Ex.No.4 **INTRODUCTION TO GGPLOT PACKAGE**

Date: 8-08-23

Aim

To implement the GGPLOT package in R programming in the experiments and learn about plots and graphs.

Procedure

1. To do programming in R, first install “RStudio” and “R” in the system. RStudio is an integrated development environment [IDE] for R and python.
2. Select the File in taskbar →open New file →R script or use shortcut “ctrl+shift+N”
3. Install the ‘ggplot’ package and load it in R.
4. Import the built-in dataset using data(iris).
5. Use the various plotting techniques on the iris data and analyse it.
6. Write the program in the script and save it using the extension R.
7. Run the program by clicking Run option or use the shortcut “ctrl+enter”.
8. See the output in the console tab.

Concepts Involved

- Plotting of data using ‘ggplot’ package.

IMPORTING IRIS DATA SET

Import iris data set which is built-in in ggplot oackage

Script

```
data(iris)
```

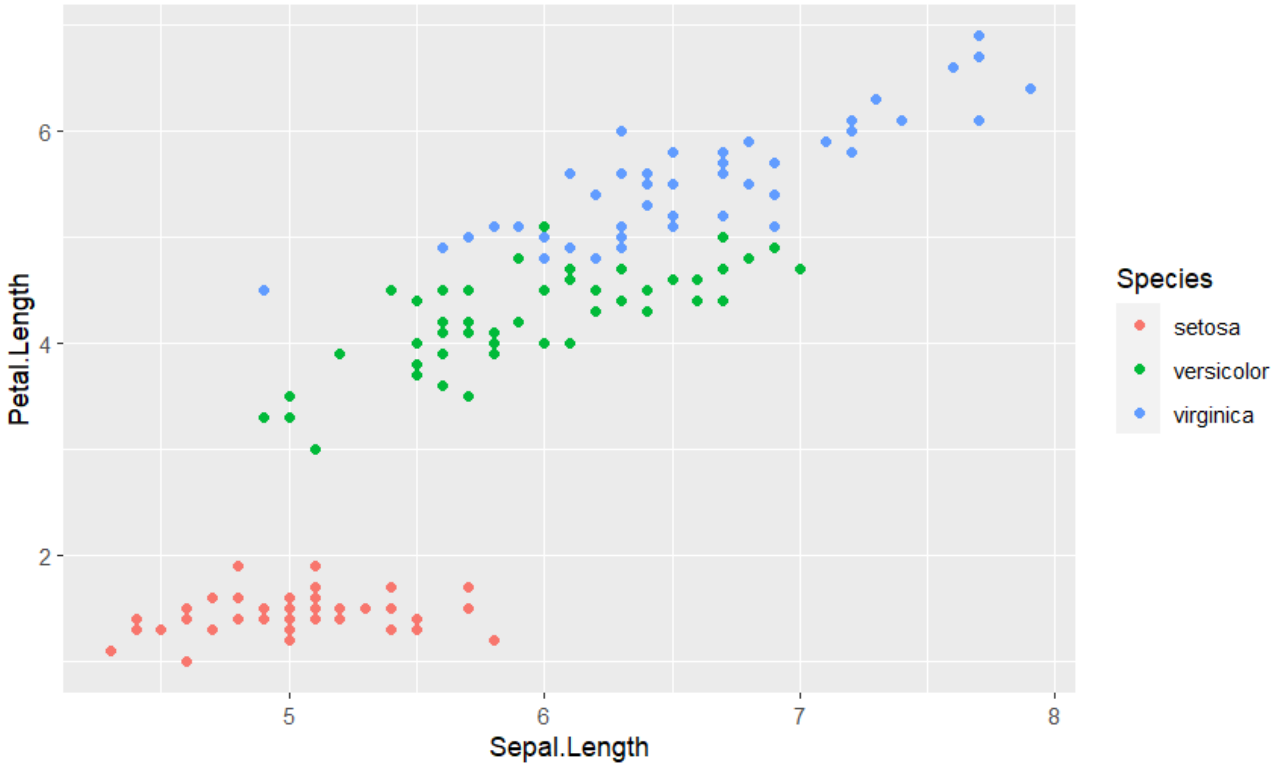
SCATTER PLOT

To plot it, we will be using the geom_point() function. Here we will plot the Sepal length variable on the x-axis and the petal length variable on the y axis.

Script

```
data(iris)

ggplot(iris, aes(x=Sepal.Length, y=Petal.Length))+geom_point()

ggplot(iris, aes(x=Sepal.Length, y=Petal.Length, col=Species))+geom_point()
```

Output



CREATING PATTERNS

Use the geom_smooth function for showing simple trends or approximations.

Script

#Creating pattrens using tidyverse

data(mtcars)

glimpse (mtcars)

