

Omkar Gurav

TE IT Batch & T3

Roll No:- 8048

## Assignment-7

- Aim:- Visualize the data using R by plotting the graphs for assignment Air Quality Data set and Facebook Data Set. (Scatter plot, bar plot, box plot, pie chart, line plot)

### ● Theory:-

Q.1. What is package in R? Explain in detail. (including command to install and load)

→ R packages are a collection of R Functions, compiled code and sample data. They are stored under a directory called 'library' in the environment.

By default, R installs a set of packages during installation. More packages are added later, when they are needed for some specific purpose.

When we start the R console, only the default packages are available by default. Other packages which are already installed have to be loaded explicitly to be ~~loaded explicitly~~ used by the R program that is going to use them.



Get the list of all the packages installed using `library()`.

Get all packages currently loaded in the R environment using `search()`.

Install new package:-

There are 2 ways to add new R packages.

① Install directly from CRAN:-

The following command gets the packages directly from CRAN webpage and installs the package in the R environment.

`install.packages("PackageName")`  
eg: `install.packages("ggplot2")`

② Install package manually:-

Go to the link of R Packages to download the package needed. Save the package as a .zip file in suitable location in the local system. Now we can run `install.packages()` command to install this package in the R environment.



`install.packages(file_name_with_path, repos=NULL, type="source")`

eg: `install.packages("E:/XML-3.98-1.3.zip", repos=NULL, type="source")`

### Load Package :-

Before a package can be used in the code, it must be loaded to the current R environment.

We also need to load a package that is already installed previously but not available in the current environment.

A package is loaded using following command -

`library("package Name")`

eg: `library("dplyr")`

Q.2. Explain following the ggplot2 package.

→ ggplot2 is an open-source data visualization package for the statistical programming language R.

It is created by Hadley Wickham in 2005, ggplot2 is an



implementation of Wilkinson's Grammar of Graphics - a general scheme for data visualization which breaks up graphs into semantic components such as scales & layers.

ggplot2 can serve as a replacement for the base graphics in R & contains a number of defaults for web & print display of common scales.

ggplot2 allows to create graphs that represent both univariate & multivariate numerical & categorical data in straightforward manner. Grouping can be represented by color, symbol, size & transparency.

There is helper function called `qplot()` (for quick plot) that can hide much of complexity when creating standard graphs.

Q.3. Explain following functions in R

a. `aes()`

→ `aes()` creates a list of unevaluated expressions.

This function also performs partial name matching, converts color to color & old style R names to ggplot names.



Aesthetic mappings describe how variables in the data are mapped to visual properties of geoms. Aesthetic mappings can be set in `ggplot()` & in individual layers.

Syntax: `aes(x, y, ...)`

`x, y, ...` : List of name-value pairs in the form  
aesthetic = variable describing which variable in the layer data should be mapped to which aesthetics used by paired geom/stat.

eg: `aes(x = mpg, y = wt)`

b. `geom-boxplot()`

→ The boxplot compactly displays the distribution of a continuous variable. It visualises five summary statistics (median, 2 hinges & 2 whiskers) and all "outlying" points individually.

Syntax: `geom-boxplot(mapping = NULL, data = NULL, ...)`

Eg: mapping: set of aesthetic mappings created by `aes()`

data: The data to be displayed in this layer.

Eg: `ggplot(mpg, aes(class, hwy)) + geom-boxplot(fill = "white")`



### c. `geom_point()`

→ A geom that draws a point defined by an x and y coordinate. It is used to draw scatterplot.

Syntax: `geom_point(x, y, ...)` (`mapping = NULL, data = NULL, ...`)

~~x, y~~ Mapping: set of aesthetic mapping created by `aes()`

data: The data to be displayed in this layer

Eg: `ggplot(mtcars, aes(wt, mpg)) + geom_point(size=3)`

### d. `geom_bar()`

→ `geom_bar()` makes the height of the bar proportional to the number of cases in each group (or if weight aesthetic is supplied, sum of the weights).

`geom_bar()` uses `stat_count()` by default; it counts the number of cases at each x position.

Syntax: `geom_bar(mapping = NULL, data = NULL, ...)`

mapping: set of aesthetic mapping created by `aes()`

data: The data to be displayed in this layer

Eg: `ggplot(mpg, aes(class)) + geom_bar(aes(weight = displ))`

- Conclusion:- Thus we have successfully visualized data using R by plotting graphs for assignment air quality data set & Facebook data set.

