graphs

18f0371

11/11/2021

```
data(airquality)
str(airquality)
```

```
## '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 Markdown

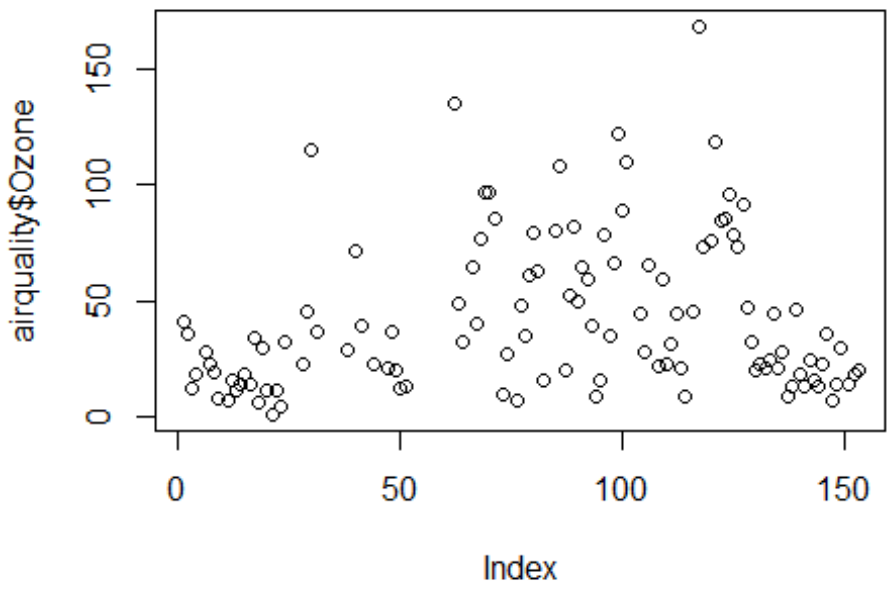
```
str(airquality)
```

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

Including Plots

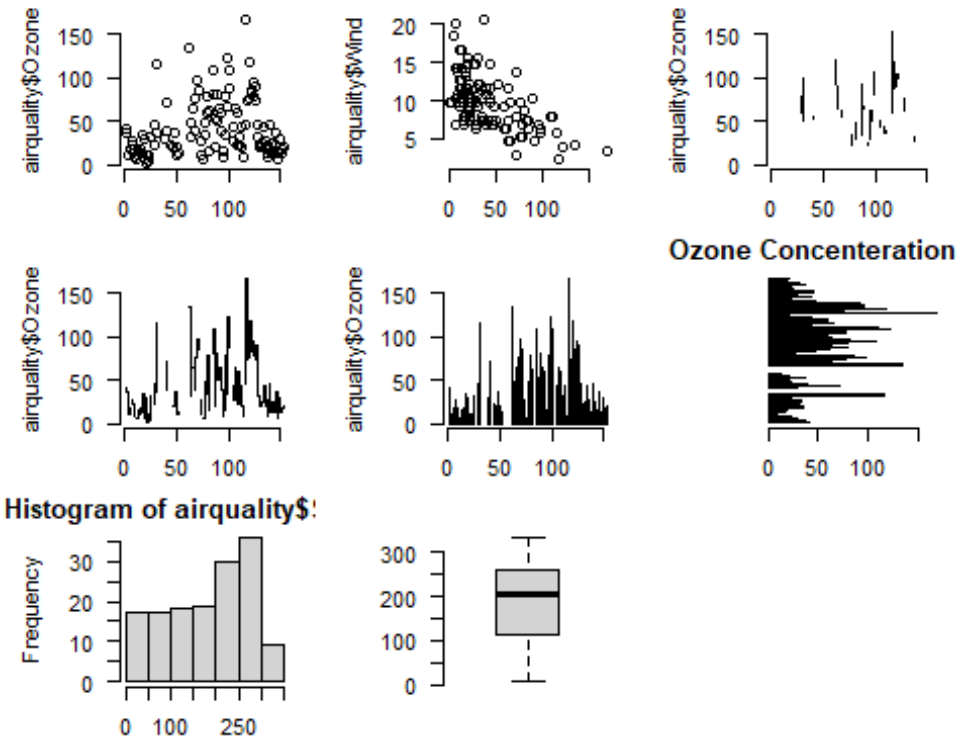
You can also embed plots, for example:

```
plot(airquality$Ozone)
```

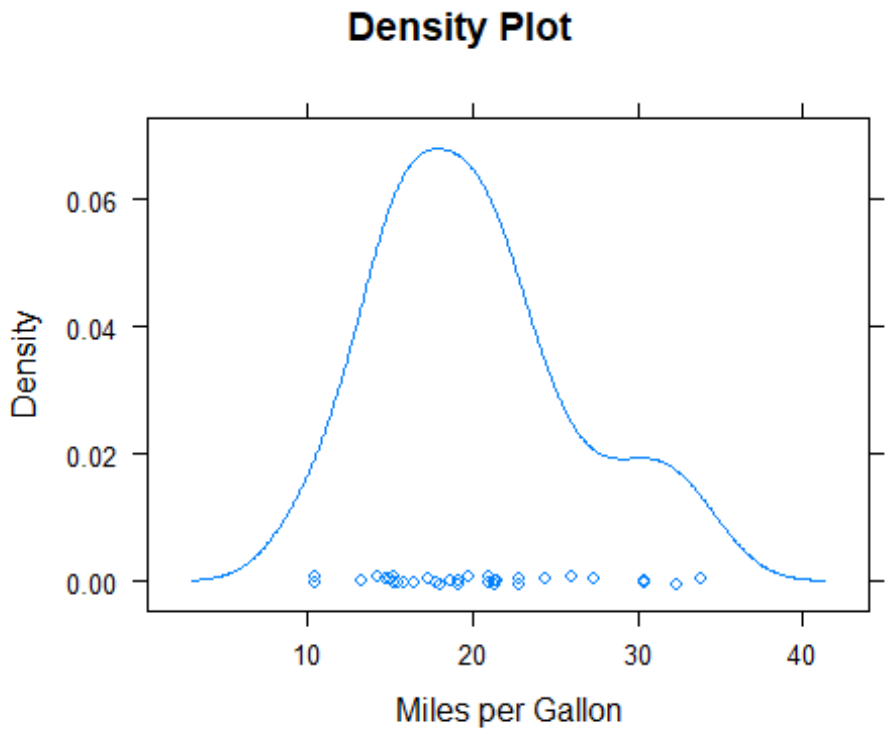



```
summary(airquality)

