ggplot()

ggplot() initializes a ggplot object. It can be used to declare the input data frame for a graphic and to specify the set of plot aesthetics intended to be common throughout all subsequent layers unless specifically overridden.

ggplot() is used to construct the initial plot object, and is almost always followed by + to add component to the plot. There are three common ways to invoke ggplot:

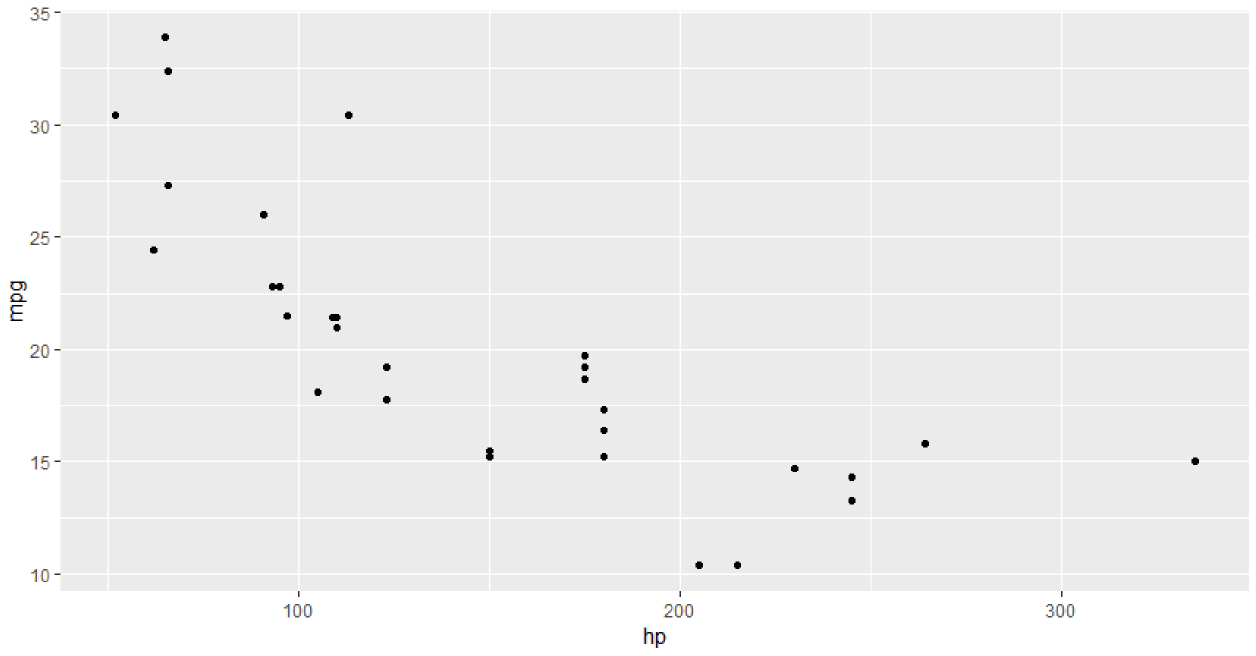
- ggplot(df, aes(x, y, other aesthetics))
- ggplot(df)
- ggplot()

The first method is recommended if all layers use the same data and the same set of aesthetics, although this method can also be used to add a layer using data from another data frame. See the first example below. The second method specifies the default data frame to use for the plot, but no aesthetics are defined up front. This is useful when one data frame is used predominantly as layers are added, but the aesthetics may vary from one layer to another. The third method initializes a skeleton ggplot object which is fleshed out as layers are added. This method is useful when multiple data frames are used to produce different layers, as is often the case in complex graphics.