# plot.R

```
# Facebook Dataset
```

```
File <- "C:/Users/DELL/Downloads/Facebook_metrics/dataset_Facebook.csv"
```

```
if(! file.exists(File)){
```

```
  zipf <- "C:/Users/DELL/Downloads/Facebook_metrics.zip"
```

```
  OutDir <- "C:/Users/DELL/Downloads/Facebook_metrics"
```

```
  unzip(zipf, exdir = OutDir)
```

```
}
```

```
Facebook_data <- read.csv("C:/Users/DELL/Downloads/Facebook_metrics/dataset_Facebook.csv", sep = ";")
```

```
library("ggplot2")
```

```
# Scatterplot
```

```
facebook_scatterplot <- ggplot(Facebook_data, aes(x = Lifetime.Post.Total.Reach, y = Lifetime.Post.Total.Impressions,  
                                                    color = Type)) +  
  geom_point(size = 3)
```

```
facebook_scatterplot
```

```
# Barplot
```

```
facebook_barplot <- ggplot(Facebook_data, aes(x = Category, fill = Type)) + geom_bar(width = 0.25)  
facebook_barplot
```

```
# Boxplot
```

```
facebook_boxplot <- ggplot(Facebook_data, aes(x = share, y = like, fill = Type)) +  
  geom_boxplot() + coord_cartesian(ylim = c(0, 1000))  
facebook_boxplot
```

```
# Piechart
```

```
ggplot(Facebook_data, aes(x = "", y = comment, fill = Type)) +  
geom_bar(stat = "identity", width = 0.2 ) + coord_polar("y")
```

```
# Lineplot
```

```
facebook_lineplot <- ggplot(Facebook_data, aes(x = Lifetime.Post.Total.Reach,  
y = Lifetime.Post.Total.Impressions,  
color = Type)) + geom_line(size = 2)
```

```
facebook_lineplot
```

```
# airquality Dataset
```

```
data("airquality")
```

```
my_airquality_data <- airquality
```

```
my_airquality_data$Month <- month.abb[my_airquality_data$Month]
```

```
# Scatterplot
```

```
airquality_scatterplot <- ggplot(my_airquality_data, aes(x = Solar.R, y = Ozone,  
color = Month)) + geom_point(size = 3)
```

```
airquality_scatterplot
```

```
# Barplot
```

```
airquality_barplot <- ggplot(my_airquality_data, aes(x = Temp, fill = Month)) + geom_bar(width = 0.35)
```

```
airquality_barplot
```

```
# Boxplot
```

```
airquality_boxplot <- ggplot(my_airquality_data, aes(x = Temp , y = Ozone, fill = Month)) +  
geom_boxplot()
```

```
airquality_boxplot
```

```
# Piechart
```

```
ggplot(my_airquality_data, aes(x = "", y = Ozone, fill = Month)) +  
geom_bar(stat = "identity", width = 0.2 ) + coord_polar("y")
```