Output

```
Rows: 32
Columns: 11
$ mpg  <dbl> 21.0, 21.0, 22.8, 21.4, 18.7, 18.1, 14.3, 24.4, 22.
8, 19.2, 17.8, 16.4, 17.3, 15…
$ cyl  <dbl> 6, 6, 4, 6, 8, 6, 8, 4, 4, 6, 6, 8, 8, 8, 8, 8, 8,
4, 4, 4, 4, 8, 8, 8, 8, 4, 4,…
$ disp <dbl> 160.0, 160.0, 108.0, 258.0, 360.0, 225.0, 360.0, 14
6.7, 140.8, 167.6, 167.6, 275…
$ hp   <dbl> 110, 110, 93, 110, 175, 105, 245, 62, 95, 123, 123,
180, 180, 180, 205, 215, 230…
```

```
$ drat <dbl> 3.90, 3.90, 3.85, 3.08, 3.15, 2.76, 3.21, 3.69, 3.9
2, 3.92, 3.92, 3.07, 3.07, 3.…
$ wt   <dbl> 2.620, 2.875, 2.320, 3.215, 3.440, 3.460, 3.570, 3.
190, 3.150, 3.440, 3.440, 4.0…
$ qsec <dbl> 16.46, 17.02, 18.61, 19.44, 17.02, 20.22, 15.84, 20
.00, 22.90, 18.30, 18.90, 17.…
$ vs   <dbl> 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0,
1, 1, 1, 1, 0, 0, 0, 0, 1, 0,…
$ am   <dbl> 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 1, 1, 0, 0, 0, 0, 0, 1, 1,…
$ gear <dbl> 4, 4, 4, 3, 3, 3, 3, 4, 4, 4, 4, 3, 3, 3, 3, 3, 3,
4, 4, 4, 3, 3, 3, 3, 3, 4, 5,…
$ carb <dbl> 4, 4, 1, 1, 2, 1, 4, 2, 2, 4, 4, 3, 3, 3, 4, 4, 4,
1, 2, 1, 1, 2, 2, 4, 2, 1, 2,…
```

BAR PLOT

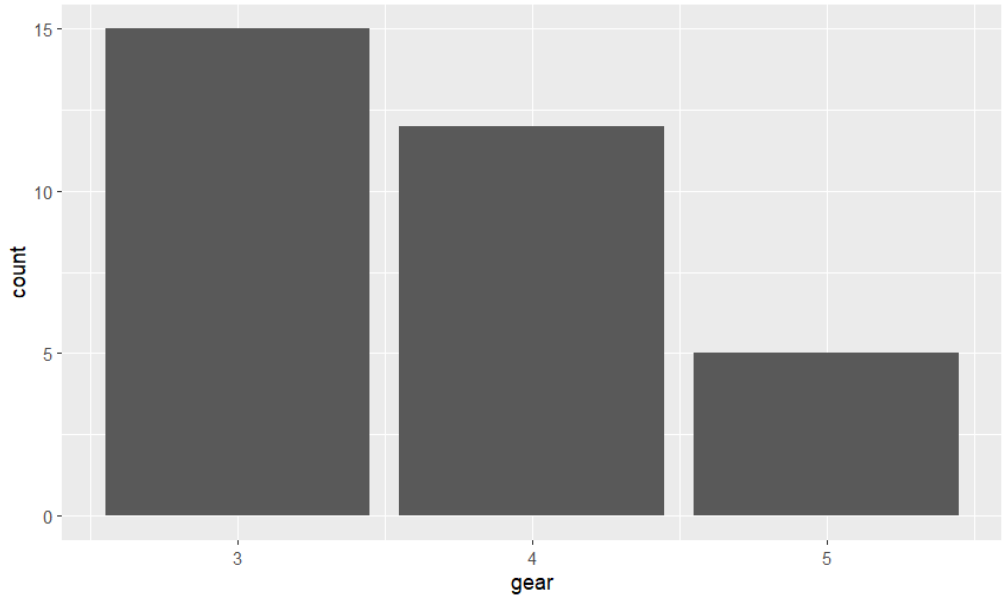
This plot is used to measure changes over a particular span of time. It is the best option to represent the data when changes are large.

Script

#bar plot

ggplot(mtcars, aes(x = gear)) +geom_bar()

Output

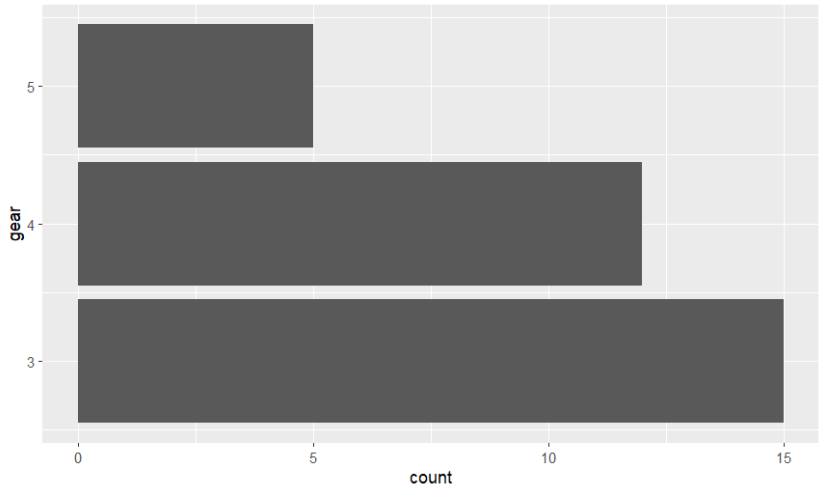


Script

#interchanging axes

ggplot(mtcars, aes(x = gear)) +geom_bar()+coord_flip()