##      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
##

par(mfrow=c(3,3), mar=c(2,5,2,1), las=1, bty="n")
plot(airquality$Ozone)
plot(airquality$Ozone, airquality$Wind)
plot(airquality$Ozone, type= "c")
plot(airquality$Ozone, type= "s")
plot(airquality$Ozone, type= "h")
barplot(airquality$Ozone, main = 'Ozone Concenteration in air',xlab = 'ozone
levels', col='green',horiz = TRUE)
```

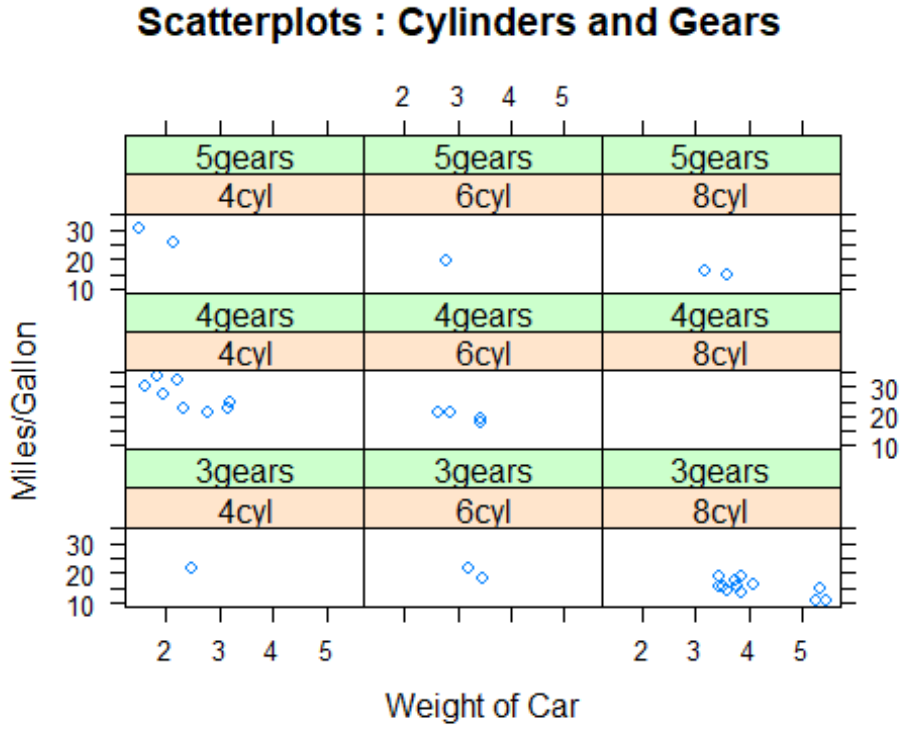
```
hist(airquality$Solar.R)
boxplot(airquality$Solar.R)
```




```
library(lattice)
#Loading the dataset
attach(mtcars)
gear_factor<-factor(gear,levels=c(3,4,5),
labels=c("3gears","4gears","5gears"))

cyl_factor <-factor(cyl,levels=c(4,6,8),
labels=c("4cyl","6cyl","8cyl"))
densityplot(~mpg, main="Density Plot",  xlab="Miles per Gallon")
```




```
xyplot(mpg~wt|cyl_factor*gear_factor,
main="Scatterplots : Cylinders and Gears",
ylab="Miles/Gallon", xlab="Weight of Car")
```



```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'mtcars':
##
##     mpg
```

```
#Loading the dataset
attach(mtcars)
```

```
## The following object is masked from package:ggplot2:
##
##     mpg
```

```
## The following objects are masked from mtcars (pos = 4):
##
##     am, carb, cyl, disp, drat, gear, hp, mpg, qsec, vs, wt
```

```
# create factors with value labels

# mtcars$gear <- factor(mtcars$gear,levels=c(3,4,5),
# labels=c("3gears", "4gears", "5gears"))
# mtcars$am <- factor(mtcars$am,levels=c(0,1),
# labels=c("Automatic","Manual"))
# mtcars$cyl <- factor(mtcars$cyl,levels=c(4,6,8),
# labels=c("4cyl","6cyl","8cyl"))
```

```
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

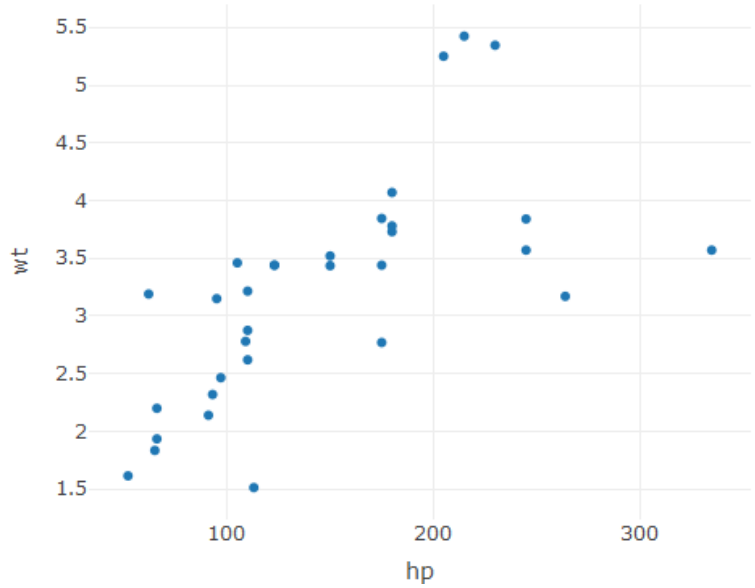
```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

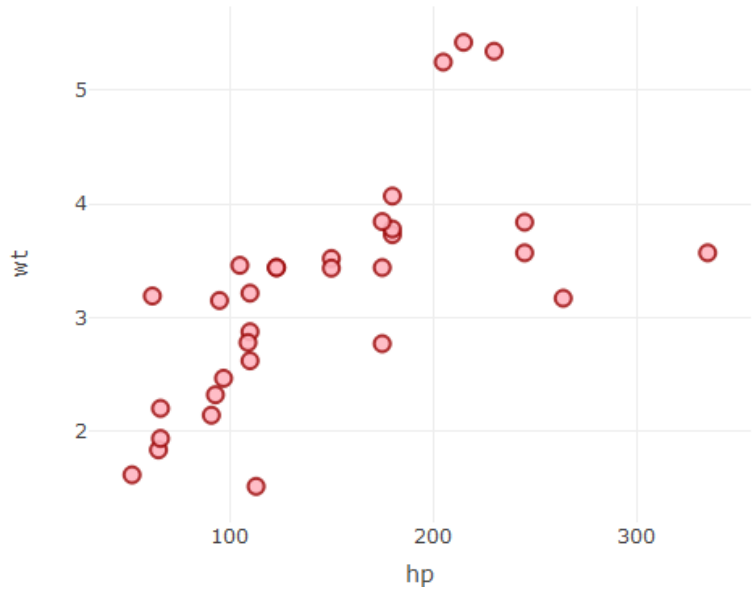
```
p <- plot_ly(data = mtcars, x = ~hp, y = ~wt)
p
```