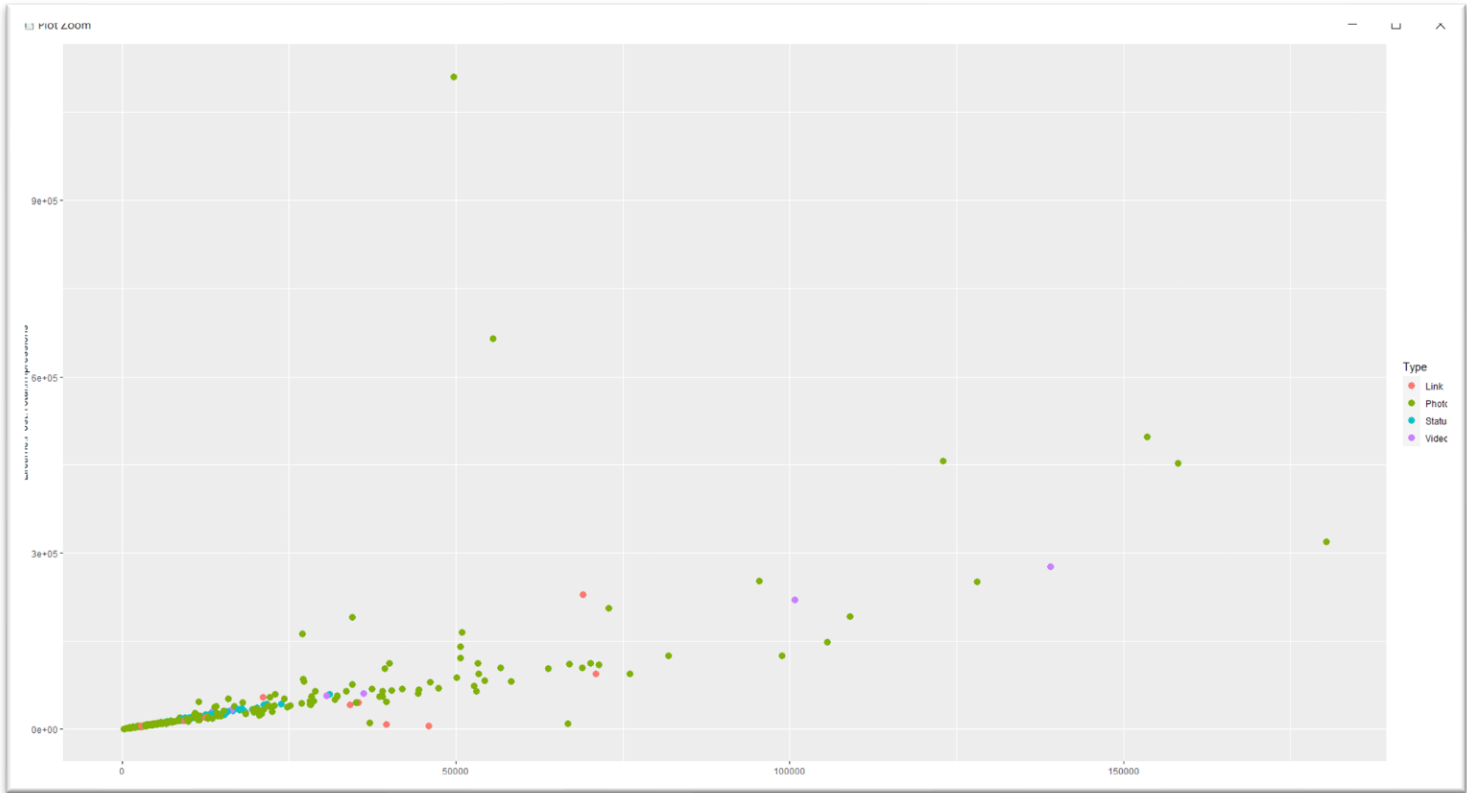
```
# Lineplot
```



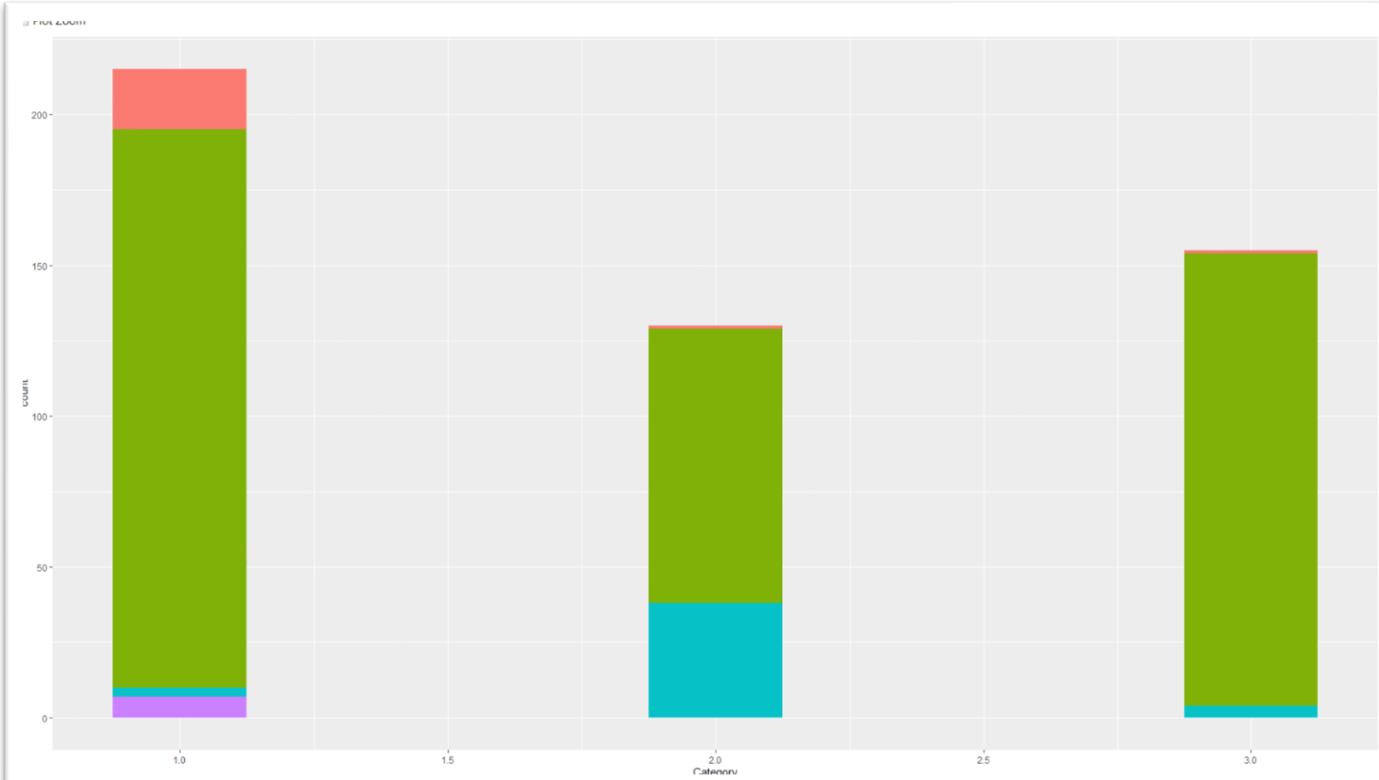
```
airquality_lineplot <- ggplot(my_airquality_data, aes(y = Wind,  
x = Ozone, color = Month)) + geom_line(size = 2)  
  
airquality_lineplot
```

Output :

```
> File <- "C:/Users/DELL/Downloads/Facebook_metrics/dataset_Facebook.csv"  
> if(! file.exists(File)){  
+  
+   zipf <- "C:/Users/DELL/Downloads/Facebook_metrics.zip"  
+  
+   OutDir <- "C:/Users/DELL/Downloads/Facebook_metrics"  
+  
+  
+   unzip(zipf, exdir = OutDir)  
+  
+ }  
  
> Facebook_data <- read.csv("C:/Users/DELL/Downloads/Facebook_metrics/dataset_Facebook.csv", sep = ";")  
> library("ggplot2")  
> # Scatterplot  
> facebook_scatterplot <- ggplot(Facebook_data, aes(x = Lifetime.Post.Total.Reach, y = Lifetime.Post.Total.Impressions,  
+   color = Type)) +  
+   geom_point(size = 3)  
  
> facebook_scatterplot
```

