts, 1 heatmap, 1 scatter plot
#update history:
#####
#####HW6 Viz HW: air quality Analysis#####
#####
#
#####
#####LOCAL FUNCTIONS#####
#####
#
EnsurePackage<-function(x){
  x<-as.character(x)
  if (!require(x,character.only=TRUE)){
    install.packages(pkgs=x, repos="http://cran.r-project.org")
    require(x, character.only=TRUE)
  }
}
Numberize <- function(inputVector)
{
  inputVector <- gsub(",", "", inputVector)
  inputVector <- gsub(" ", "", inputVector)
  return(inputVector)
}
#
#####
#####IMPORTS SECTION#####
#####
#
EnsurePackage("ggplot2")

## Loading required package: ggplot2

## Warning: S3 methods '[.fun_list', '[.grouped_df', 'all.equal.tbl_df',
## 'anti_join.data.frame', 'anti_join.tbl_df', 'arrange.data.frame',
## 'arrange.default', 'arrange.grouped_df', 'arrange.tbl_df',
## 'arrange_.data.frame', 'arrange_.tbl_df', 'as.data.frame.grouped_df',
## 'as.data.frame.rowwise_df', 'as.data.frame.tbl_cube', 'as.table.tbl_cube',
```

```
## 'as.tbl.data.frame', 'as.tbl.tbl', 'as.tbl_cube.array',
## 'as.tbl_cube.data.frame', 'as.tbl_cube.matrix', 'as.tbl_cube.table',
## 'as_tibble.grouped_df', 'as_tibble.tbl_cube', 'auto_copy.tbl_cube',
## 'auto_copy.tbl_df', 'cbind.grouped_df', 'collapse.data.frame',
## 'collect.data.frame', 'common_by.NULL', 'common_by.character',
## 'common_by.default', 'common_by.list', 'compute.data.frame',
## 'copy_to.DBIConnection', 'copy_to.src_local',
'default_missing.data.frame',
## 'default_missing.default', 'dim.tbl_cube', 'distinct.data.frame',
## 'distinct.default', 'distinct.grouped_df', 'distinct.tbl_df',
## 'distinct_.data.frame', 'distinct_.grouped_df', 'distinct_.tbl_df',
## 'do.NULL', 'do.data.frame', 'do.default', 'do.grouped_df',
'do.rowwise_df',
## 'do_.NULL', 'do_.data.frame', 'do_.grouped_df', 'do_.rowwise_df',
## 'filter.data.frame', 'filter.default', 'filter.tbl_cube', 'filter.tbl_df',
## 'filter.ts', 'filter_.data.frame', 'filter_.tbl_cube', 'filter_.tbl_df',
## 'format.src_local', 'full_join.data.frame', 'full_join.tbl_df',
## 'group_by.data.frame', 'group_by.default', 'group_by.rowwise_df',
## 'group_by.tbl_cube', 'group_by_.data.frame', 'group_by_.rowwise_df',
## 'group_by_.tbl_cube', 'group_data.data.frame', 'group_data.grouped_df',
## 'group_data.rowwise_df', 'group_indices.data.frame',
## 'group_indices.default', 'group_indices.grouped_df',
## 'group_indices.rowwise_df', 'group_indices_.data.frame',
## 'group_indices_.grouped_df', 'group_indices_.rowwise_df',
## 'group_keys.data.frame', 'group_keys.grouped_df', 'group_keys.rowwise_df',
## 'group_map.function', 'group_map.data.frame', 'group_map.formula',
## 'group_map.grouped_df', 'group_nest.data.frame', 'group_nest.grouped_df',
## 'group_size.data.frame', 'group_size.grouped_df', 'group_size.rowwise_df',
## 'group_split.data.frame', 'group_split.grouped_df',
## 'group_split.rowwise_df', 'group_trim.data.frame',
'group_trim.grouped_df',
## 'group_vars.default', 'group_vars.grouped_df', 'group_vars.tbl_cube',
## 'group_walk.data.frame', 'group_walk.grouped_df', 'groups.data.frame',
## 'groups.grouped_df', 'groups.tbl_cube', 'hybrid_call.data.frame',
## 'inner_join.data.frame', 'inner_join.tbl_df', 'intersect.data.frame',
## 'intersect.default', 'left_join.data.frame', 'left_join.tbl_df',
## 'mutate.data.frame', 'mutate.default', 'mutate.tbl_df',
## 'mutate_.data.frame', 'mutate_.tbl_df', 'n_groups.data.frame',
## 'n_groups.grouped_df', 'n_groups.rowwise_df', 'nest_join.data.frame',
## 'nest_join.tbl_df', 'print.BoolResult', 'print.all_vars',
'print.any_vars',
## 'print.dplyr_sel_vars', 'print.fun_list', 'print.hybrid_call',
## 'print.location', 'print.rowwise_df', 'print.src', 'print.tbl_cube',
## 'pull.data.frame', 'rbind.grouped_df', 'recode.character',
## 'recode.factor', 'recode.numeric', 'rename.data.frame', 'rename.default',
## 'rename.grouped_df', 'rename.tbl_cube', 'rename_.data.frame',
## 'rename_.grouped_df', 'rename_.tbl_cube', 'right_join.data.frame',
## 'right_join.tbl_df', 'same_src.data.frame', 'same_src.tbl_cube',
## 'sample_frac.data.frame', 'sample_frac.default', 'sample_n.data.frame',
## 'sample_n.default', 'select.data.frame', 'select.default',
```

```
## 'select.grouped_df', 'select.tbl_cube', 'select_.data.frame',
## 'select_.grouped_df', 'select_.tbl_cube', 'semi_join.data.frame',
## 'semi_join.tbl_df', 'setdiff.data.frame', 'setdiff.default',
## 'setequal.data.frame', 'setequal.default', 'slice.data.frame',
## 'slice.default', 'slice.tbl_df', 'slice_.data.frame', 'slice_.tbl_df',
## 'src_tbls.src_local', 'summarise.data.frame', 'summarise.default',
## 'summarise.tbl_cube', 'summarise.tbl_df', 'summarise_.data.frame',
## 'summarise_.tbl_cube', 'summarise_.tbl_df', 'tbl.DBIConnection',
## 'tbl.src_local', 'tbl_sum.grouped_df', 'tbl_vars.data.frame',
## 'tbl_vars.tbl_cube', 'transmute.default', 'transmute.grouped_df',
## 'transmute_.default', 'transmute_.grouped_df', 'ungroup.data.frame',
## 'ungroup.grouped_df', 'ungroup.rowwise_df', 'union.data.frame',
## 'union.default', 'union_all.data.frame', 'union_all.default' were declared
## in NAMESPACE but not found
```

EnsurePackage("ggmap")