```
## No trace type specified:
##   Based on info supplied, a 'scatter' trace seems appropriate.
##   Read more about this trace type -> https://plotly.com/r/reference/#scatt
er
```

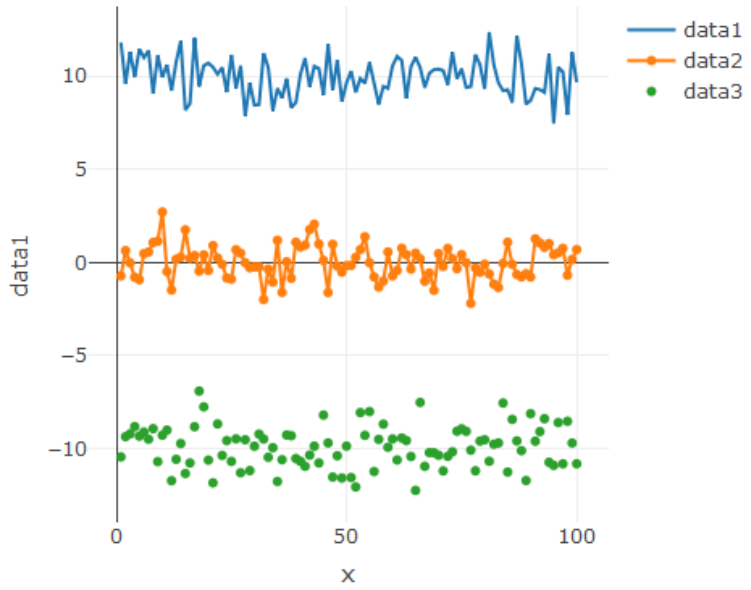
```
## No scatter mode specifed:
##   Setting the mode to markers
##   Read more about this attribute -> https://plotly.com/r/reference/#scatte
r-mode
```



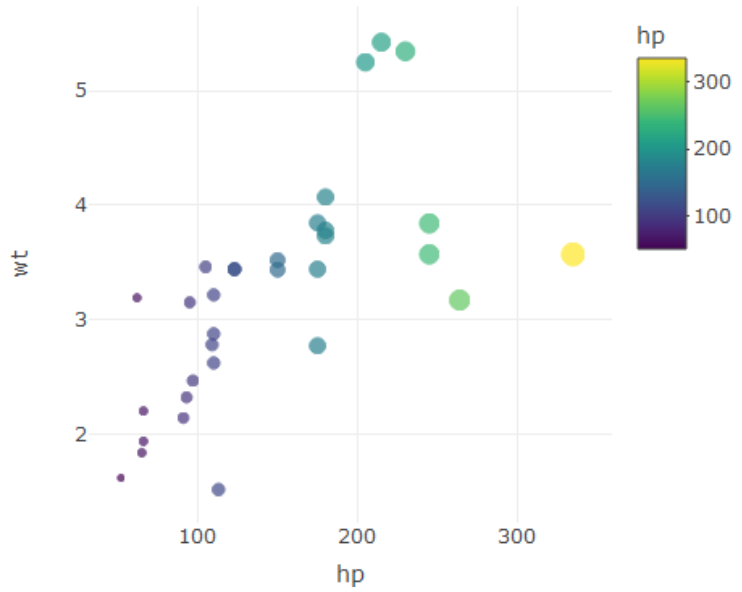

```
p <- plot_ly(data = mtcars, x = ~hp, y = ~wt, marker = list(size = 10, color
= 'rgba(255, 182, 193, .9)', line = list(color = 'rgba(152, 0, 0, .8)', width
= 2)))
p

## No trace type specified:
##   Based on info supplied, a 'scatter' trace seems appropriate.
##   Read more about this trace type -> https://plotly.com/r/reference/#scatt
er

## No scatter mode specifed:
##   Setting the mode to markers
##   Read more about this attribute -> https://plotly.com/r/reference/#scatte
r-mode
```




```
data1 <- rnorm(100, mean = 10)
data2 <- rnorm(100, mean = 0)
data3 <- rnorm(100, mean = -10)
x <- c(1:100)
data <- data.frame(x, data1, data2, data3)
p <- plot_ly(data, x = ~x)%>%

add_trace(y = ~data1, name = 'data1',mode = 'lines')%>%
add_trace(y = ~data2, name = 'data2', mode = 'lines+markers')%>%
add_trace(y = ~data3, name = 'data3', mode = 'markers')
p

## No trace type specified:
##   Based on info supplied, a 'scatter' trace seems appropriate.
##   Read more about this trace type -> https://plotly.com/r/reference/#scatt
er
## No trace type specified:
##   Based on info supplied, a 'scatter' trace seems appropriate.
##   Read more about this trace type -> https://plotly.com/r/reference/#scatt
er
## No trace type specified:
##   Based on info supplied, a 'scatter' trace seems appropriate.
##   Read more about this trace type -> https://plotly.com/r/reference/#scatt
er
```




```
p <- plot_ly(data = mtcars, x =~hp, y = ~wt,color = ~hp, size = ~hp )
p

## No trace type specified:
##   Based on info supplied, a 'scatter' trace seems appropriate.
##   Read more about this trace type -> https://plotly.com/r/reference/#scatt
er

## No scatter mode specifed:
##   Setting the mode to markers
##   Read more about this attribute -> https://plotly.com/r/reference/#scatte
r-mode