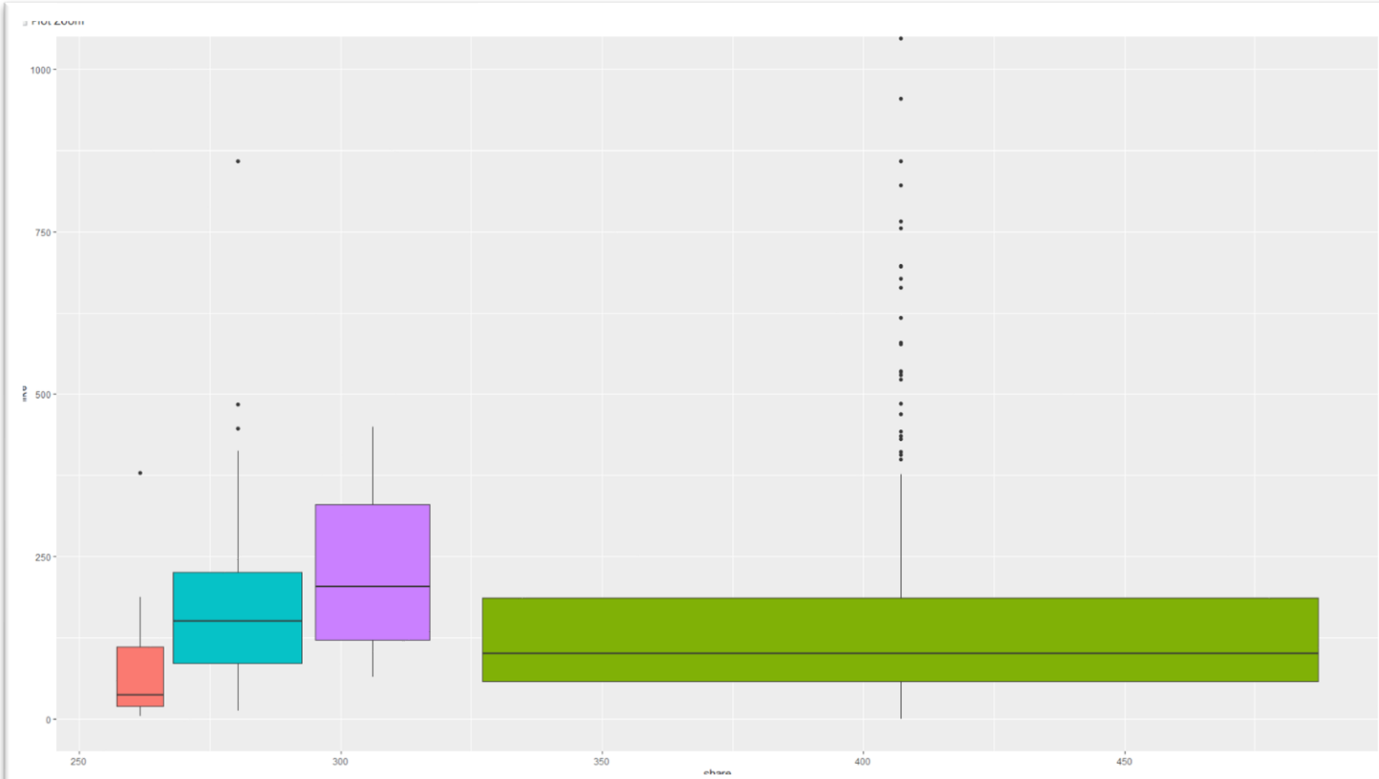


```
> # Barplot
> facebook_barplot <- ggplot(Facebook_data, aes(x = Category, fill = Type)) + geom_bar(width = 0.25)
> facebook_barplot
```



```
> # Boxplot
> facebook_boxplot <- ggplot(Facebook_data, aes(x = share, y = like, fill = Type)) +
+   geom_boxplot() + coord_cartesian(ylim = c(0, 1000))
> facebook_boxplot
```

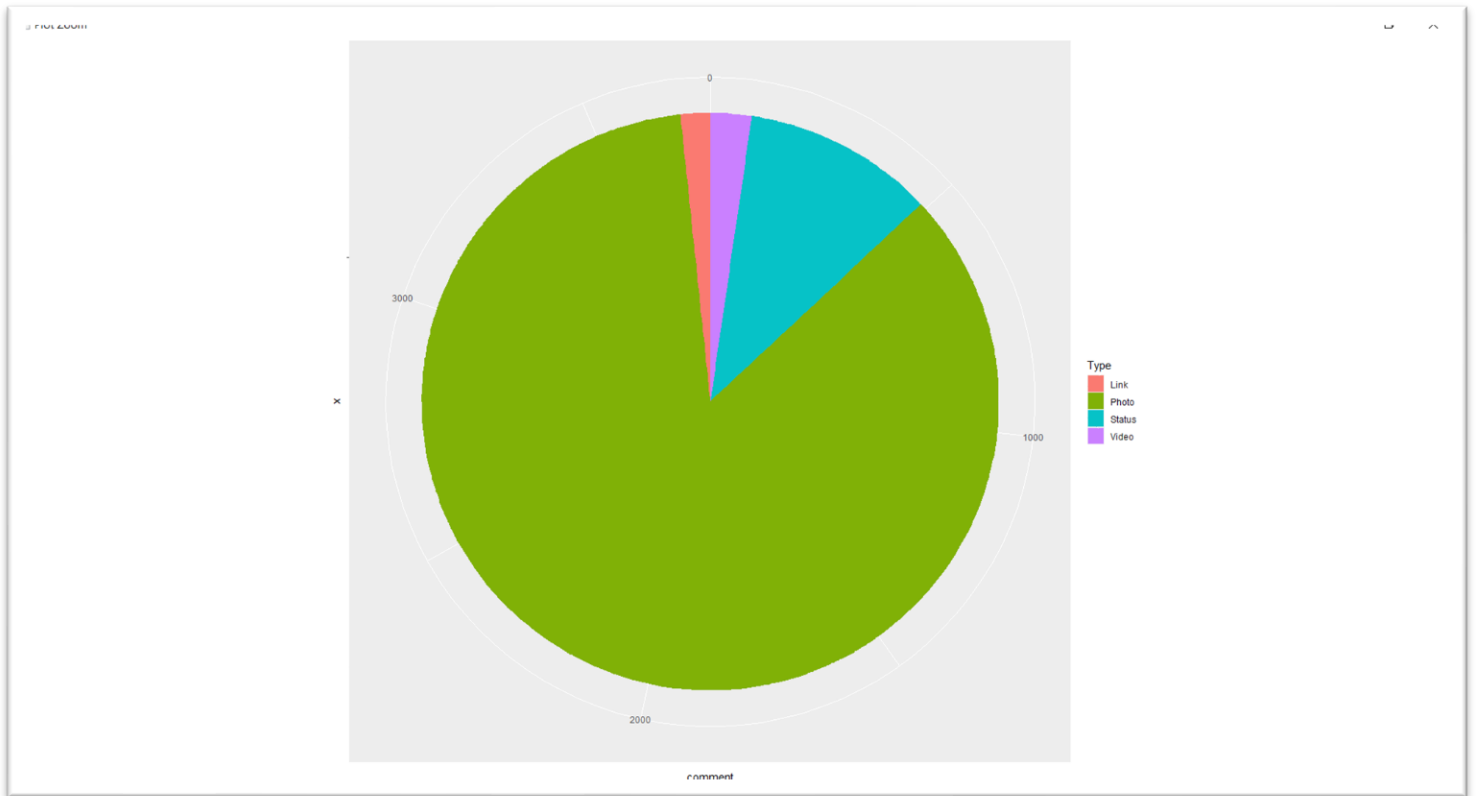
Warning message:  
Removed 4 rows containing missing values (stat\_boxplot).



```
> # Piechart
> ggplot(Facebook_data, aes(x = "", y = comment, fill = Type)) +
```



```
+ geom_bar(stat = "identity", width = 0.2 ) + coord_polar("y")
```