Examples

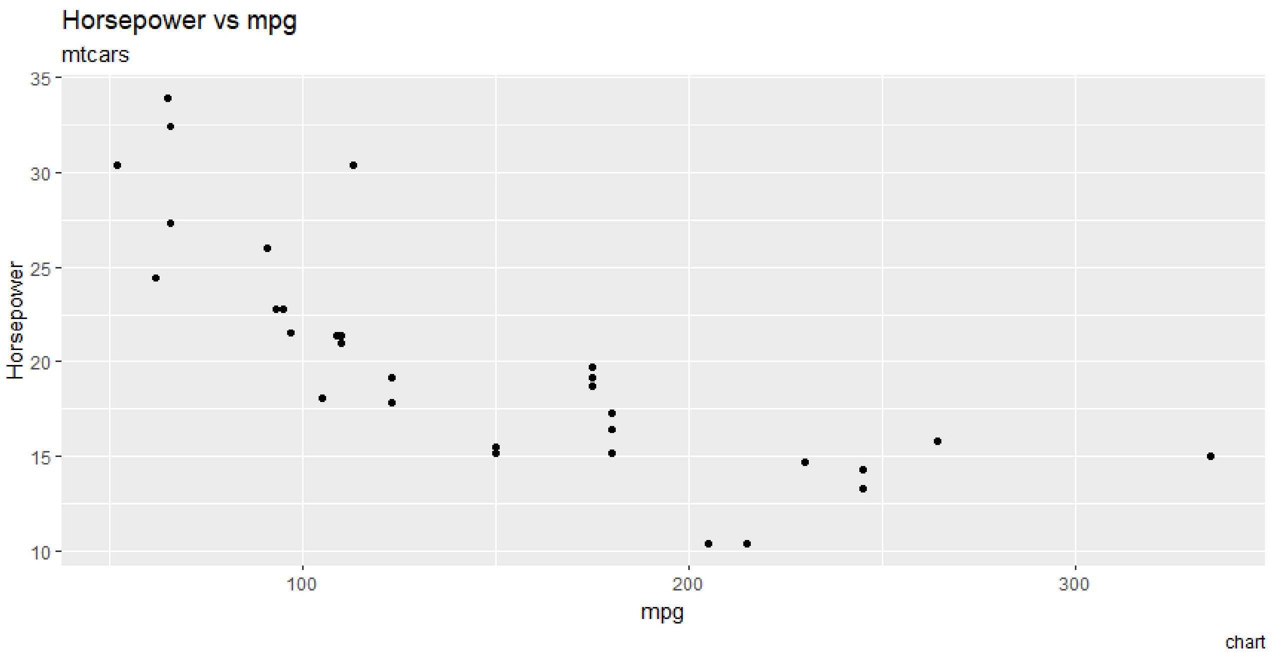
```
install.packages("ggplot2")
install.packages("tidyverse")
library(tidyverse)
library(ggplot2)
```

```
#Draw scatter plots with ggplot2
g <- ggplot(mtcars, aes(x=hp, y=mpg))
g+geom_point()
```