```
## Loading required package: ggmap
```

```
## Warning: S3 methods ' [.fun_list', ' [.grouped_df', 'all.equal.tbl_df',
## 'anti_join.data.frame', 'anti_join.tbl_df', 'arrange.data.frame',
## 'arrange.default', 'arrange.grouped_df', 'arrange.tbl_df',
## 'arrange_.data.frame', 'arrange_.tbl_df', 'as.data.frame.grouped_df',
## 'as.data.frame.rowwise_df', 'as.data.frame.tbl_cube', 'as.table.tbl_cube',
## 'as.tbl.data.frame', 'as.tbl.tbl', 'as.tbl_cube.array',
## 'as.tbl_cube.data.frame', 'as.tbl_cube.matrix', 'as.tbl_cube.table',
## 'as_tibble.grouped_df', 'as_tibble.tbl_cube', 'auto_copy.tbl_cube',
## 'auto_copy.tbl_df', 'cbind.grouped_df', 'collapse.data.frame',
## 'collect.data.frame', 'common_by.NULL', 'common_by.character',
## 'common_by.default', 'common_by.list', 'compute.data.frame',
## 'copy_to.DBIConnection', 'copy_to.src_local',
## 'default_missing.data.frame',
## 'default_missing.default', 'dim.tbl_cube', 'distinct.data.frame',
## 'distinct.default', 'distinct.grouped_df', 'distinct.tbl_df',
## 'distinct_.data.frame', 'distinct_.grouped_df', 'distinct_.tbl_df',
## 'do.NULL', 'do.data.frame', 'do.default', 'do.grouped_df',
## 'do.rowwise_df',
## 'do_.NULL', 'do_.data.frame', 'do_.grouped_df', 'do_.rowwise_df',
## 'filter.data.frame', 'filter.default', 'filter.tbl_cube', 'filter.tbl_df',
## 'filter.ts', 'filter_.data.frame', 'filter_.tbl_cube', 'filter_.tbl_df',
## 'format.src_local', 'full_join.data.frame', 'full_join.tbl_df',
## 'group_by.data.frame', 'group_by.default', 'group_by.rowwise_df',
## 'group_by.tbl_cube', 'group_by_.data.frame', 'group_by_.rowwise_df',
## 'group_by_.tbl_cube', 'group_data.data.frame', 'group_data.grouped_df',
## 'group_data.rowwise_df', 'group_indices.data.frame',
## 'group_indices.default', 'group_indices.grouped_df',
## 'group_indices.rowwise_df', 'group_indices_.data.frame',
## 'group_indices_.grouped_df', 'group_indices_.rowwise_df',
## 'group_keys.data.frame', 'group_keys.grouped_df', 'group_keys.rowwise_df',
## 'group_map.function', 'group_map.data.frame', 'group_map.formula',
```

```

## 'group_map.grouped_df', 'group_nest.data.frame', 'group_nest.grouped_df',
## 'group_size.data.frame', 'group_size.grouped_df', 'group_size.rowwise_df',
## 'group_split.data.frame', 'group_split.grouped_df',
## 'group_split.rowwise_df', 'group_trim.data.frame',
'group_trim.grouped_df',
## 'group_vars.default', 'group_vars.grouped_df', 'group_vars.tbl_cube',
## 'group_walk.data.frame', 'group_walk.grouped_df', 'groups.data.frame',
## 'groups.grouped_df', 'groups.tbl_cube', 'hybrid_call.data.frame',
## 'inner_join.data.frame', 'inner_join.tbl_df', 'intersect.data.frame',
## 'intersect.default', 'left_join.data.frame', 'left_join.tbl_df',
## 'mutate.data.frame', 'mutate.default', 'mutate.tbl_df',
## 'mutate_.data.frame', 'mutate_.tbl_df', 'n_groups.data.frame',
## 'n_groups.grouped_df', 'n_groups.rowwise_df', 'nest_join.data.frame',
## 'nest_join.tbl_df', 'print.BoolResult', 'print.all_vars',
'print.any_vars',
## 'print.dplyr_sel_vars', 'print.fun_list', 'print.hybrid_call',
## 'print.location', 'print.rowwise_df', 'print.src', 'print.tbl_cube',
## 'pull.data.frame', 'rbind.grouped_df', 'recode.character',
## 'recode.factor', 'recode.numeric', 'rename.data.frame', 'rename.default',
## 'rename.grouped_df', 'rename.tbl_cube', 'rename_.data.frame',
## 'rename_.grouped_df', 'rename_.tbl_cube', 'right_join.data.frame',
## 'right_join.tbl_df', 'same_src.data.frame', 'same_src.tbl_cube',
## 'sample_frac.data.frame', 'sample_frac.default', 'sample_n.data.frame',
## 'sample_n.default', 'select.data.frame', 'select.default',
## 'select.grouped_df', 'select.tbl_cube', 'select_.data.frame',
## 'select_.grouped_df', 'select_.tbl_cube', 'semi_join.data.frame',
## 'semi_join.tbl_df', 'setdiff.data.frame', 'setdiff.default',
## 'setequal.data.frame', 'setequal.default', 'slice.data.frame',
## 'slice.default', 'slice.tbl_df', 'slice_.data.frame', 'slice_.tbl_df',
## 'src_tbls.src_local', 'summarise.data.frame', 'summarise.default',
## 'summarise.tbl_cube', 'summarise.tbl_df', 'summarise_.data.frame',
## 'summarise_.tbl_cube', 'summarise_.tbl_df', 'tbl.DBIConnection',
## 'tbl.src_local', 'tbl_sum.grouped_df', 'tbl_vars.data.frame',
## 'tbl_vars.tbl_cube', 'transmute.default', 'transmute.grouped_df',
## 'transmute_.default', 'transmute_.grouped_df', 'ungroup.data.frame',
## 'ungroup.grouped_df', 'ungroup.rowwise_df', 'union.data.frame',
## 'union.default', 'union_all.data.frame', 'union_all.default' were declared
## in NAMESPACE but not found

## Error: package or namespace load failed for 'ggmap' in library.dynam(lib,
package, package.lib):
## DLL 'dplyr' not found: maybe not installed for this architecture?

## package 'ggmap' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
## C:\Users\dbLewis\AppData\Local\Temp\RtmpWA9FEw\downloaded_packages