Output



HISTOGRAM

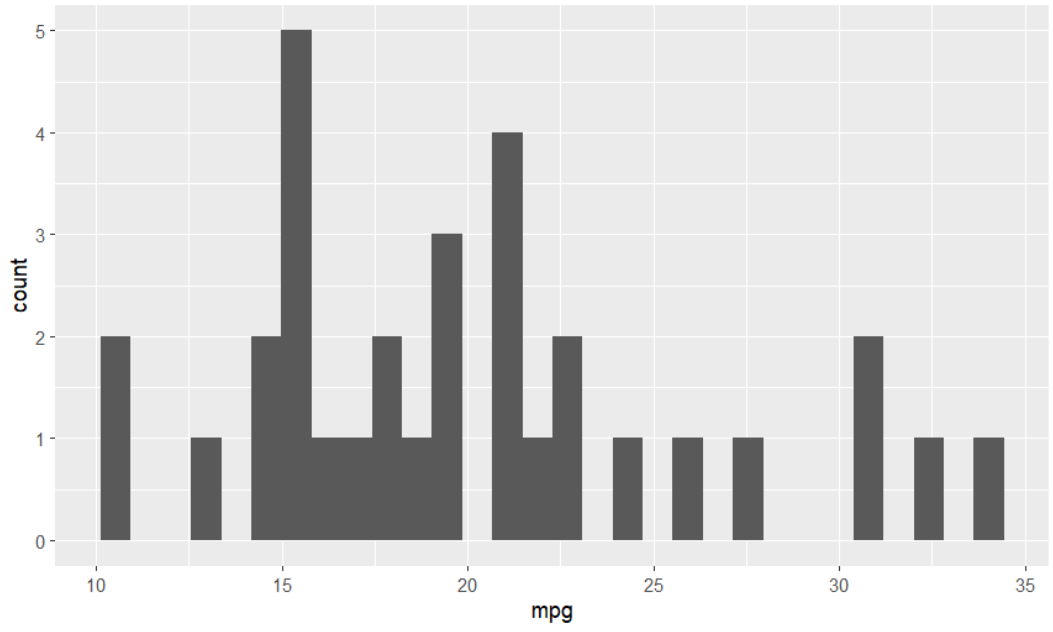
A Histogram is used to show the frequency distribution of a continuous-discrete variable.

Script

#histogram

ggplot(mtcars,aes(x=mpg)) + geom_histogram()

Output



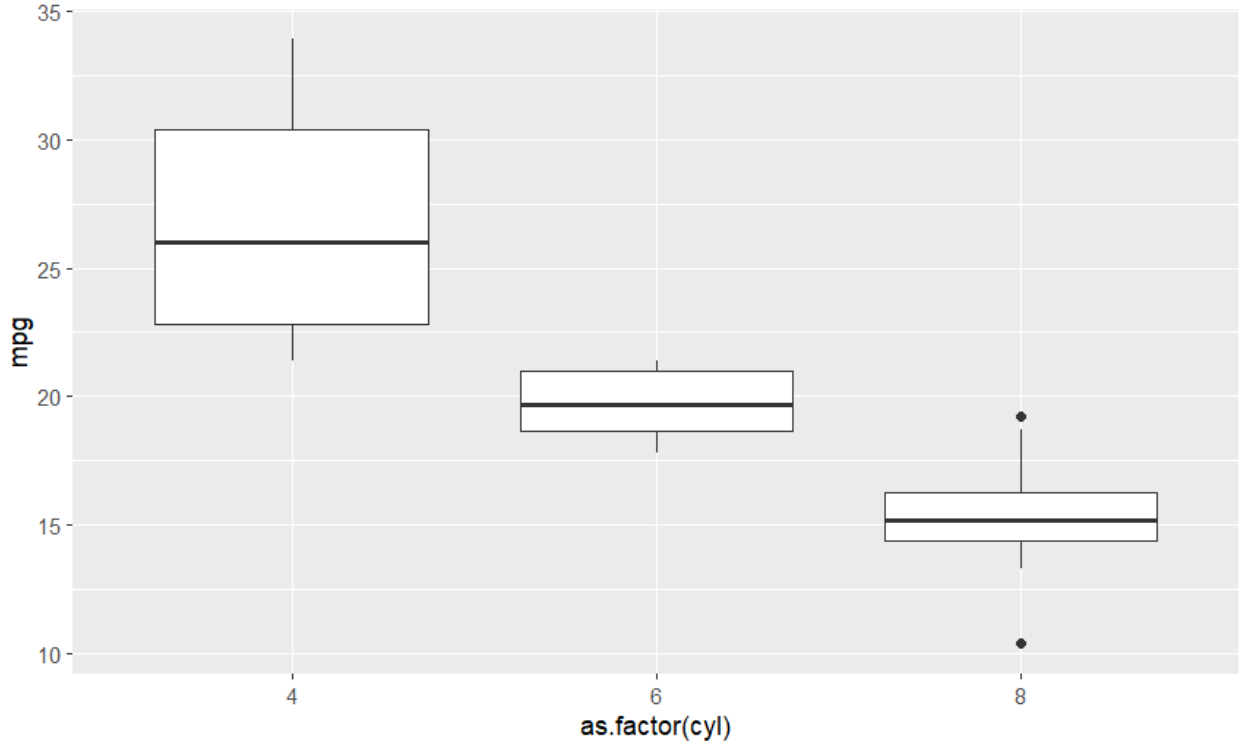
BOX PLOT

A Box plot displays the distribution of the data and skewness in the data with the help of quartile and averages.

Script

```
#box plot

ggplot(mtcars, aes(x=as.factor(cyl), y=mpg)) + geom_boxplot()
```