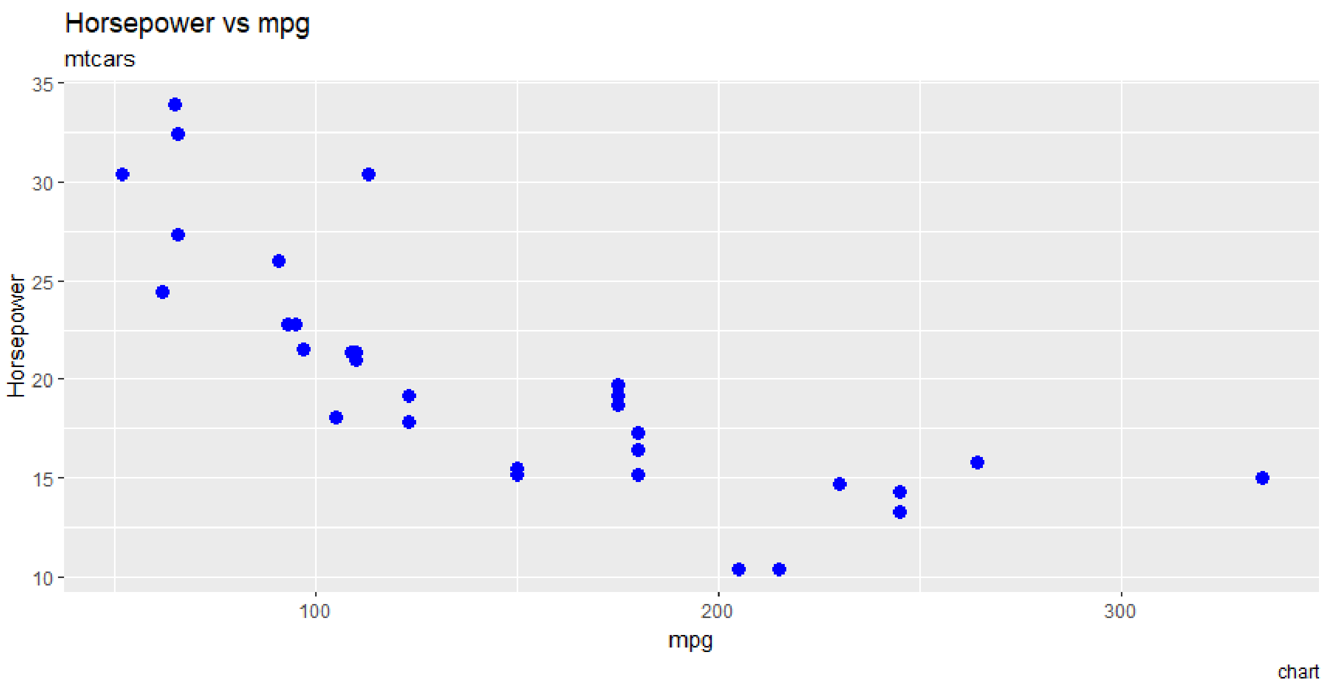
# Add Labels

```
g+geom_point() + labs(title="Horsepower vs mpg", subtitle="mtcars", y="Horsepower", x="mpg",
caption="chart")
```



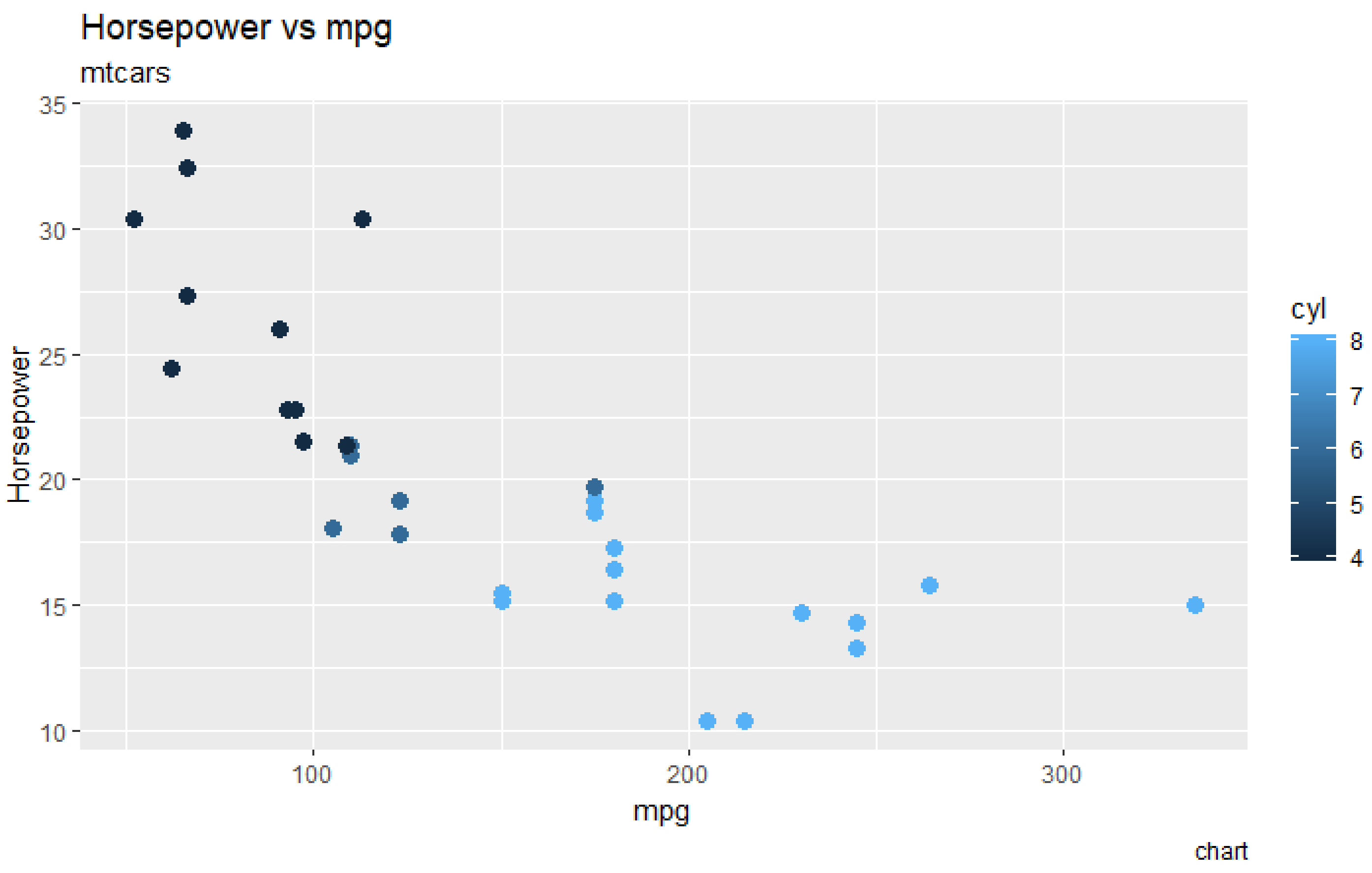
# Change the size and color of the points