## Loading required package: ggmap

```

```

## Warning: S3 methods '[.fun_list', '[.grouped_df', 'all.equal.tbl_df',
## 'anti_join.data.frame', 'anti_join.tbl_df', 'arrange.data.frame',
## 'arrange.default', 'arrange.grouped_df', 'arrange.tbl_df',
## 'arrange_.data.frame', 'arrange_.tbl_df', 'as.data.frame.grouped_df',
## 'as.data.frame.rowwise_df', 'as.data.frame.tbl_cube', 'as.table.tbl_cube',
## 'as.tbl.data.frame', 'as.tbl.tbl', 'as.tbl_cube.array',
## 'as.tbl_cube.data.frame', 'as.tbl_cube.matrix', 'as.tbl_cube.table',
## 'as_tibble.grouped_df', 'as_tibble.tbl_cube', 'auto_copy.tbl_cube',
## 'auto_copy.tbl_df', 'cbind.grouped_df', 'collapse.data.frame',
## 'collect.data.frame', 'common_by.NULL', 'common_by.character',
## 'common_by.default', 'common_by.list', 'compute.data.frame',
## 'copy_to.DBIConnection', 'copy_to.src_local',
'default_missing.data.frame',
## 'default_missing.default', 'dim.tbl_cube', 'distinct.data.frame',
## 'distinct.default', 'distinct.grouped_df', 'distinct.tbl_df',
## 'distinct_.data.frame', 'distinct_.grouped_df', 'distinct_.tbl_df',
## 'do.NULL', 'do.data.frame', 'do.default', 'do.grouped_df',
'do.rowwise_df',
## 'do_.NULL', 'do_.data.frame', 'do_.grouped_df', 'do_.rowwise_df',
## 'filter.data.frame', 'filter.default', 'filter.tbl_cube', 'filter.tbl_df',
## 'filter.ts', 'filter_.data.frame', 'filter_.tbl_cube', 'filter_.tbl_df',
## 'format.src_local', 'full_join.data.frame', 'full_join.tbl_df',
## 'group_by.data.frame', 'group_by.default', 'group_by.rowwise_df',
## 'group_by.tbl_cube', 'group_by_.data.frame', 'group_by_.rowwise_df',
## 'group_by_.tbl_cube', 'group_data.data.frame', 'group_data.grouped_df',
## 'group_data.rowwise_df', 'group_indices.data.frame',
## 'group_indices.default', 'group_indices.grouped_df',
## 'group_indices.rowwise_df', 'group_indices_.data.frame',
## 'group_indices_.grouped_df', 'group_indices_.rowwise_df',
## 'group_keys.data.frame', 'group_keys.grouped_df', 'group_keys.rowwise_df',
## 'group_map.function', 'group_map.data.frame', 'group_map.formula',
## 'group_map.grouped_df', 'group_nest.data.frame', 'group_nest.grouped_df',
## 'group_size.data.frame', 'group_size.grouped_df', 'group_size.rowwise_df',
## 'group_split.data.frame', 'group_split.grouped_df',
## 'group_split.rowwise_df', 'group_trim.data.frame',
'group_trim.grouped_df',
## 'group_vars.default', 'group_vars.grouped_df', 'group_vars.tbl_cube',
## 'group_walk.data.frame', 'group_walk.grouped_df', 'groups.data.frame',
## 'groups.grouped_df', 'groups.tbl_cube', 'hybrid_call.data.frame',
## 'inner_join.data.frame', 'inner_join.tbl_df', 'intersect.data.frame',
## 'intersect.default', 'left_join.data.frame', 'left_join.tbl_df',
## 'mutate.data.frame', 'mutate.default', 'mutate.tbl_df',
## 'mutate_.data.frame', 'mutate_.tbl_df', 'n_groups.data.frame',
## 'n_groups.grouped_df', 'n_groups.rowwise_df', 'nest_join.data.frame',
## 'nest_join.tbl_df', 'print.BoolResult', 'print.all_vars',
'print.any_vars',
## 'print.dplyr_sel_vars', 'print.fun_list', 'print.hybrid_call',
## 'print.location', 'print.rowwise_df', 'print.src', 'print.tbl_cube',
## 'pull.data.frame', 'rbind.grouped_df', 'recode.character',
## 'recode.factor', 'recode.numeric', 'rename.data.frame', 'rename.default',

```

```
## 'rename.grouped_df', 'rename.tbl_cube', 'rename_.data.frame',
## 'rename_.grouped_df', 'rename_.tbl_cube', 'right_join.data.frame',
## 'right_join.tbl_df', 'same_src.data.frame', 'same_src.tbl_cube',
## 'sample_frac.data.frame', 'sample_frac.default', 'sample_n.data.frame',
## 'sample_n.default', 'select.data.frame', 'select.default',
## 'select.grouped_df', 'select.tbl_cube', 'select_.data.frame',
## 'select_.grouped_df', 'select_.tbl_cube', 'semi_join.data.frame',
## 'semi_join.tbl_df', 'setdiff.data.frame', 'setdiff.default',
## 'setequal.data.frame', 'setequal.default', 'slice.data.frame',
## 'slice.default', 'slice.tbl_df', 'slice_.data.frame', 'slice_.tbl_df',
## 'src_tbls.src_local', 'summarise.data.frame', 'summarise.default',
## 'summarise.tbl_cube', 'summarise.tbl_df', 'summarise_.data.frame',
## 'summarise_.tbl_cube', 'summarise_.tbl_df', 'tbl.DBIConnection',
## 'tbl.src_local', 'tbl_sum.grouped_df', 'tbl_vars.data.frame',
## 'tbl_vars.tbl_cube', 'transmute.default', 'transmute.grouped_df',
## 'transmute_.default', 'transmute_.grouped_df', 'ungroup.data.frame',
## 'ungroup.grouped_df', 'ungroup.rowwise_df', 'union.data.frame',
## 'union.default', 'union_all.data.frame', 'union_all.default' were declared
## in NAMESPACE but not found
```

```
## Error: package or namespace load failed for 'ggmap' in library.dynam(lib,
package, package.lib):
```

```
## DLL 'dplyr' not found: maybe not installed for this architecture?
```

```
EnsurePackage("RJSONIO")
```

```
## Loading required package: RJSONIO
```

```
EnsurePackage("RCurl")
```

```
## Loading required package: RCurl
```

```
## Loading required package: bitops
```

```
EnsurePackage("reshape2")
```

```
## Loading required package: reshape2
```

```
EnsurePackage("sqldf")
```

```
## Loading required package: sqldf
```

```
## Loading required package: gsubfn
```

```
## Loading required package: proto
```

```
## Loading required package: RSQLite
```

```
EnsurePackage("tidyr")
```

```
## Loading required package: tidyr
```

```
##
## Attaching package: 'tidyr'

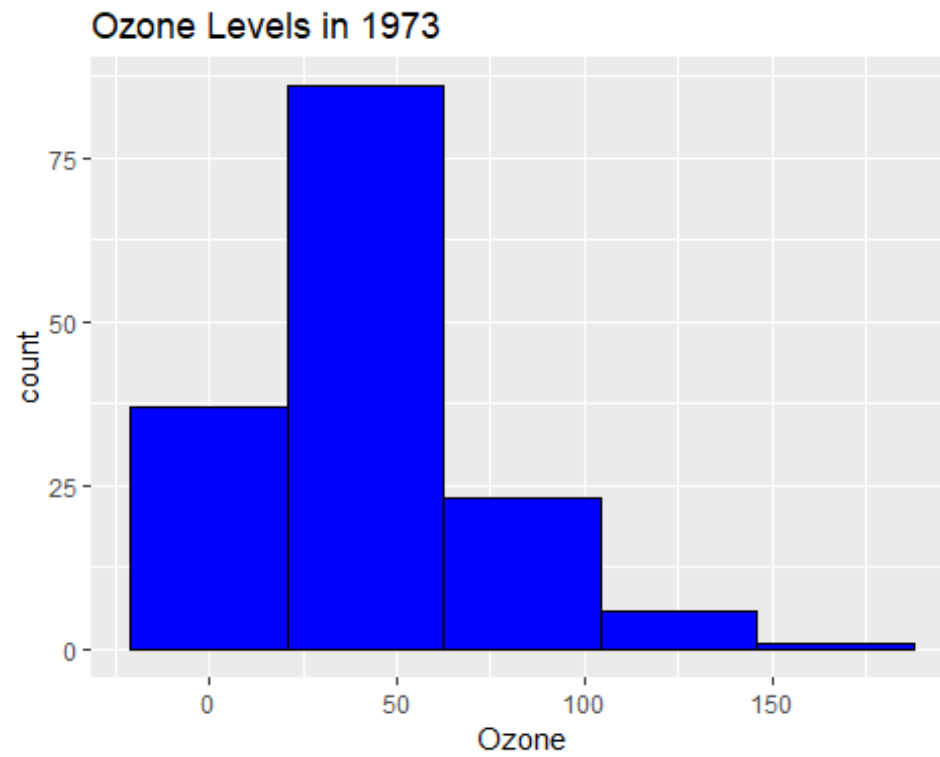
## The following object is masked from 'package:reshape2':
##
##      smiths