## Warning: `line.width` does not currently support multiple values.
```




```
library(maps)
map(database='state')
```

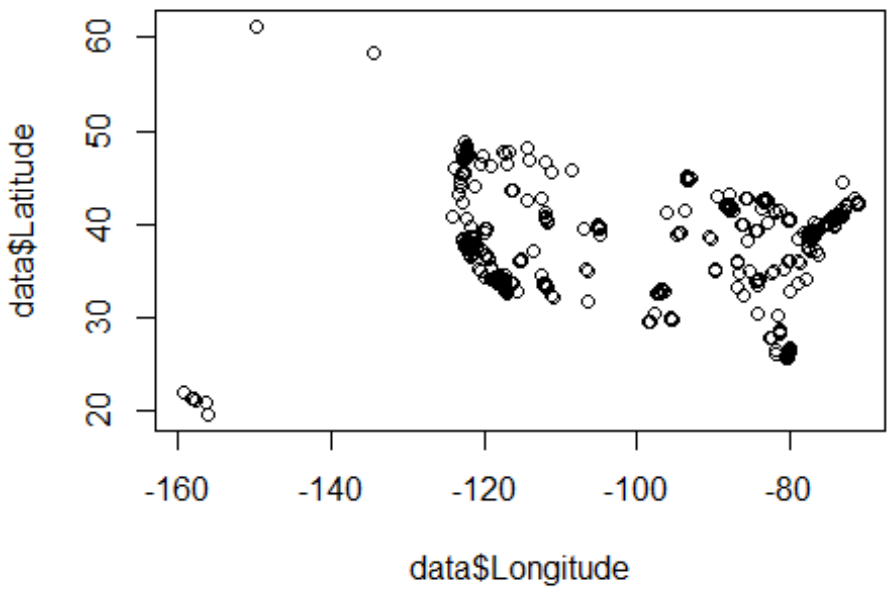
```
data <- read.csv('ABC_locations.csv', sep=",")
head(data)

##                    Address       City   State   Zip.Code Latitude  Longit
ude
```

```
## 1  1205 N. Memorial Parkway Huntsville Alabama 35801-5930 34.74309  -86.60
096
## 2      3650 Galleria Circle     Hoover Alabama 35244-2346 33.37765  -86.81
242
## 3    8251 Eastchase Parkway Montgomery Alabama      36117 32.36389  -86.15
088
## 4 5225 Commercial Boulevard     Juneau  Alaska 99801-7210 58.35920 -134.48
300
## 5      330 West Dimond Blvd  Anchorage  Alaska 99515-1950 61.14327 -149.88
422
## 6          4125 DeBarr Road  Anchorage  Alaska 99508-3115 61.21081 -149.80
434
```

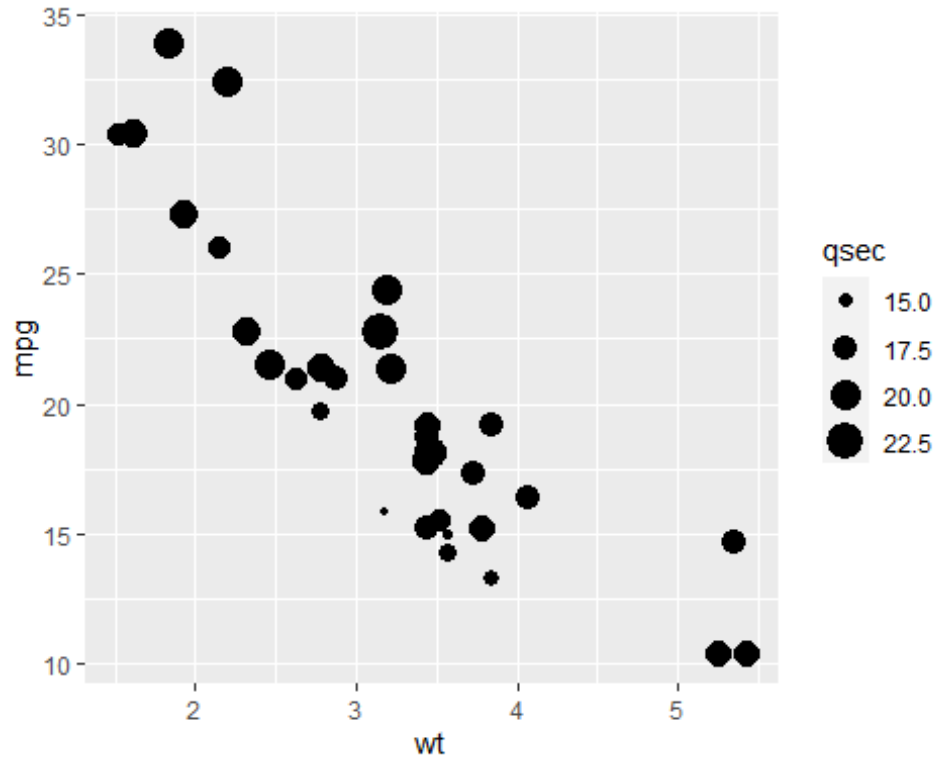
```
data <- read.csv('ABC_locations.csv', sep=",")
head(data)
```