g+geom_point(col="blue", size=3) + labs(title="Horsepower vs mpg", subtitle="mtcars", y="Horsepower", x="mpg", caption="chart"



# Use color to code another column "cyl"

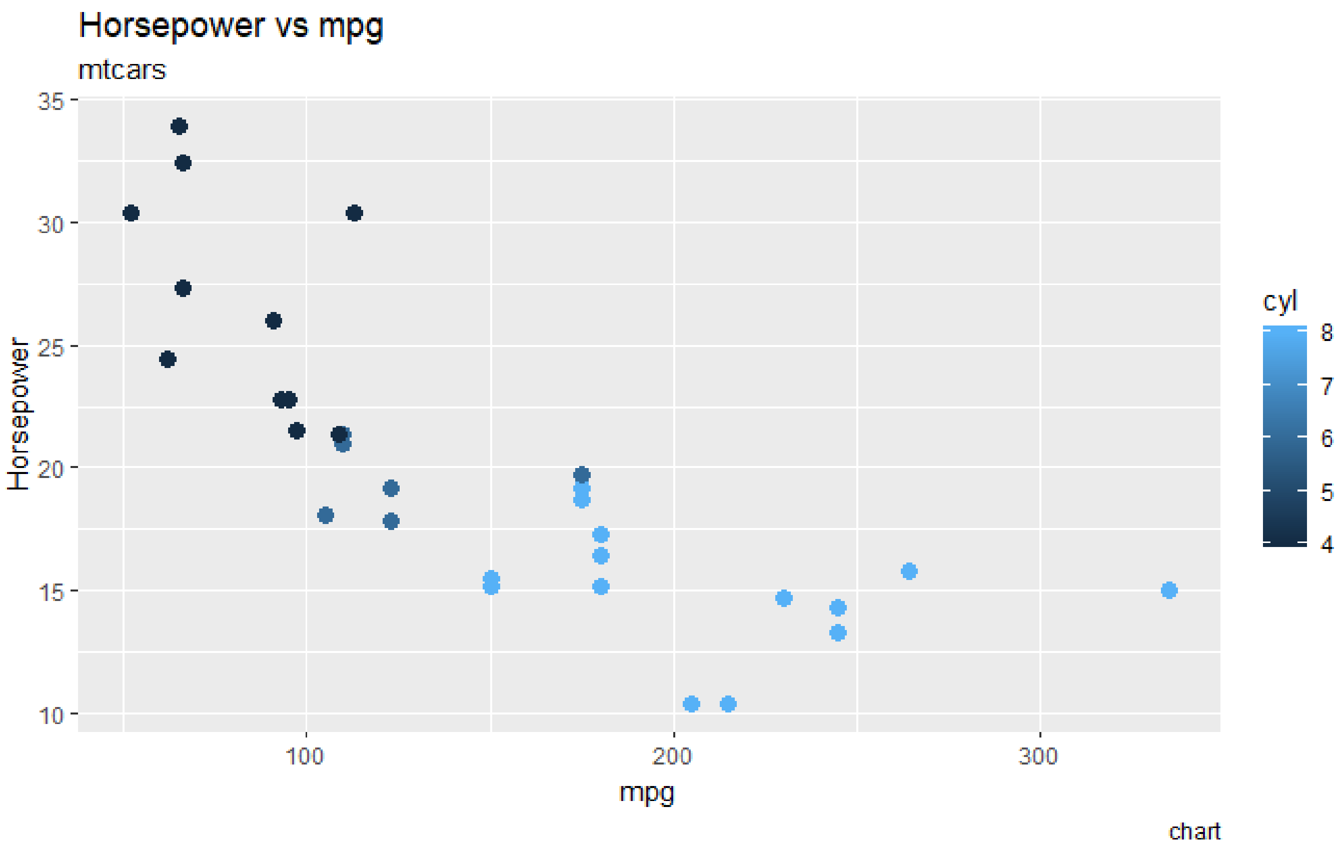
g+geom_point(aes(col=cyl), size=3) + labs(title="Horsepower vs