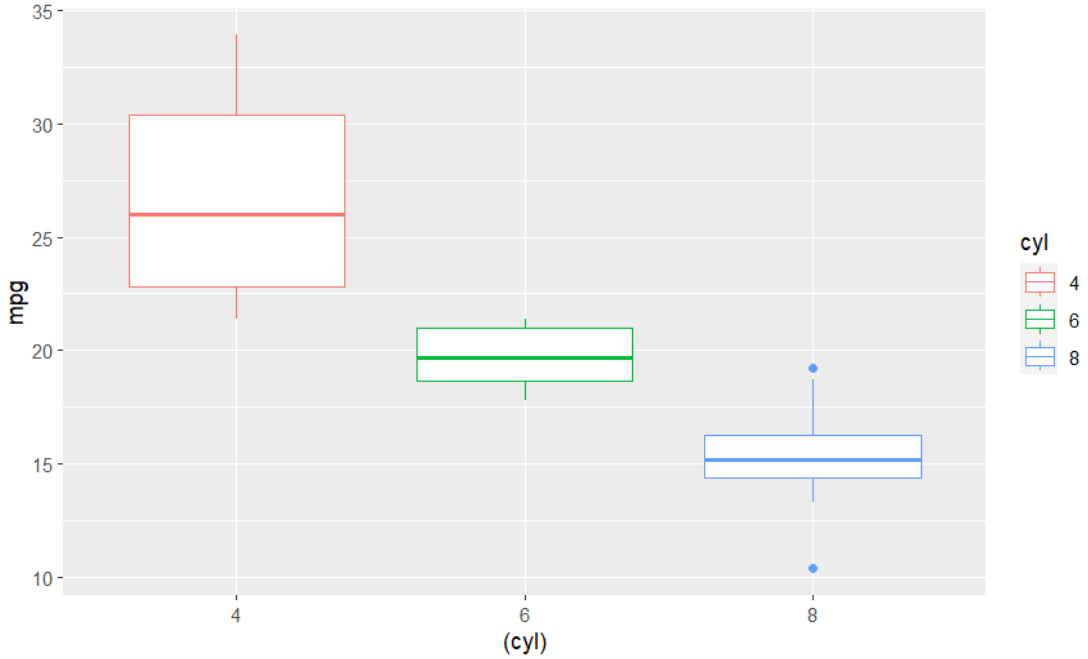
Output



Script

```
#change colours

mtcars$cyl <- as.factor(mtcars$cyl)

ggplot(mtcars, aes(x=(cyl), y=mpg,color = cyl)) + geom_boxplot()

+scale_color_manual(values = c("#3a0ca3", "#c9184a", "#3a5a40"))
```

Output



VIOLIN PLOT

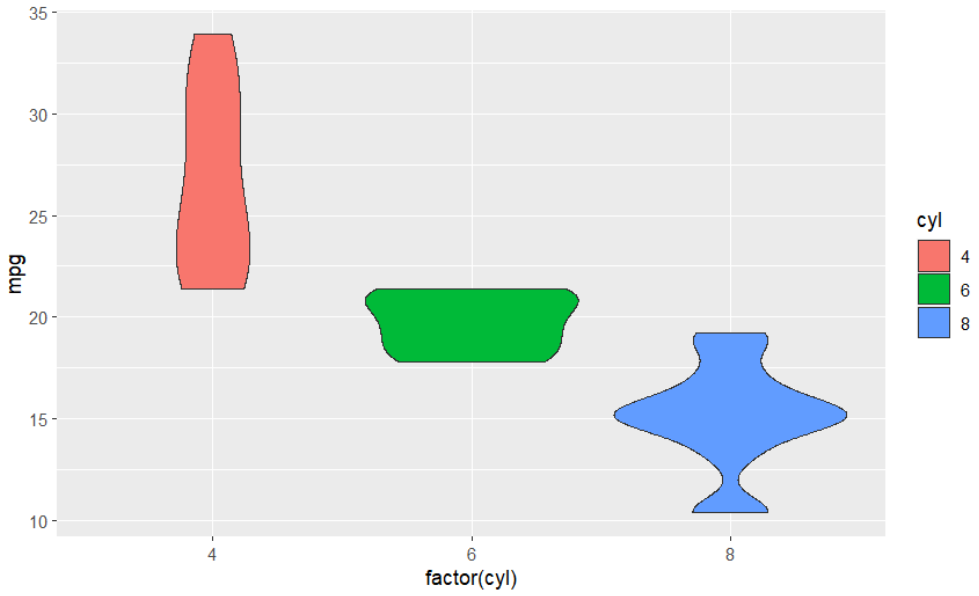
This plot is used to plot the numeric data, which is similar to a box plot and kernel density plot combination. It can show data peaks and distribution of the data.

Script

#violin plot

ggplot(mtcars, aes(factor(cyl), mpg))+ geom_violin(aes(fill = cyl))

Output



PIE CHART

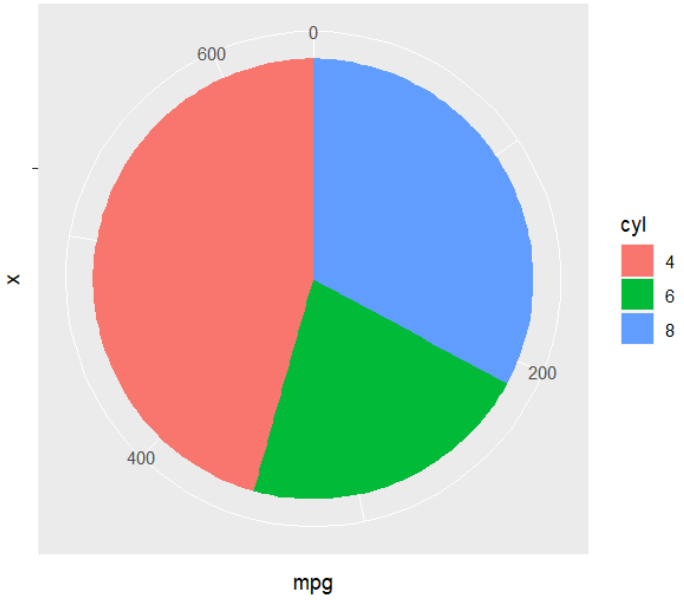
The pie chart shows the proportions as a part of the whole in the data.