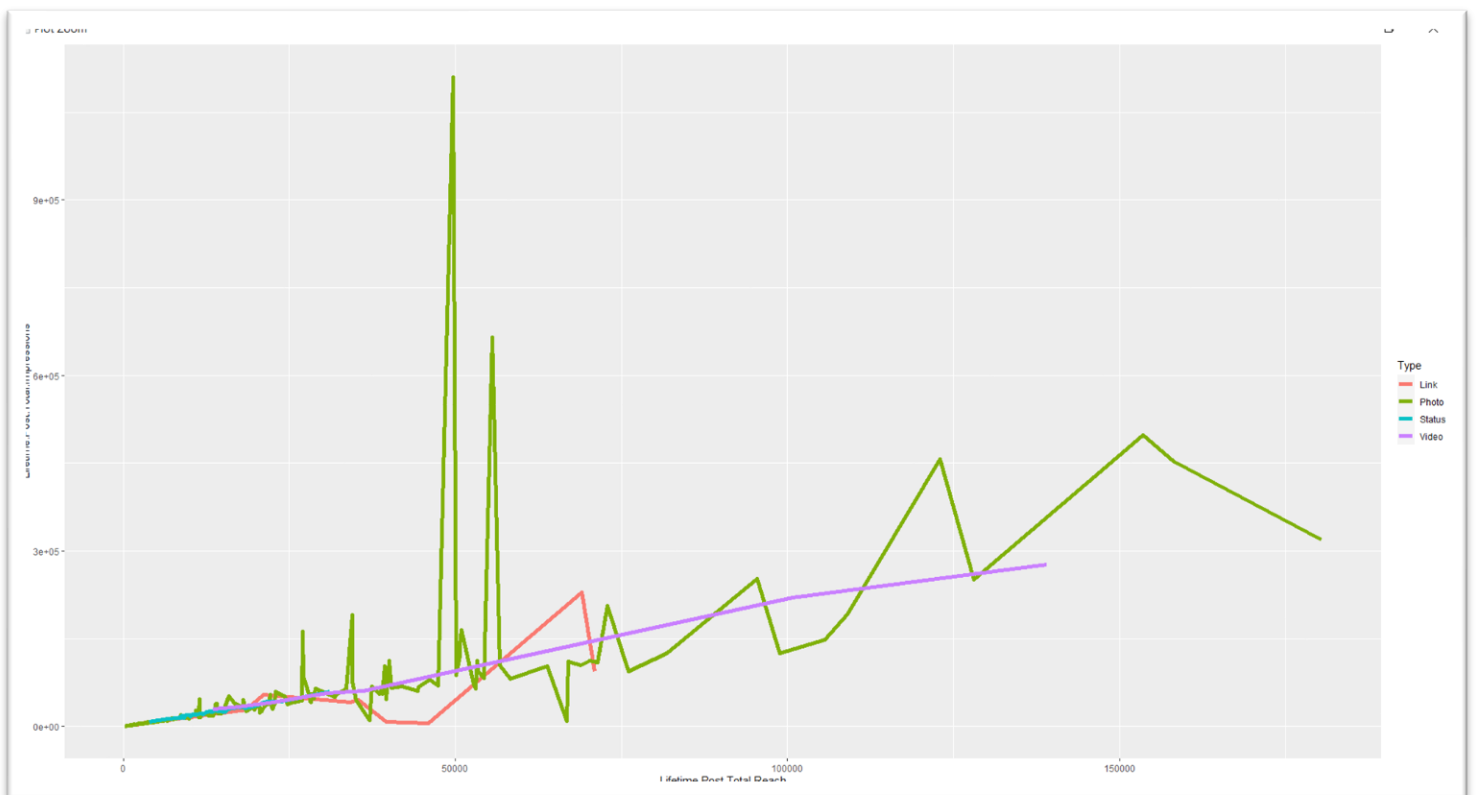
```
> # Lineplot
```

```
> facebook_lineplot <- ggplot(Facebook_data, aes(x = Lifetime.Post.Total.Reach,
```

+ y = Lifetime.Post.Total.Impressions,

```
+ color = Type)) + geom_line()
```

```
> facebook_lineplot
```