## The following object is masked from 'package:RCurl':
##
##      complete

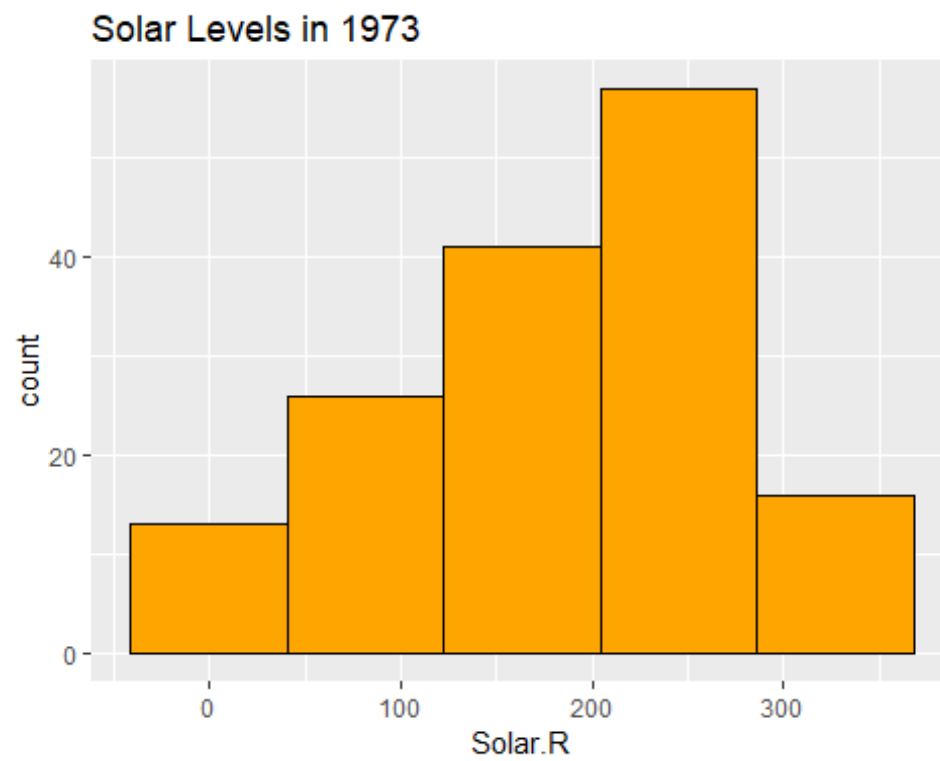
#
#####
#####Problems Solved#####
#####
#
#
#Step 1: Load the data
#We will use the air quality data set, which you should already have as part
of your R installation.
#
myAirquality<-airquality
#
#Step 2: Clean the data
#After you load the data, there will be some NAs in the data. You need to
figure out what to do
#about those nasty NAs.
#
colnames(myAirquality)[colSums(is.na(myAirquality))>0]