```
##                     Address       City   State   Zip.Code Latitude  Longit
ude
## 1  1205 N. Memorial Parkway Huntsville Alabama 35801-5930 34.74309  -86.60
096
## 2      3650 Galleria Circle     Hoover Alabama 35244-2346 33.37765  -86.81
242
## 3    8251 Eastchase Parkway Montgomery Alabama      36117 32.36389  -86.15
088
## 4 5225 Commercial Boulevard     Juneau  Alaska 99801-7210 58.35920 -134.48
300
## 5      330 West Dimond Blvd  Anchorage  Alaska 99515-1950 61.14327 -149.88
422
## 6          4125 DeBarr Road  Anchorage  Alaska 99508-3115 61.21081 -149.80
434
```

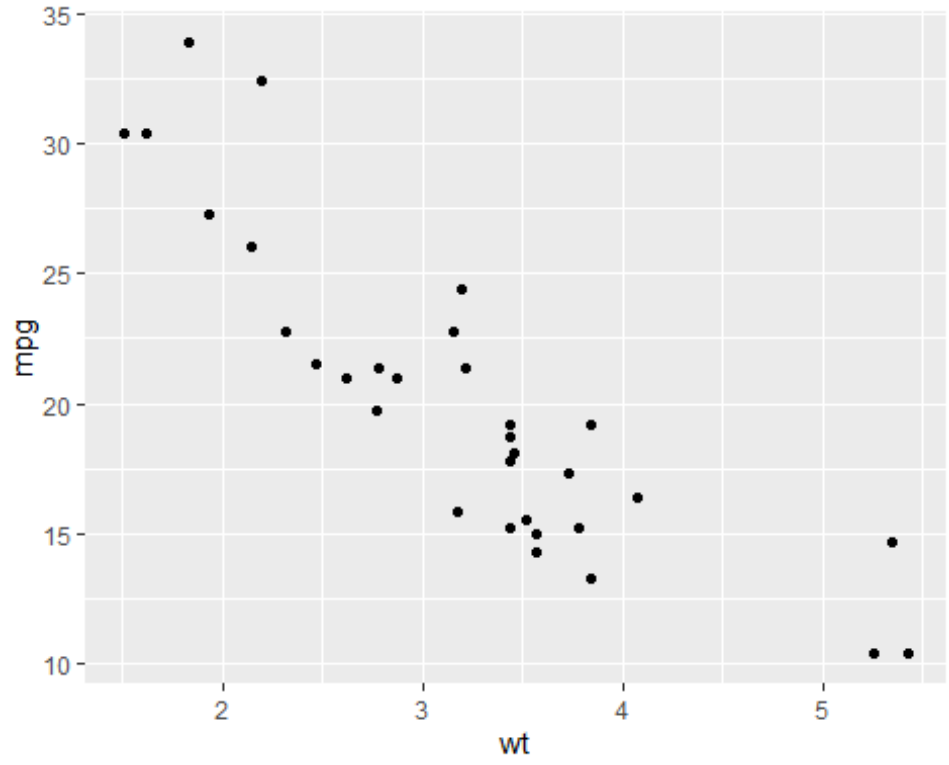
```
plot(data$Longitude,data$Latitude)
```



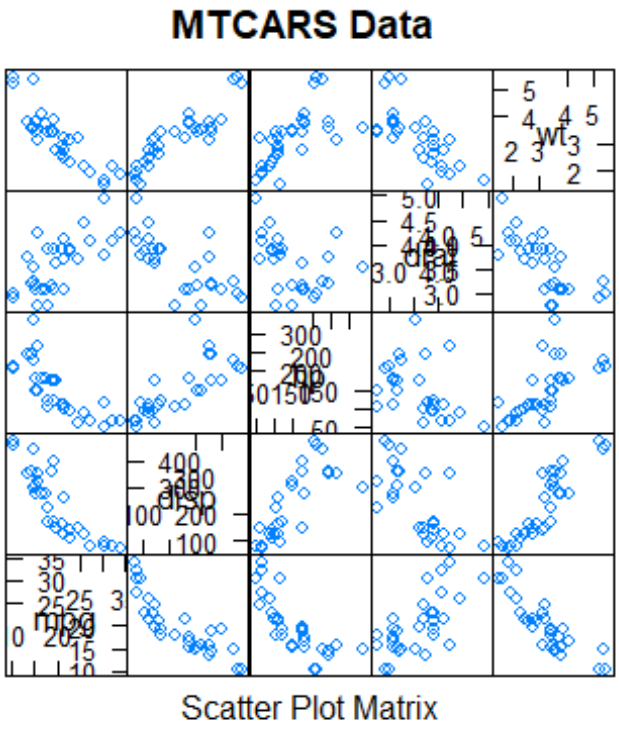

```
# library(plotly)
# library(maps)

ggplot(data = mtcars, mapping = aes(x = wt, y = mpg, size = qsec)) + geom_poi
nt()
```



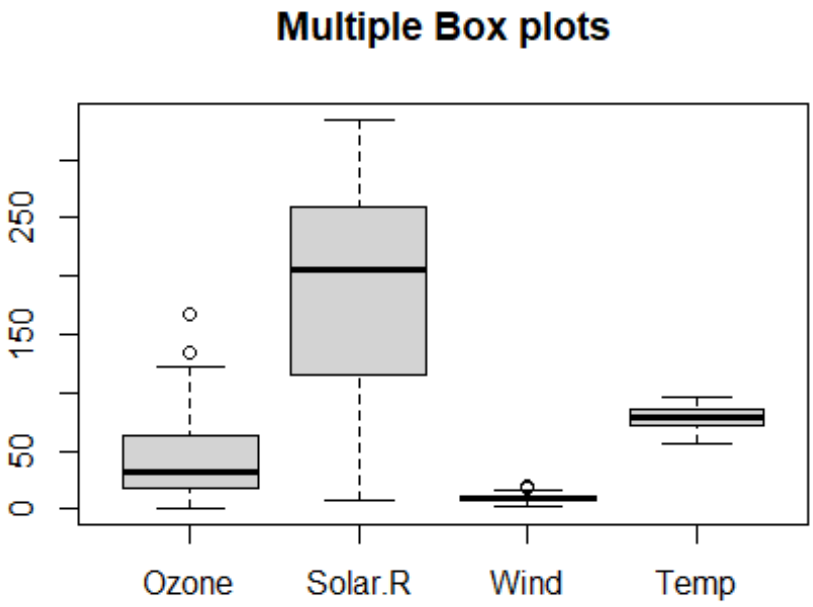

```
ggplot(data = mtcars, mapping = aes(x = wt, y = mpg)) + geom_point()
```



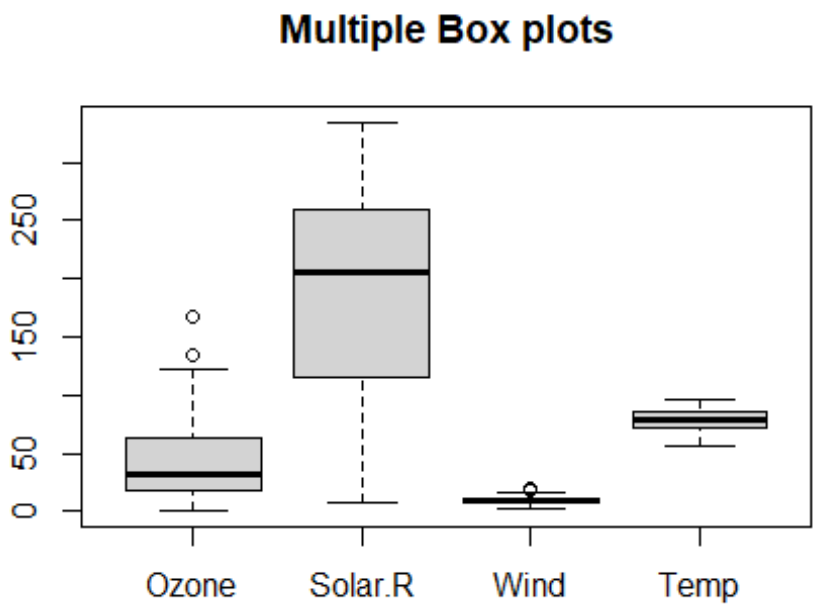