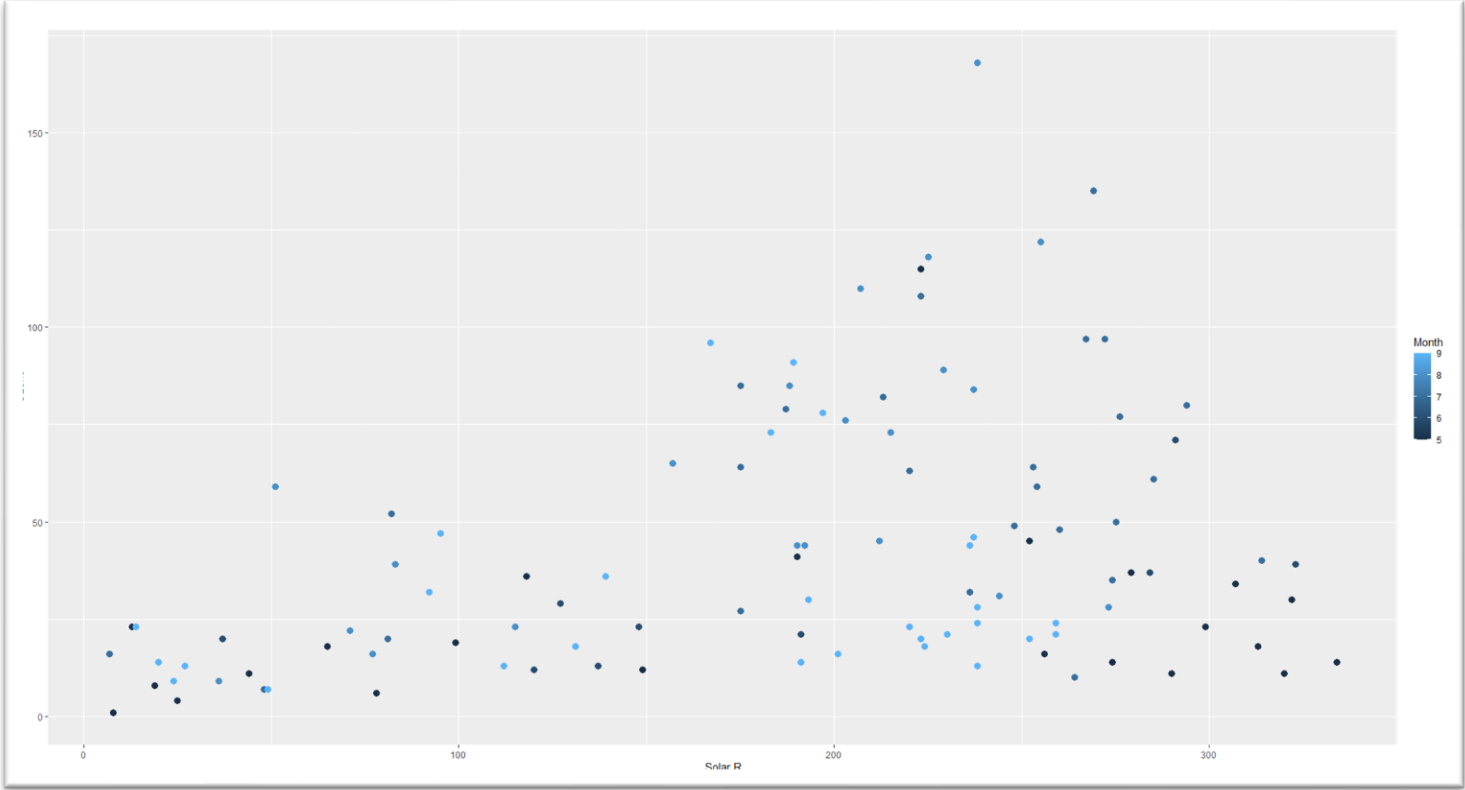
```
> data("airquality")
```

```
> my_airquality_data <- airquality
```