Script

#pie chart

ggplot(mtcars, aes(x="", y=mpg, fill=cyl)) + geom_bar(stat="identity", width=1) + coord_polar("y", start=0)

Output



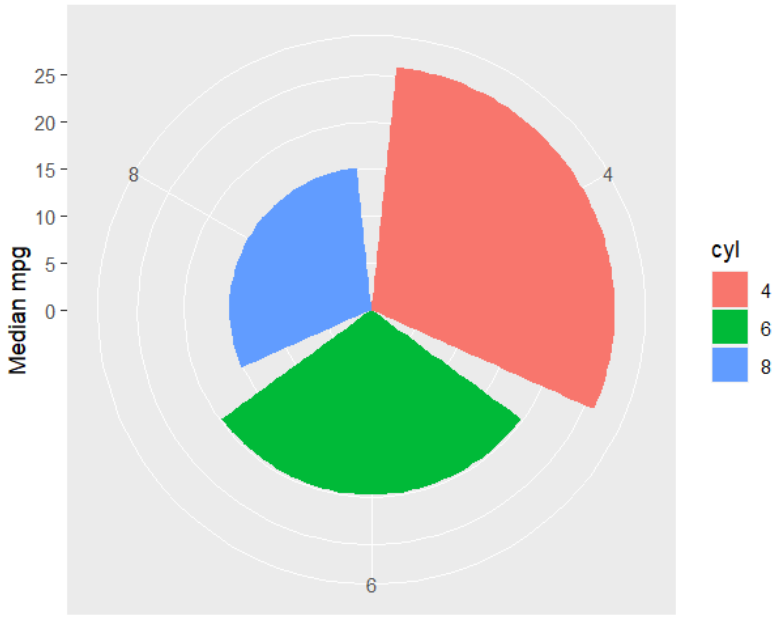
POLAR PLOT

This plot shows the magnitude value versus phase angle on polar coordinates. You can polarise the plot by using the coord_polar() function.

Script

```
#polar plot

mtcars %>%

  dplyr::group_by(cyl) %>%

  dplyr::summarize(mpg = median(mpg)) %>%

  ggplot(aes(x = cyl, y = mpg)) + geom_col(aes(fill =cyl), color = NA) + labs(x = "", y = "Median mpg") + coord_polar()
```

Output



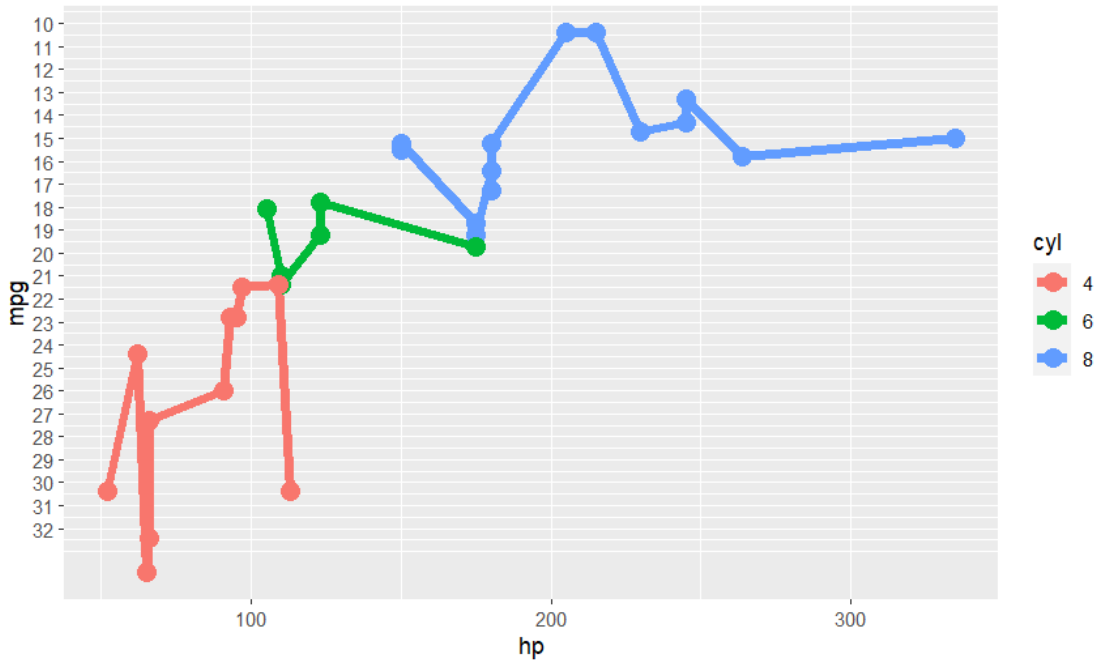
BUMP CHART

A bump chart is a type of chart that displays rankings of distinct groups over time rather than absolute numbers. This is to emphasize the order of the groups rather than the amount of change.

Script

```
#bump chart

ggplot(mtcars, aes(x = hp, y = mpg, group = cyl)) + geom_line(aes(color = cyl), size = 2) + geom_point(aes(color = cyl), size = 4) + scale_y_reverse(breaks = 1:nrow(mtcars))
```

Output



PAIR PLOT WITH GGPAIRS

The GGally provides a function called ggpairs. This ggplot2 command is similar to the basic R pairs function. A data frame holding continuous and categorical variables can be passed.Install GGally package for this and load it.