## [1] "Ozone"    "Solar.R"

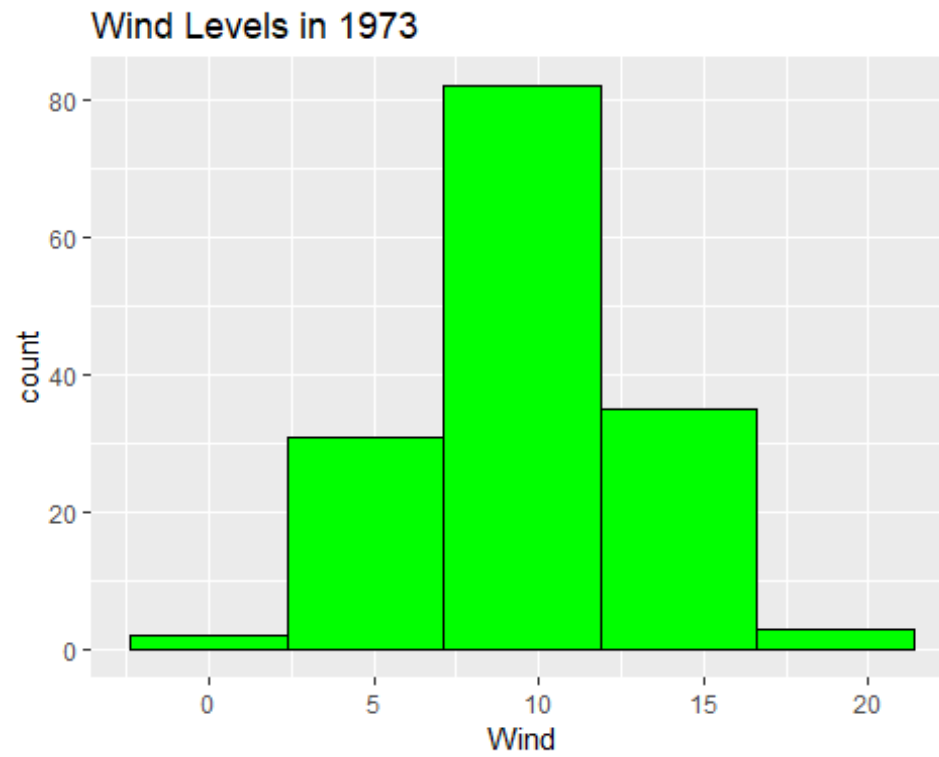
myAirquality$Ozone[is.na(myAirquality$Ozone)] <- mean(myAirquality$Ozone,
na.rm=TRUE)
myAirquality$Solar.R[is.na(myAirquality$Solar.R)] <-
mean(myAirquality$Solar.R, na.rm=TRUE)
#
#Step 3: Understand the data distribution
#Create the following visualizations using ggplot:
#.Histograms for each of the variables
#
ozone_hist<- ggplot(myAirquality, aes(x=Ozone)) + geom_histogram(bins=5,
color="black", fill="blue")+ggtitle("Ozone Levels in 1973")
solar_hist<- ggplot(myAirquality, aes(x=Solar.R)) + geom_histogram(bins=5,
color="black", fill="orange")+ggtitle("Solar Levels in 1973")
wind_hist<- ggplot(myAirquality, aes(x=Wind)) + geom_histogram(bins=5,
color="black", fill="green")+ggtitle("Wind Levels in 1973")
temp_hist<- ggplot(myAirquality, aes(x=Temp)) + geom_histogram(bins=5,
color="black", fill="purple")+ggtitle("Temp Levels in 1973")
ozone_hist
```



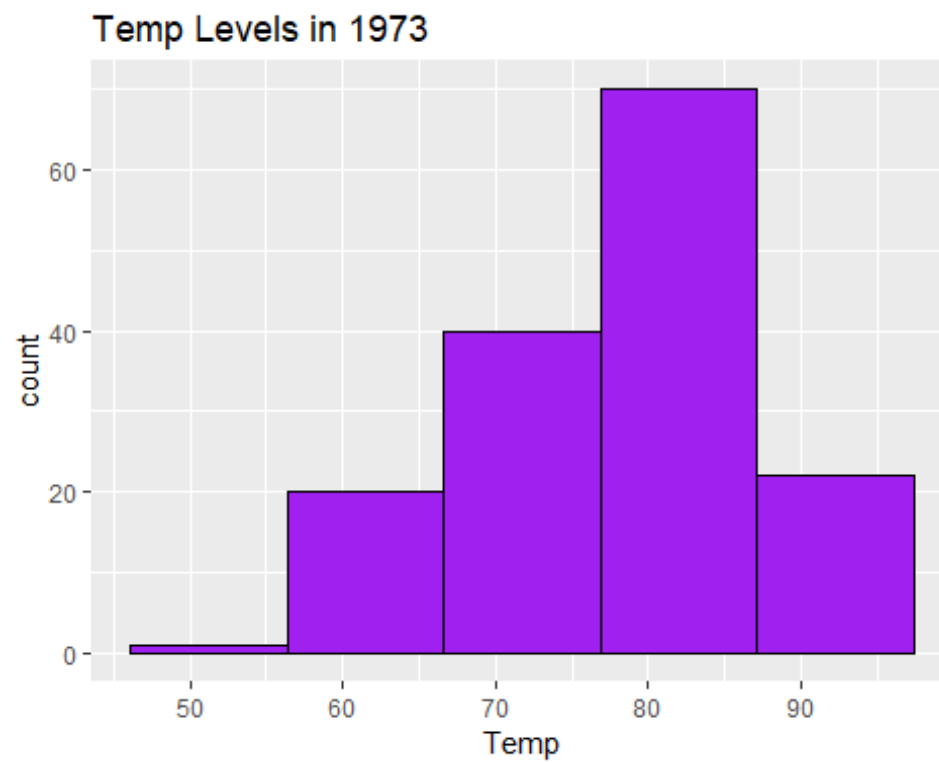
solar_hist



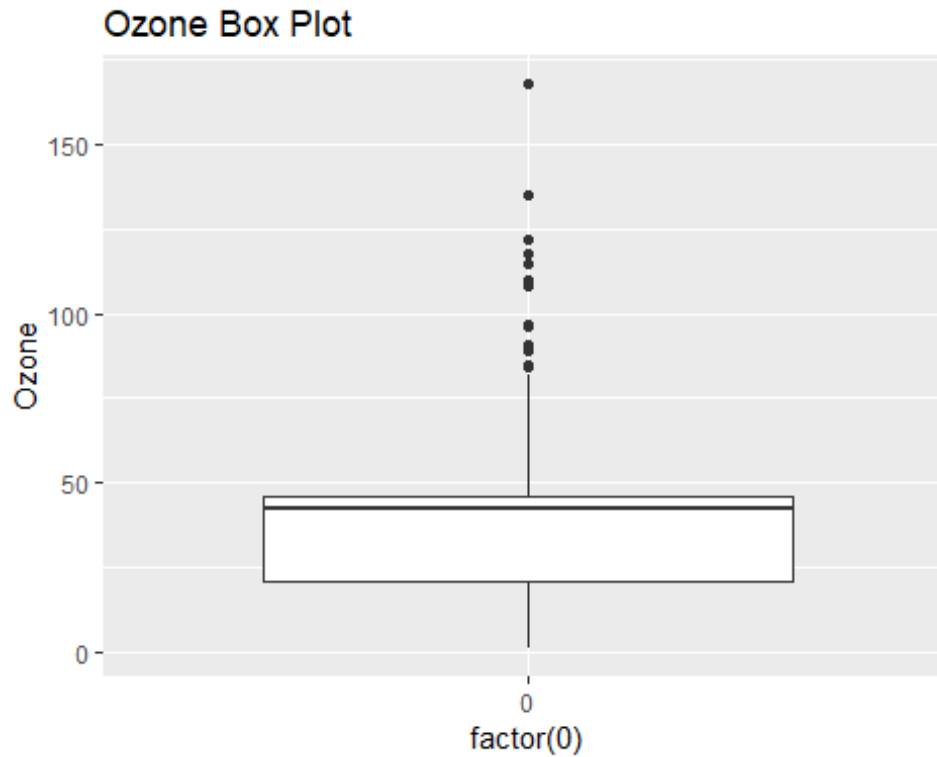
wind_hist



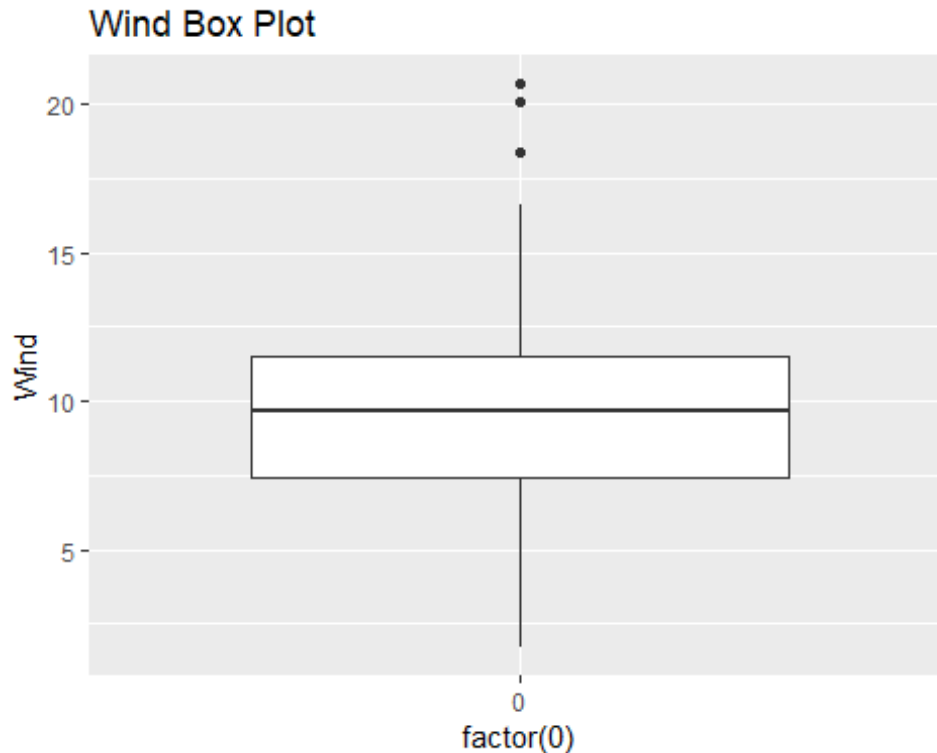
temp_hist