```
splom(mtcars[c(1,3,4,5,6)], main="MTCARS Data")
```



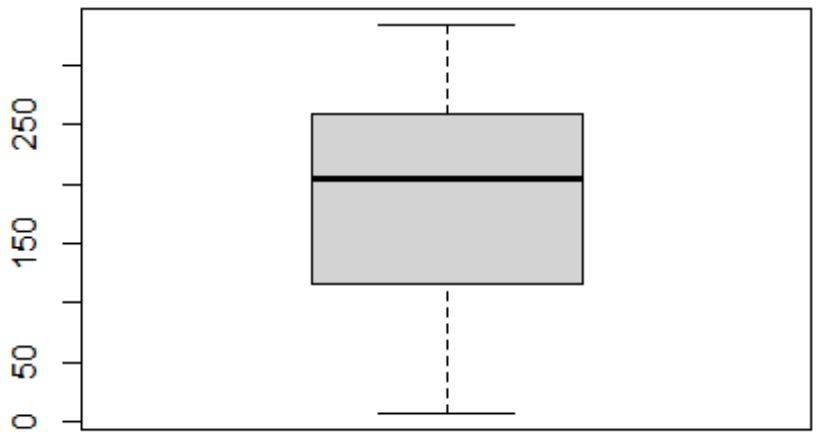

```
boxplot(airquality[,0:4], main='Multiple Box plots')
```



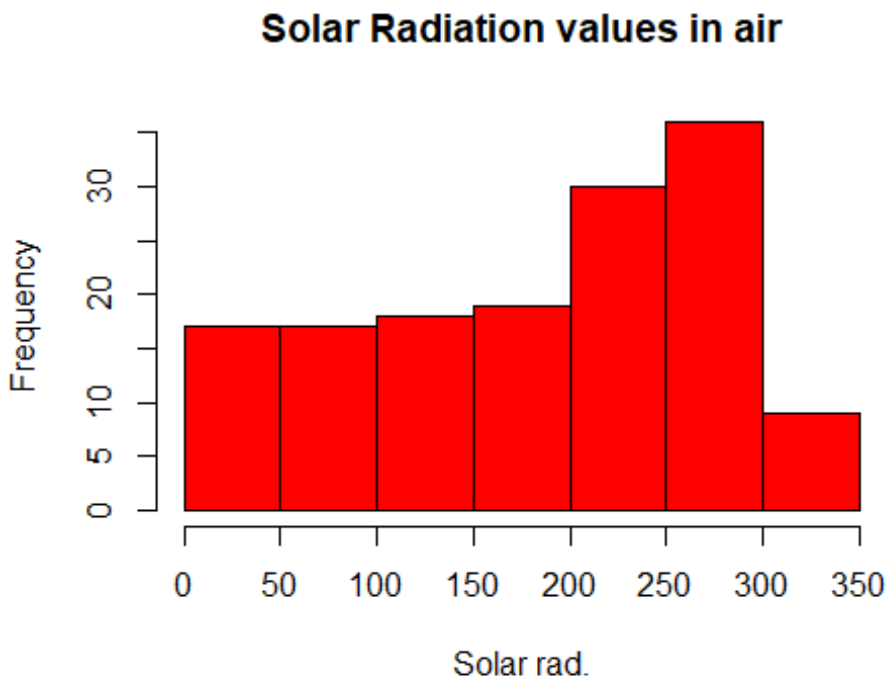

```
boxplot(airquality[,0:4], main='Multiple Box plots')
```



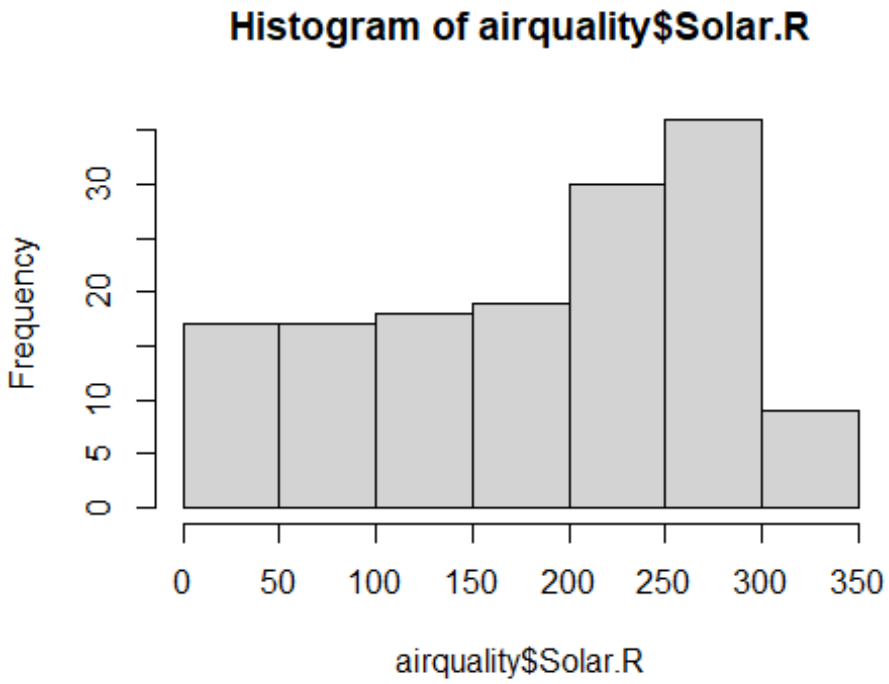