mpg", subtitle="mtcars", y="Horsepower", x="mpg", caption="chart")



Change the theme of the chart

gg <- g+geom_point(aes(col=cyl), size=3) + labs(title="Horsepower vs mpg", subtitle="mtcars", y="Horsepower", x="mpg", caption="chart")

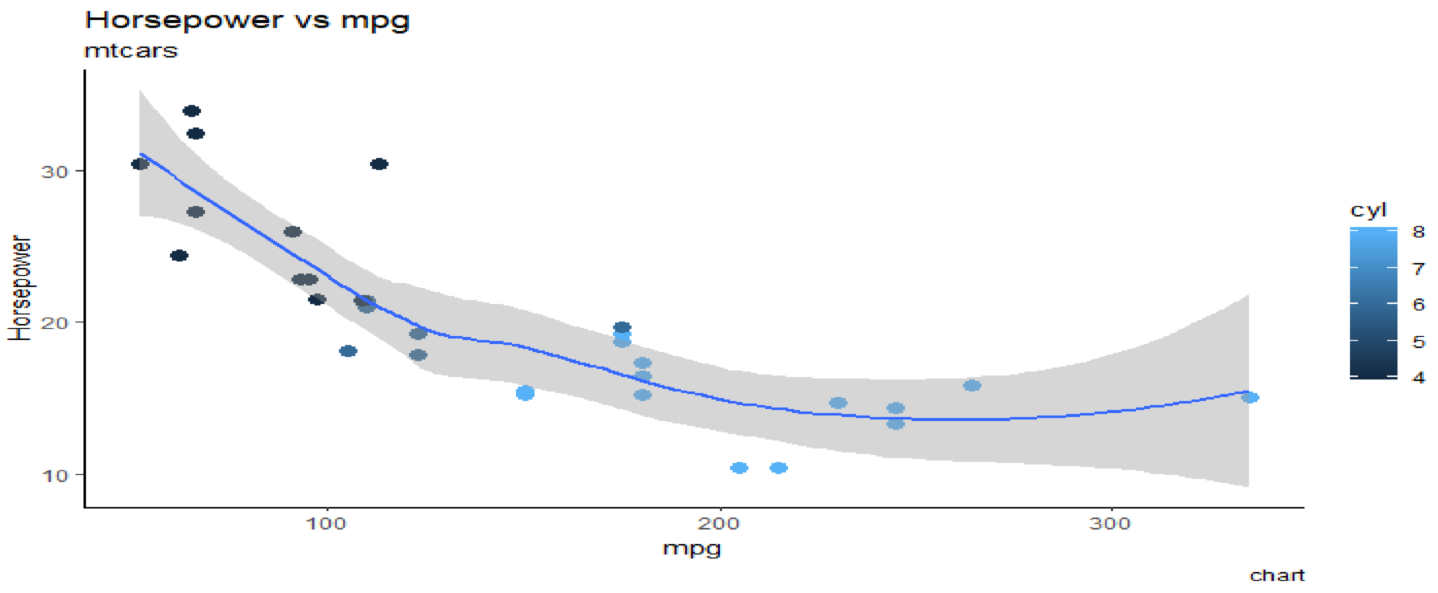
gg+ theme_classic()