```
#
#Boxplot for Ozone
#
ozone_box<- ggplot(myAirquality, aes(x=factor(0),Ozone)) +geom_boxplot() +
ggtitle("Ozone Box Plot")
ozone_box
```

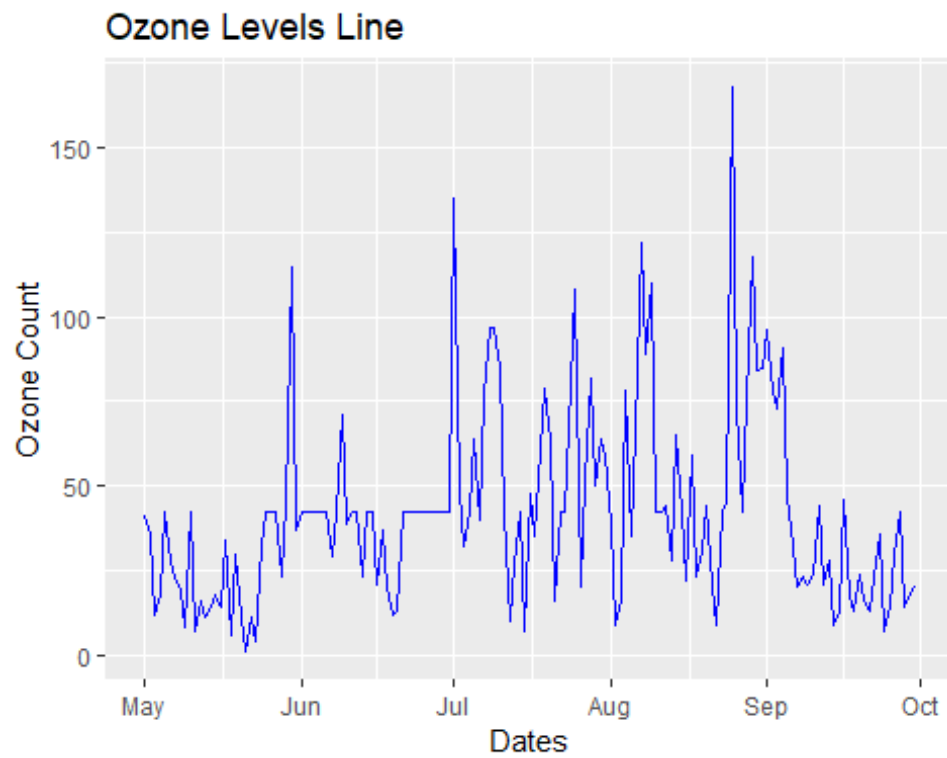


```
#
#Boxplot for wind values (round the wind to get a good number of "buckets")
#
wind_box<- ggplot(myAirquality, aes(x=factor(0),Wind)) +geom_boxplot() +
ggtitle("Wind Box Plot")
wind_box
```

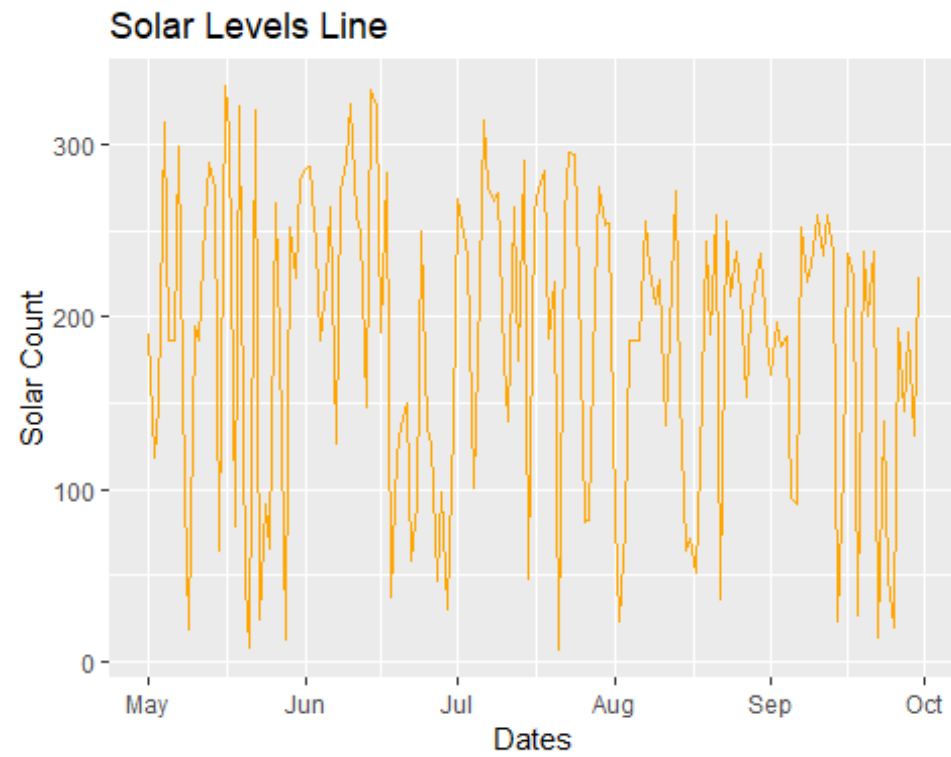