```
boxplot(airquality$Solar.R)
```



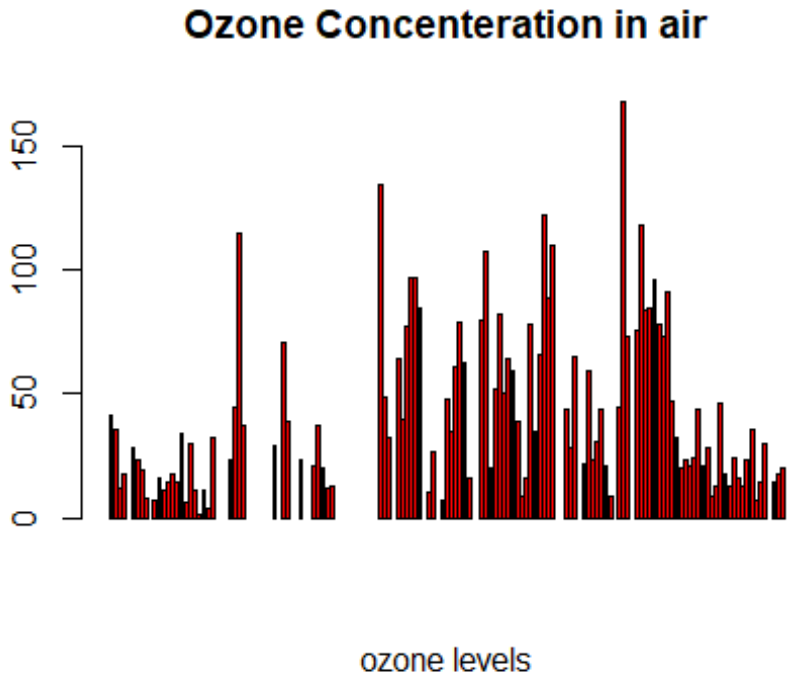
```
hist(airquality$Solar.R, main = 'Solar Radiation values in air',xlab = 'Solar rad.', col='red')
```



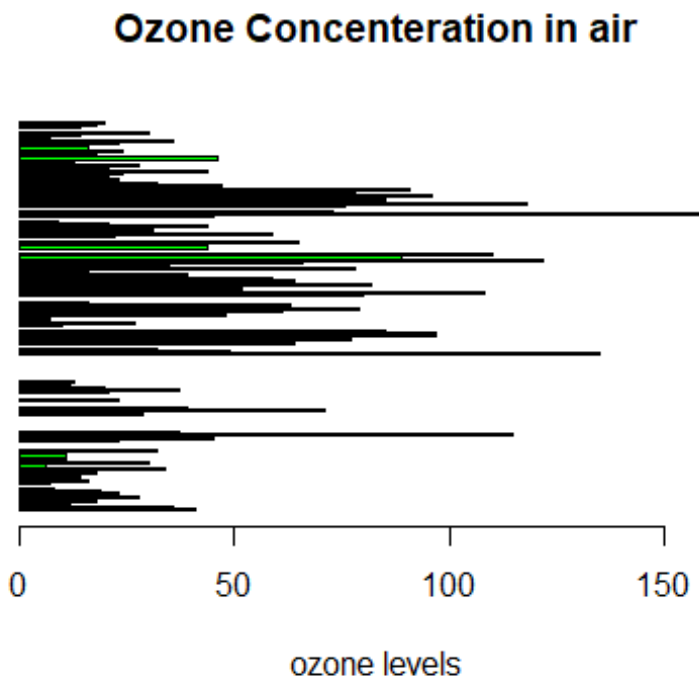