Script

#pairplot with ggpairs

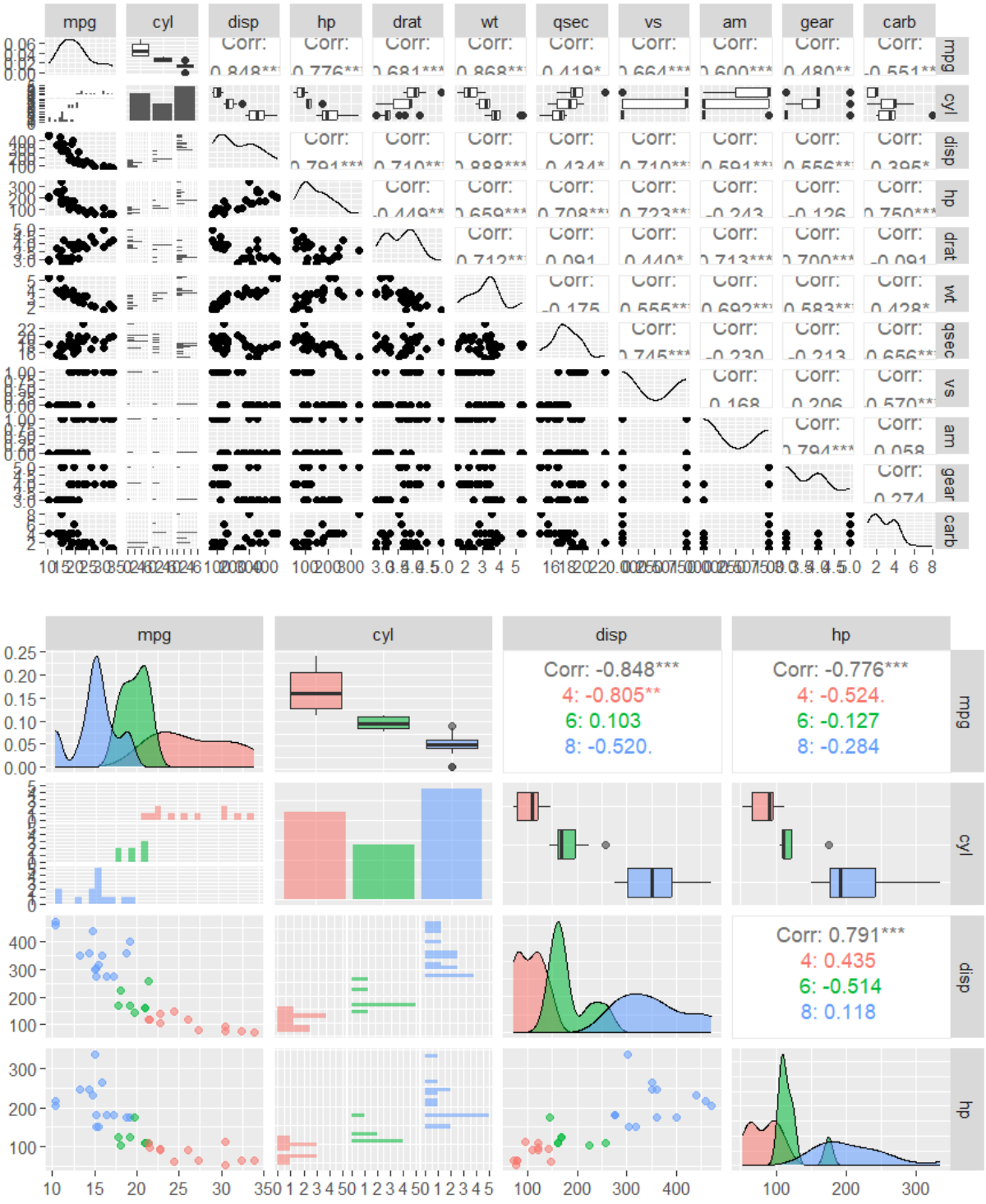
library(GGally)

ggpairs(mtcars)

#for adding colour density to it:

ggpairs(mtcars,columns = 1:4,aes(color = cyl, alpha = 0.5))

Output


mpg
cyl
disp
hp
drat
wt
qsec
vs
am
gear
carb
Corr: -0.848***
4: -0.805**
6: 0.103
8: -0.520.
Corr: -0.776***
4: -0.524.
6: -0.127
8: -0.284
Corr: 0.791***
4: 0.435
6: -0.514
8: 0.118