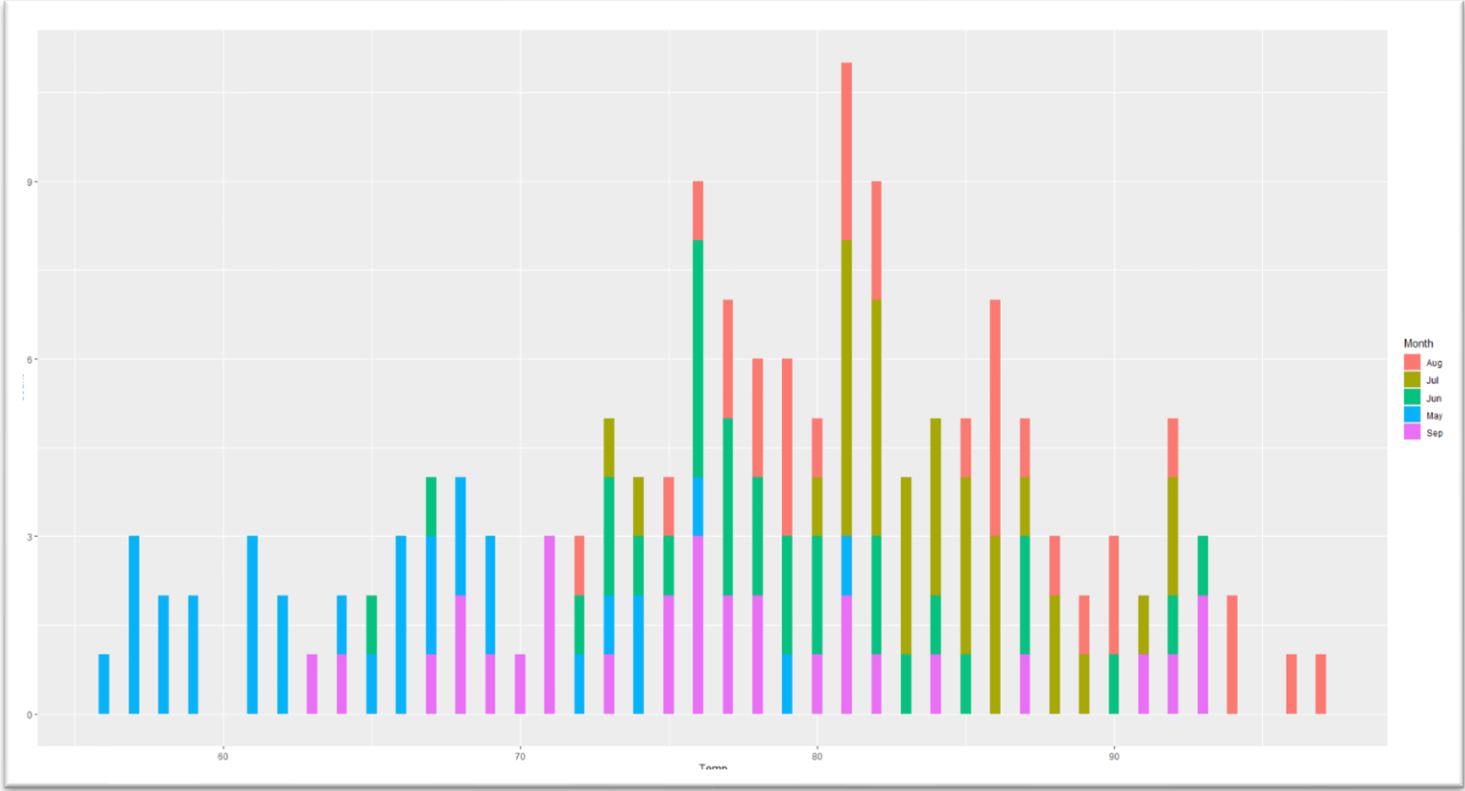
```
> my_airquality_data$Month <- month.abb[my_airquality_data$Month]
```

```
> # Scatterplot
> airquality_scatterplot <- ggplot(my_airquality_data, aes(x = Solar.R, y = Ozone,
+
+                               color = Month)) + geom_point(size = 3)
> airquality_scatterplot
```

Warning message:  
Removed 42 rows containing missing values (geom\_point).



```
> # Barplot
> airquality_barplot <- ggplot(my_airquality_data, aes(x = Temp, fill = Month)) + geom_bar(width = 0.35)
> airquality_barplot
```

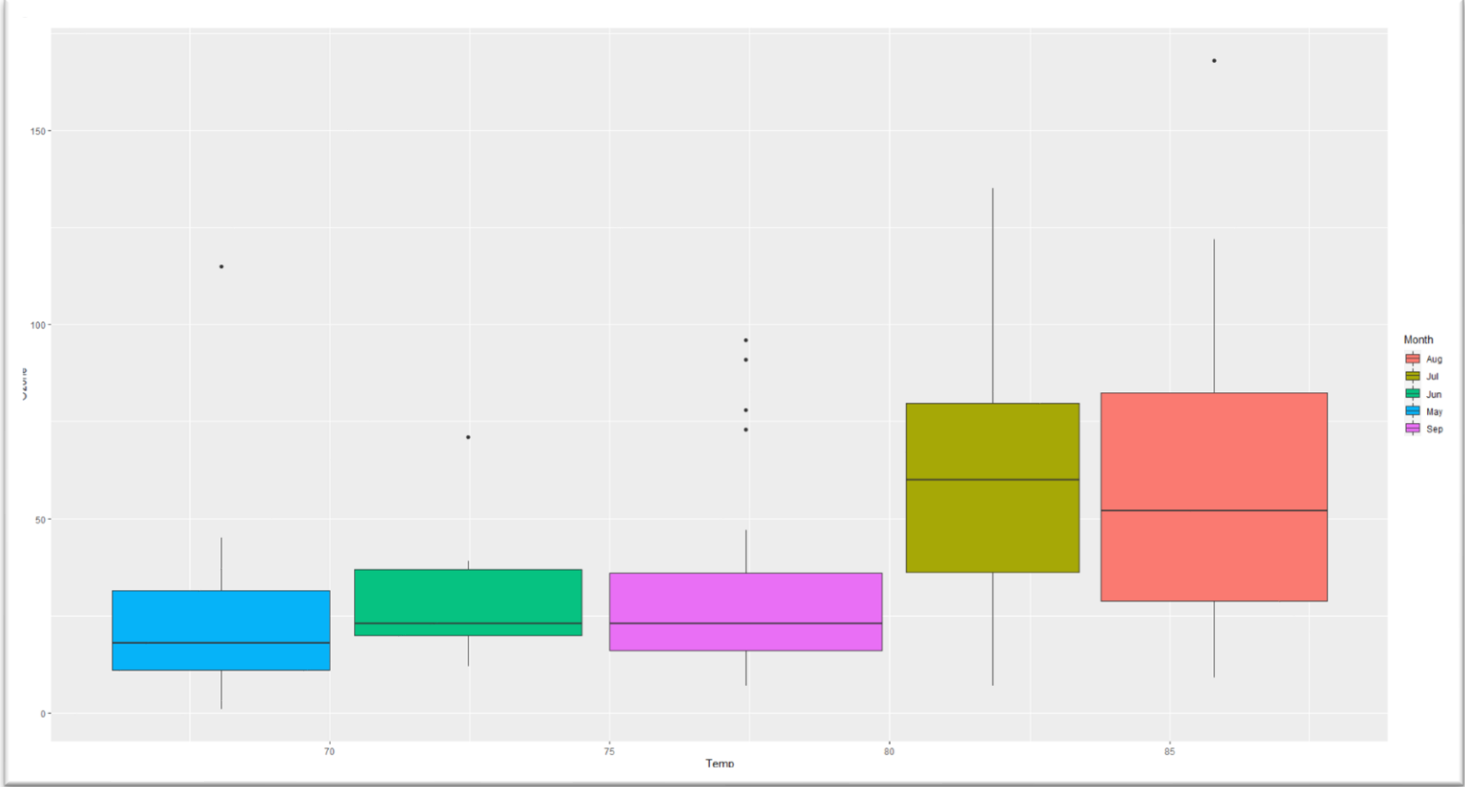


```
> # Boxplot
> airquality_boxplot <- ggplot(my_airquality_data, aes(x = Temp , y = Ozone, fill = Month)) +
```