```
hist(airquality$Solar.R)
```



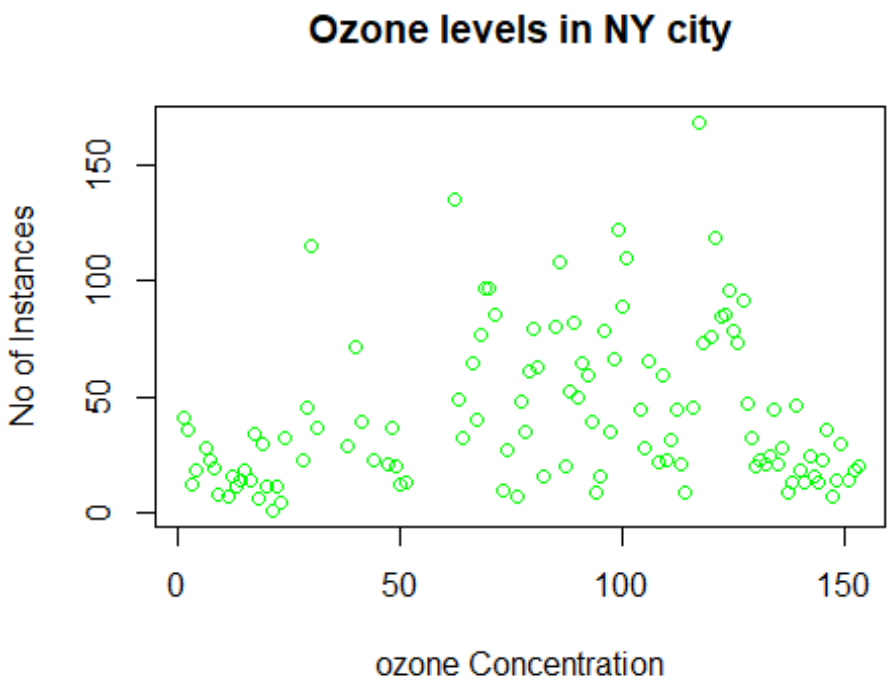

```
barplot(airquality$Ozone, main = 'Ozone Concenteration in air',xlab = 'ozone
levels', col='red',horiz = FALSE)
```



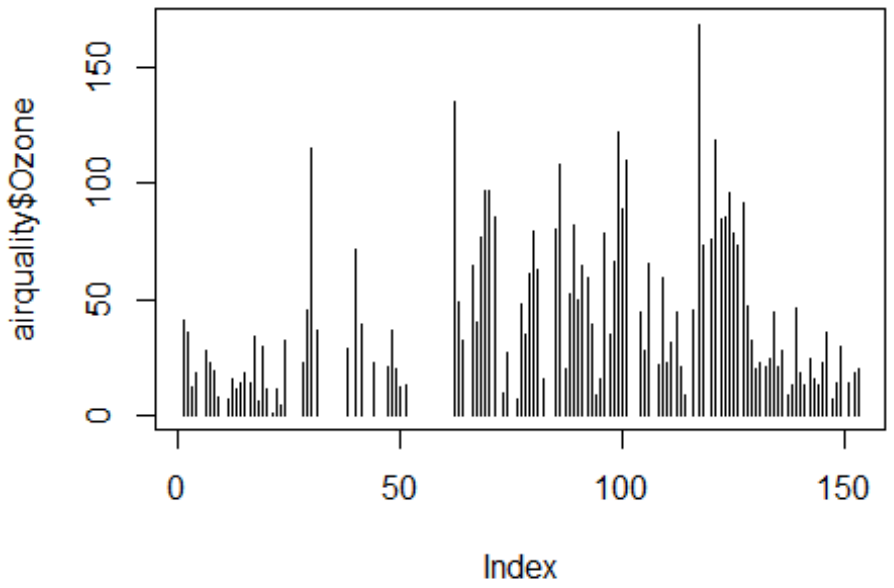
```
barplot(airquality$Ozone, main = 'Ozone Concenteration in air',xlab = 'ozone
levels', col= 'green',horiz = TRUE)
```



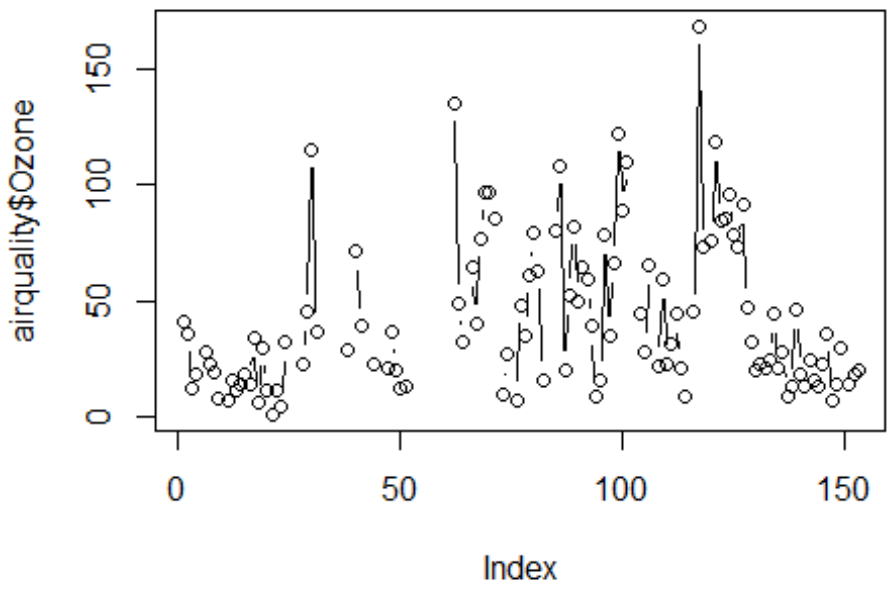

```
plot(airquality$Ozone, xlab = 'ozone Concentration', ylab = 'No of Instances'
, main = 'Ozone levels in NY city', col = 'green')
```



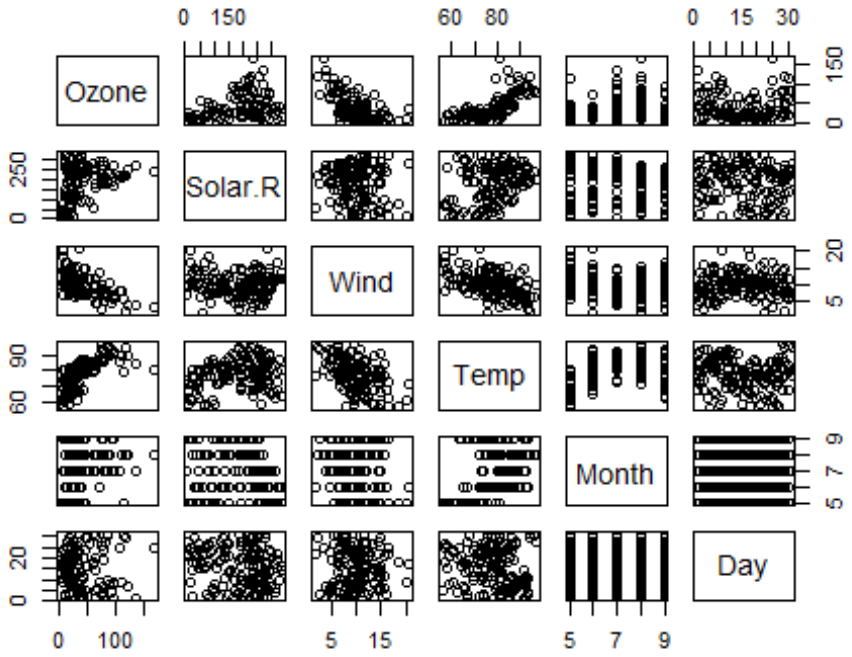

```
plot(airquality$Ozone, type= "h")
```



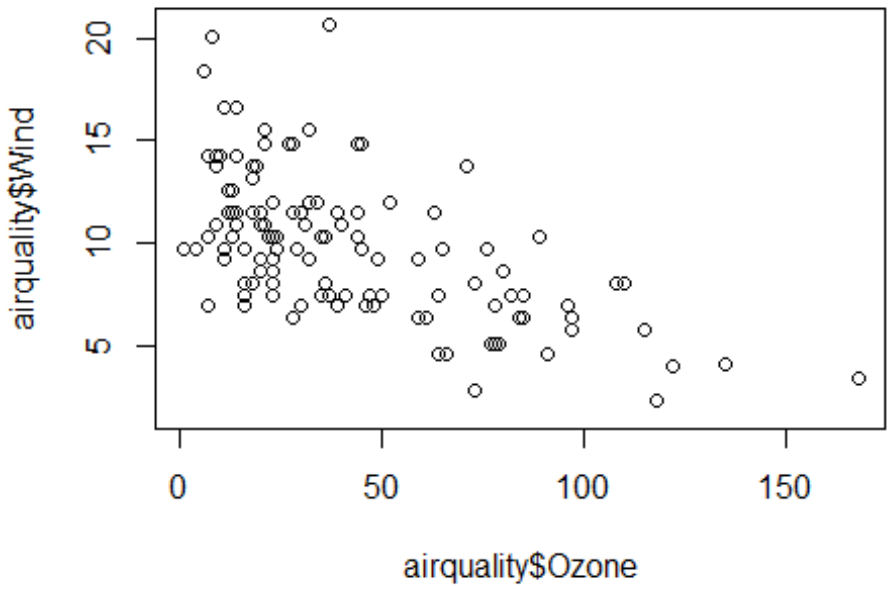

```
plot(airquality$Ozone, type= "b")
```




```
plot(airquality)
```




```
plot(airquality$Ozone, airquality$Wind)
```



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.