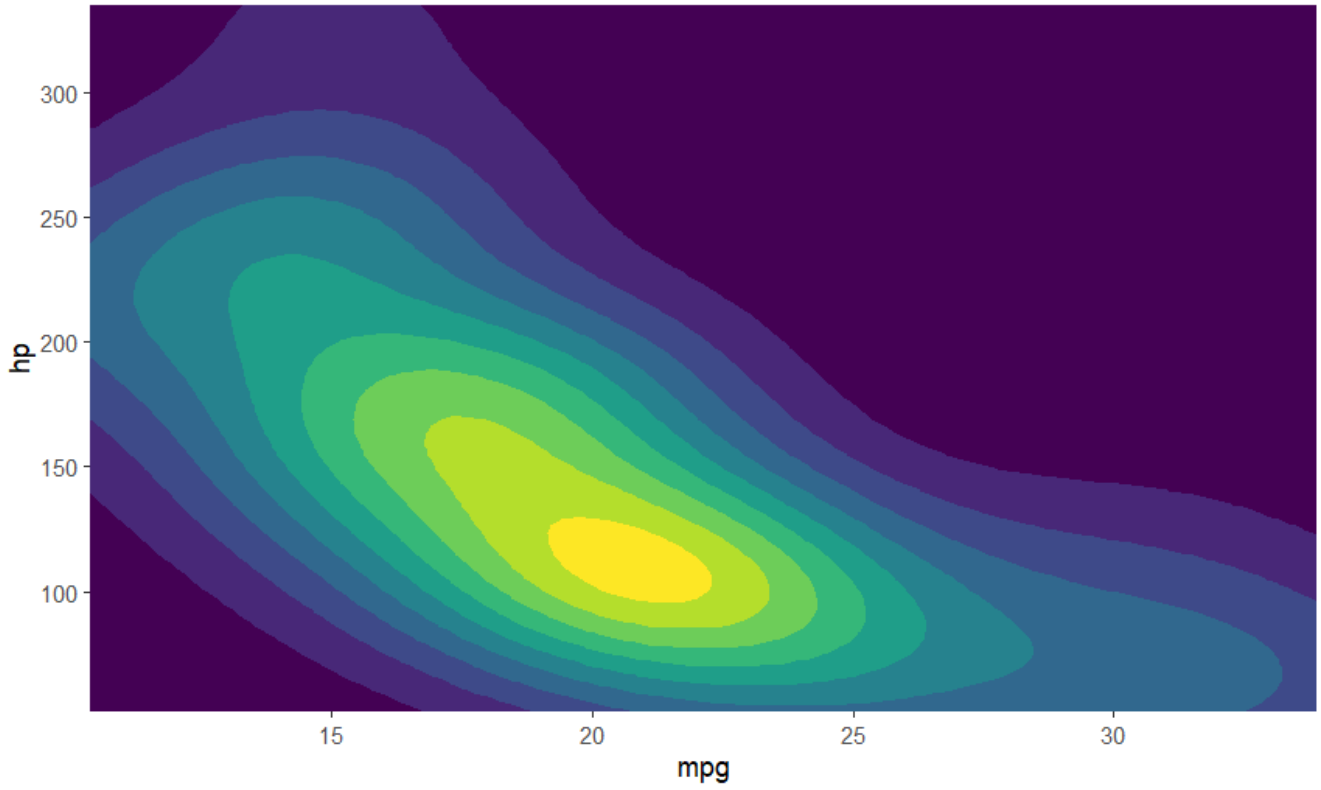
CONTOUR PLOT

ggplot2 can generate a 2D density contour plot with geom_density_2d. You only need to provide your data frame with the x and y values inside aes

Script

```
#contour plot

ggplot(mtcars, aes(mpg, hp)) + geom_density_2d_filled(show.legend = FALSE) +
coord_cartesian(expand = FALSE) + labs(x = "mpg")
```

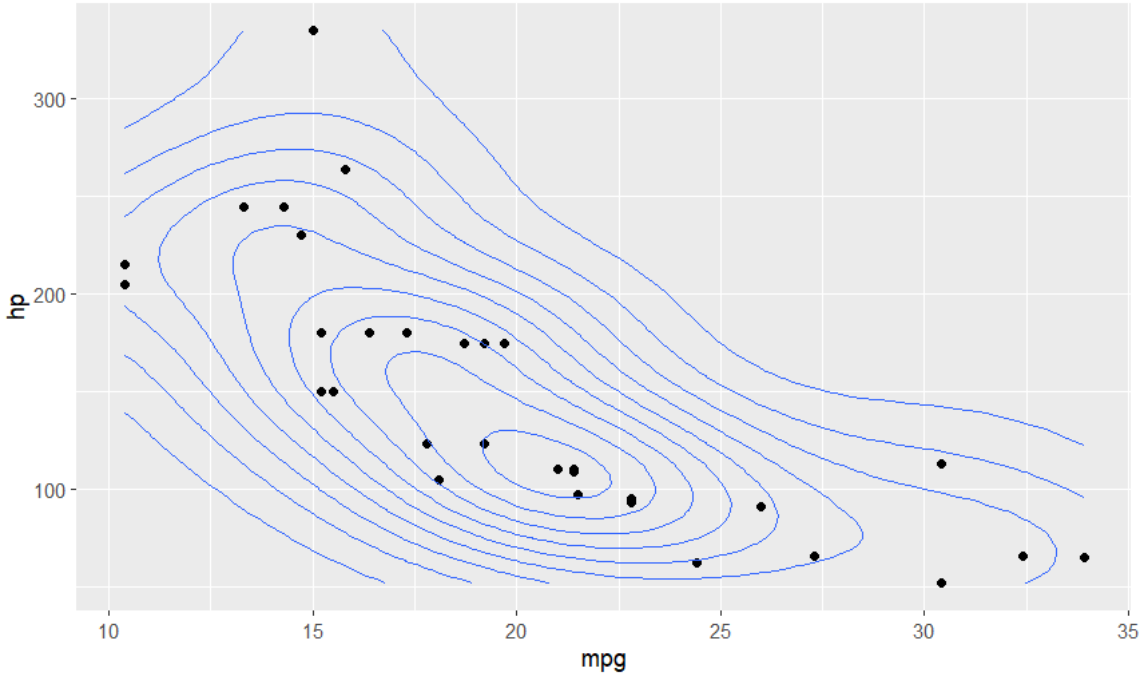
Output



Script

```
#using contour lines

ggplot(mtcars, aes(x = mpg, y = hp)) + geom_point() + geom_density_2d()
```

Output



CORRELOGRAM

A correlogram, or a correlation matrix, can be used to find the relationship between each pair of numeric variables in a dataset. It provides a high-level summary of the entire dataset. It is used for exploratory purposes rather than explanatory purposes