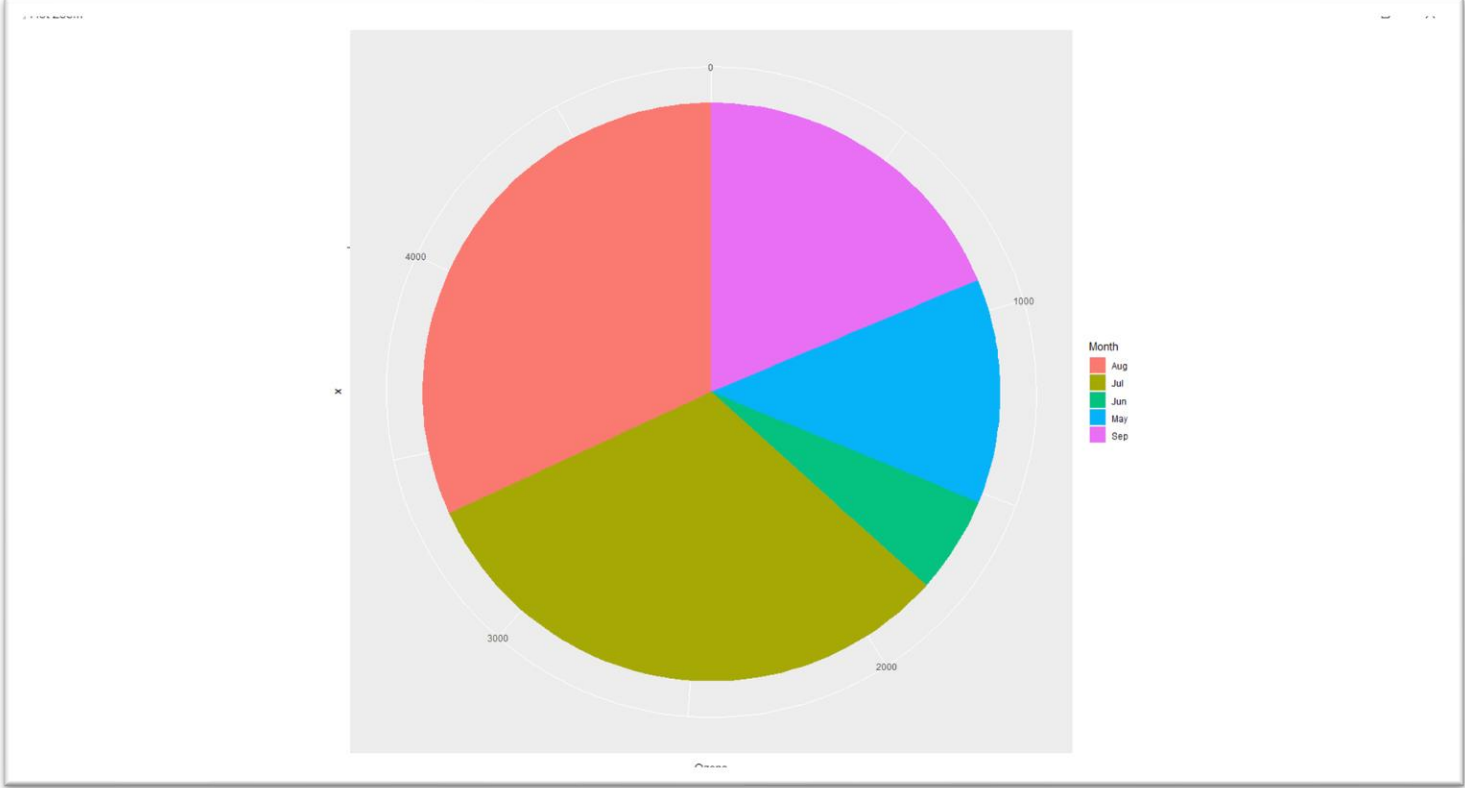
```
+ geom_boxplot()
> airquality_boxplot
```

Warning message:  
Removed 37 rows containing non-finite values (stat\_boxplot).



```
> # Piechart
> ggplot(my_airquality_data, aes(x = "", y = Ozone, fill = Month)) +
+ geom_bar(stat = "identity", width = 0.2 ) + coord_polar("y")
```

Warning message:  
Removed 37 rows containing missing values (position\_stack).



```
> # Lineplot
> airquality_lineplot <- ggplot(my_airquality_data, aes(y = Wind,
```

```
+ x = Ozone, color = Month)) + geom_line(size = 2)
> airquality_lineplot
```

Warning message:  
Removed 37 row(s) containing missing values (geom\_path).