```
#
#Step 3: Explore how the data changes over time
#First, make sure to create appropriate dates (this data was from 1973). Then
create line charts
#for ozone, temp, wind and solar.R(one line chart for each, and then one
chart with 4 lines,
#each having a different color).
#
myAirquality$date<-
as.Date(Numberize(c(paste(myAirquality$Month,"/",myAirquality$Day,"/",
"73"))), format = "%m/%d/%y")
ozone_line<-ggplot(data=myAirquality,aes(x=date, y=Ozone)) +
geom_line(color="blue") + ggtitle("Ozone Levels Line") + labs(x="Dates",
y="Ozone Count")
solar_line<-ggplot(data=myAirquality,aes(x=date, y=Solar.R)) +
geom_line(color="orange") + ggtitle("Solar Levels Line") + labs(x="Dates",
y="Solar Count")
wind_line<-ggplot(data=myAirquality,aes(x=date, y=Wind)) +
geom_line(color="green") + ggtitle("wind Levels Line") + labs(x="Dates",
y="Wind Count")
temp_line<-ggplot(data=myAirquality,aes(x=date, y=Temp)) +
geom_line(color="purple") + ggtitle("Temp Levels Line") + labs(x="Dates",
y="Temp Count")
all_line<-ggplot(data=myAirquality, aes(date)) +
geom_line(aes(y=Ozone),color="blue") +
geom_line(aes(y=Solar.R),color="orange") +
geom_line(aes(y=Wind),color="green") + geom_line(aes(y=Temp), color="purple")
```

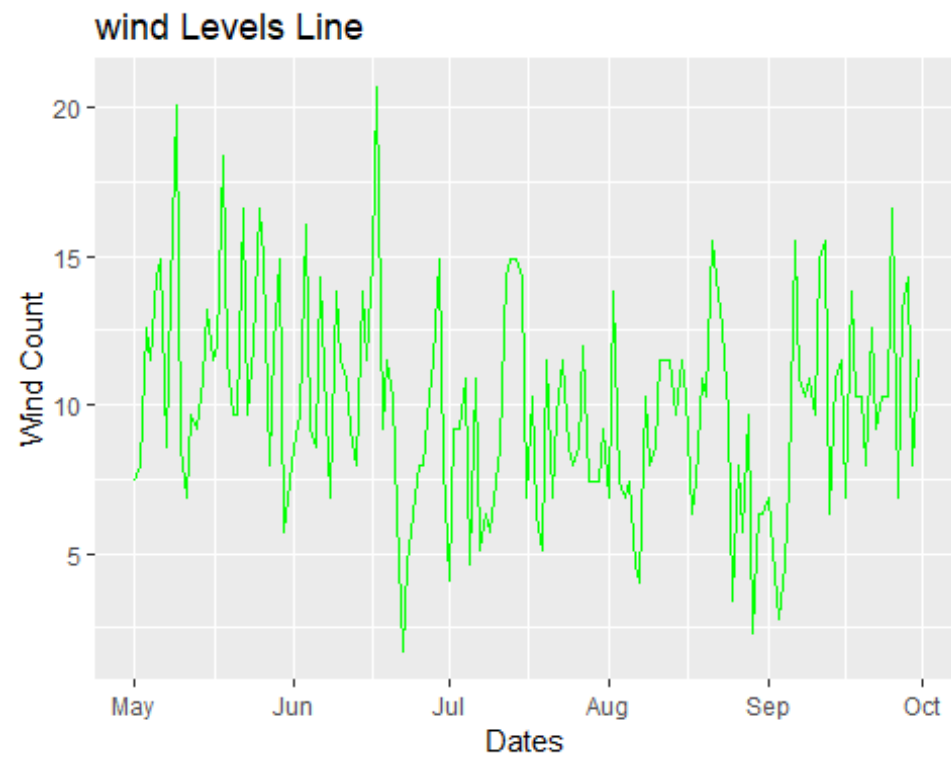
```
all_line<-all_line + ggtitle("Ozone, Solar, Wind, and Temp Levels 1973")
all_line<-all_line + labs(x="Dates", y="Ozone/Solar/Wind/Temp")
ozone_line
```



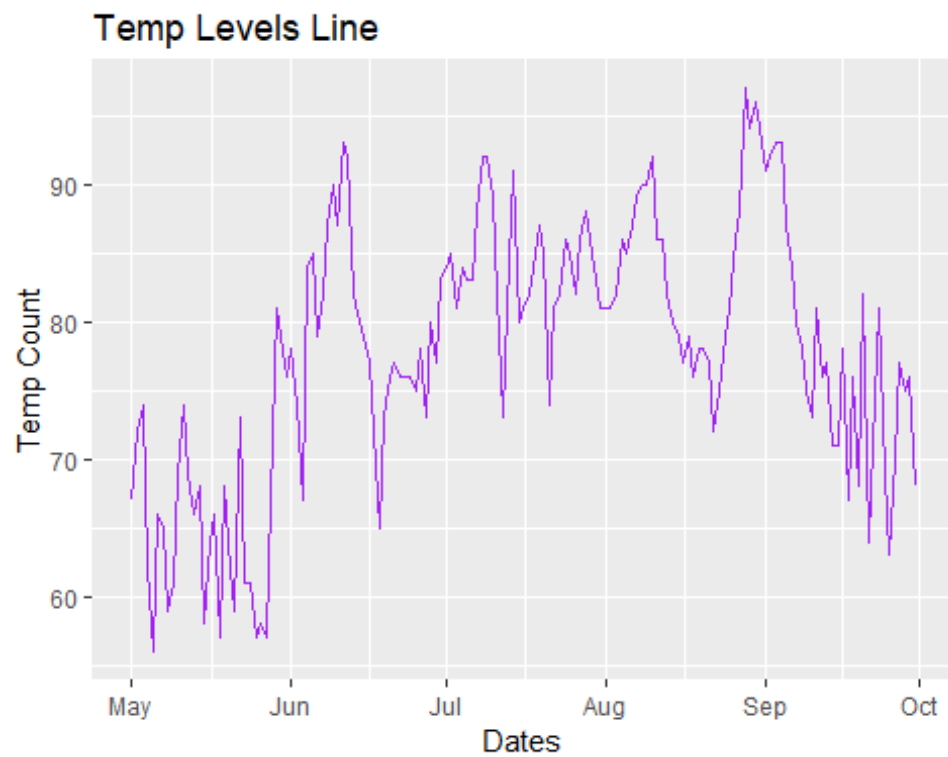
```
solar_line
```



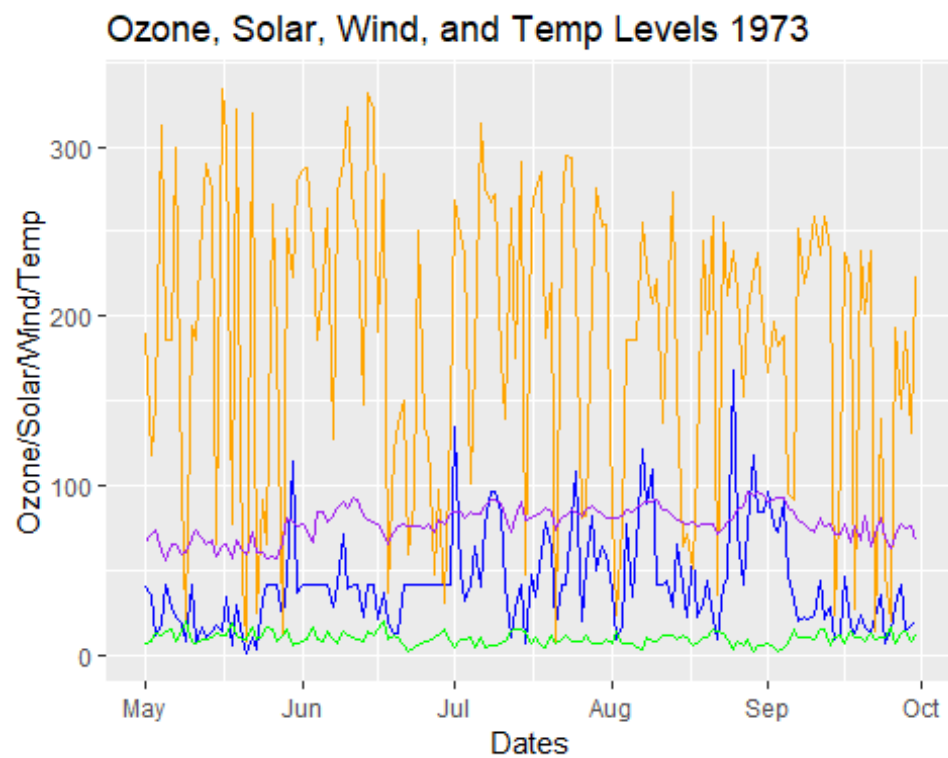
wind_line



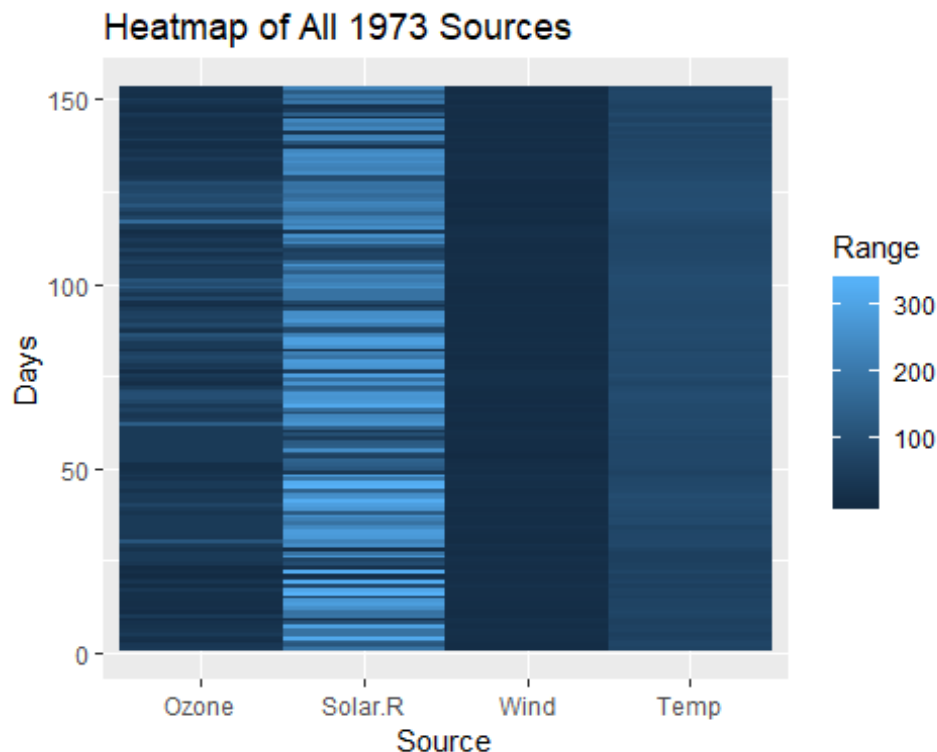
temp_line



all_line

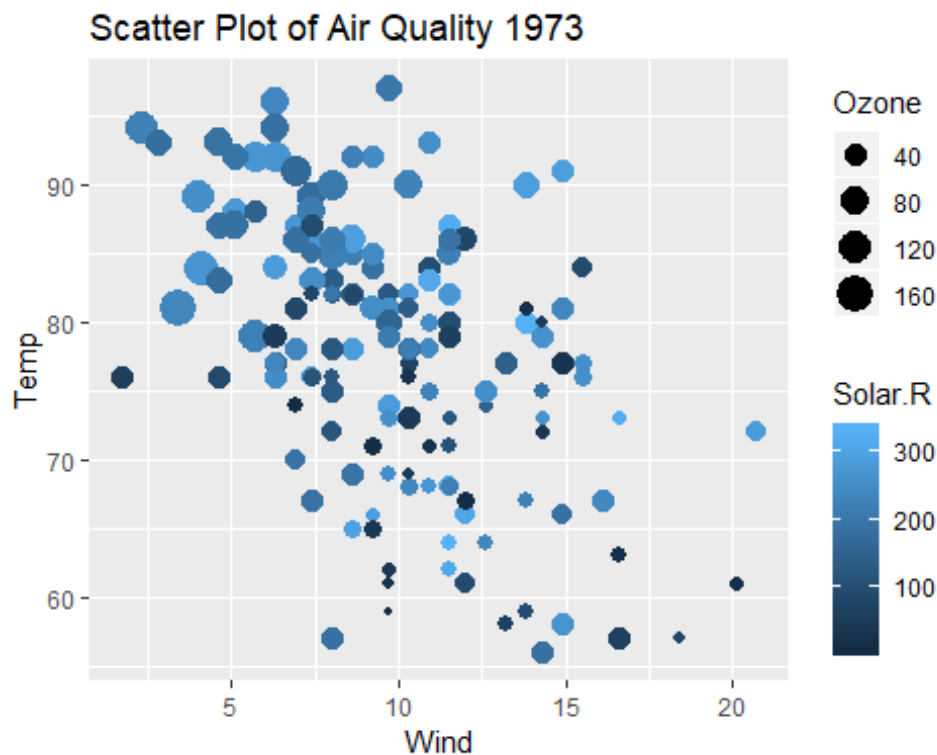