Script

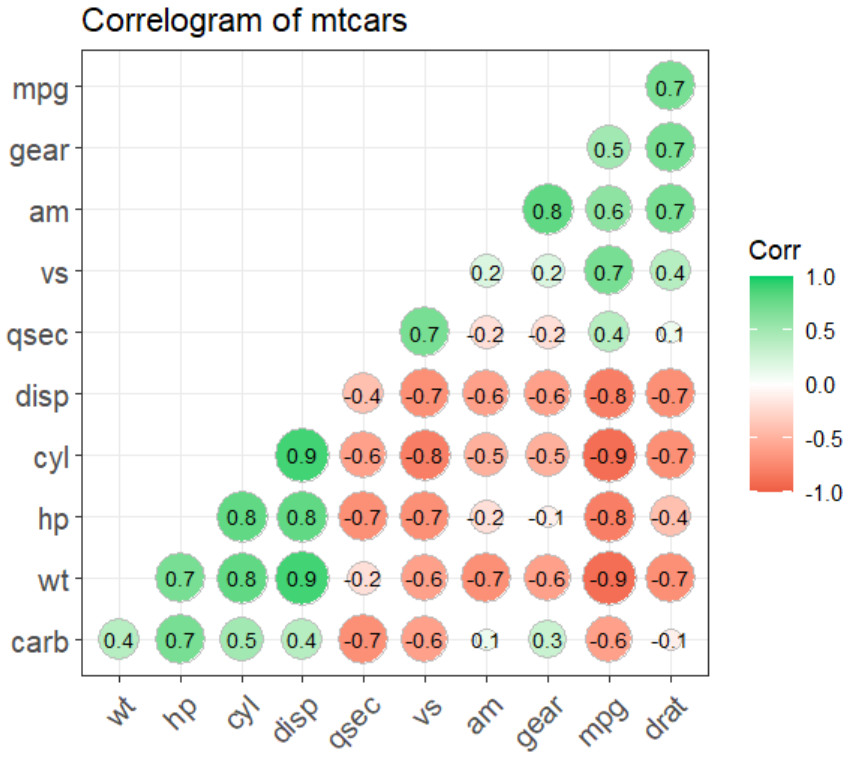
```
#CORRELOGRAM

library(ggcorrplot)

data(mtcars)

corr <- round(cor(mtcars), 1)

ggcorrplot(corr, hc.order = TRUE,

        type = "lower",

        lab = TRUE,

        lab_size = 3,

        method="circle",
```

colors = c("tomato2", "white", "springgreen3"),

title="Correlogram of mtcars",

ggtheme=theme_bw)

Output



HEATMAP

In ggplot2, a heat map can be built by supplying the categorical variables to the x and y parameters and the continuous variable to the fill argument of aes. Similar to contour maps, geom_hex() may be used to display the point counts or densities that are binned to a hexagonal grid.

Script

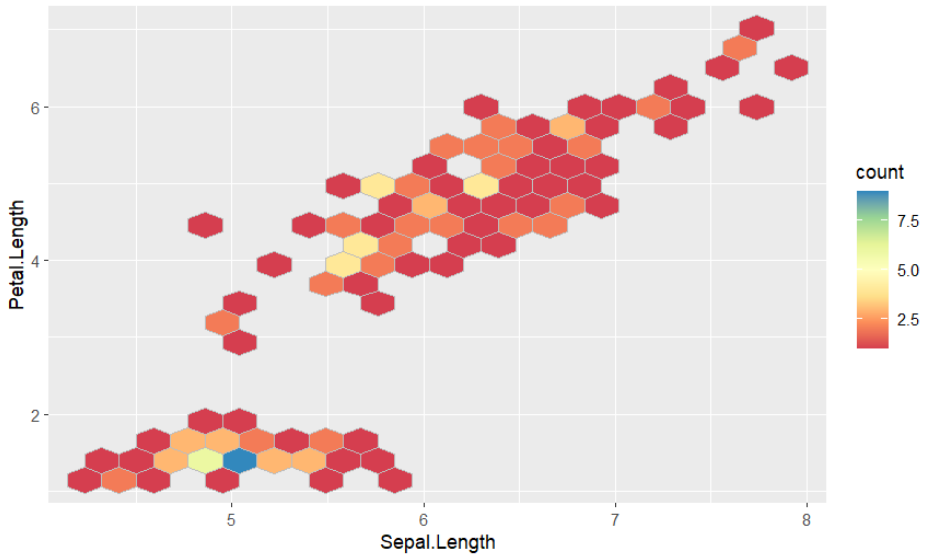
#Heatmap

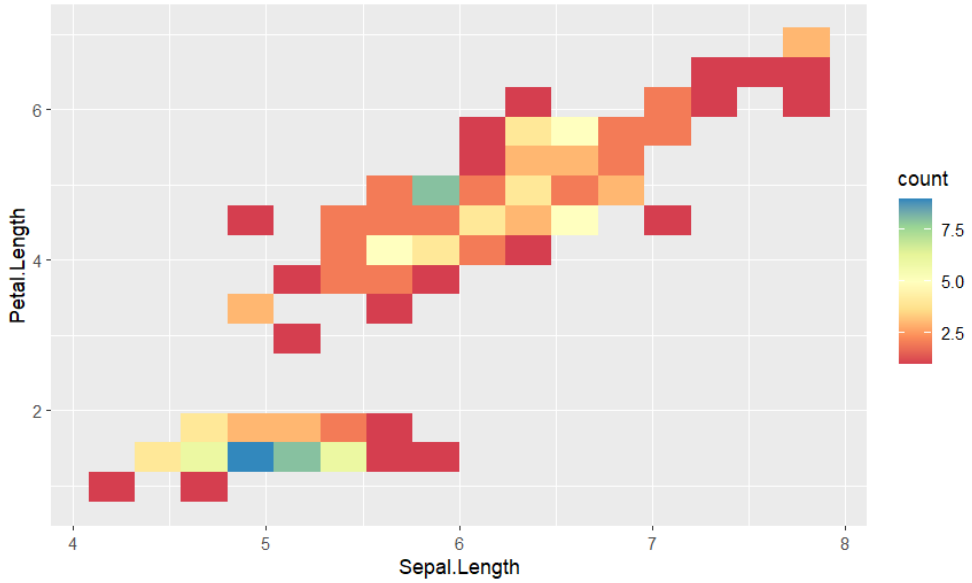
ggplot(iris, aes(Sepal.Length, Petal.Length)) + geom_hex(bins = 20, color = "grey") + scale_fill_distiller(palette = "Spectral", direction = 1)

#Regular rectangular grids

ggplot(iris, aes(Sepal.Length, Petal.Length)) + geom_bin2d(bins = 15) + scale_fill_distiller(palette = "Spectral", direction = 1)

Output





RESULT

Thus, the plotting of data and data visualization by ‘ggplot’ package has been done successfully in R.