```
#
#Create these visualizations using ggplot.Note that for the chart with 4
lines, you need to think
#about how to effectively use the y-axis.
#
#Step 4: Look at all thedata via a Heatmap
#Create a heatmap, with each day along the x-axis and ozone, temp, wind and
solar.r along the y-axis,
#and days as rows along the y-axis. Great the heatmap using geom_tile(this
defines the ggplotgeometry
#to be 'tiles' as opposed to 'lines' and the other geometry we have
previously used).
#Note that you need to figure out how to show the relative change equally
across all the variables.
#
melted_data <- melt(as.matrix(myAirquality[,1:4]))
names(melted_data) <- c('Days', 'Source', 'Range')
all_heatmap<-ggplot(data=melted_data, aes(y=Days, x=Source, fill=Range)) +
geom_tile() + ggtitle("Heatmap of All 1973 Sources")
all_heatmap
```



```
#
#Step 5: Look at all the data via a scatter chart
#Create a scatter chart(using ggplot geom_point), with the x-axis
representing the wind, the y-axis
#representing the temperature, the size of each dot representing the ozone
and the color representing
```

```
#the solar.
#
scatterplot <- ggplot(data=myAirquality, aes(x=Wind, y=Temp, size=Ozone,
color=Solar.R)) + geom_point() + ggtitle("Scatter Plot of Air Quality 1973")
scatterplot
```



```
#
#RStep 6: Final Analysis
#Do you see any patterns after exploring the data?
#
print("I found that the data sets were all roughly displayed the same. When
you brought
    the Solar measurements into the mix it would throw off the whole scale
on the y axis.")

## [1] "I found that the data sets were all roughly displayed the same. When
you brought \n    the Solar measurements into the mix it would throw off
the whole scale on the y axis."

#
#What was the most useful visualization?
#
print("I believe the most useful of these metrics to me would have been the
line graph.
    This is because you could easily just take this infomraiton and compare
each day to each other.")
```



```
## [1] "I believe the most useful of these metrics to me would have been the  
line graph.\n      This is because you could easily just take this  
infomraiton and compare each day to each other."
```

```
#
```

```
#
```

```
#END OF SCRIPT
```