#Draw a smoothing line based on losses

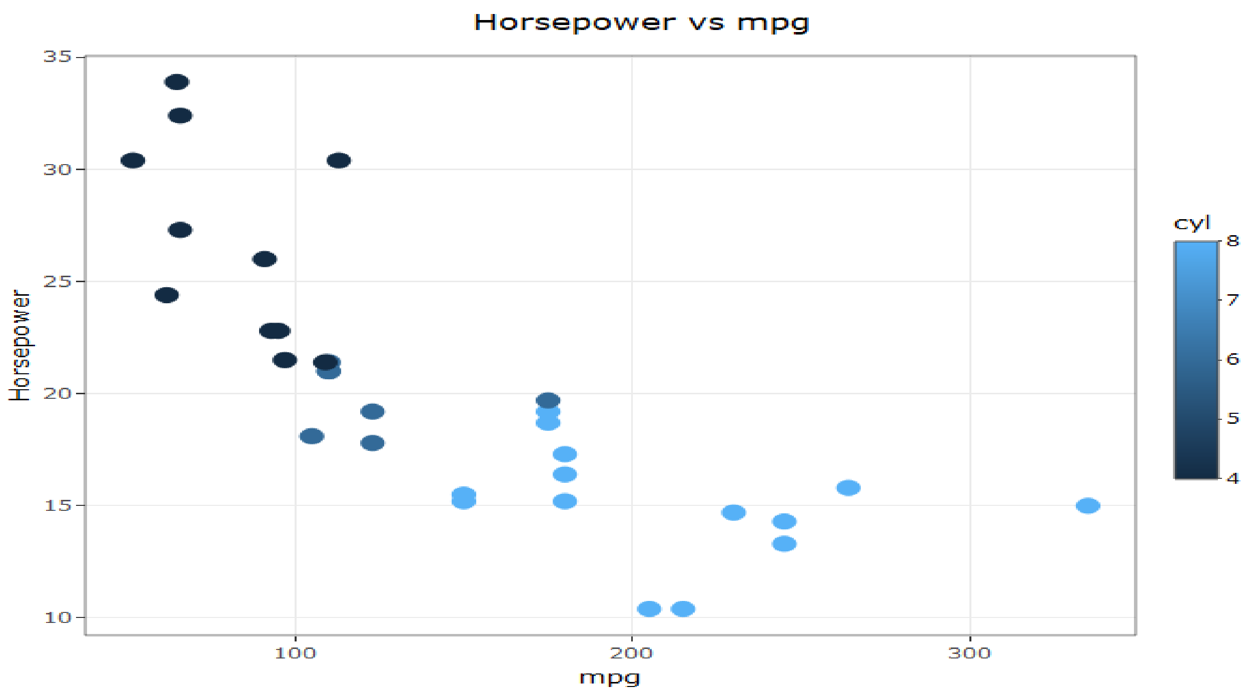
gg <- g+geom_point(aes(col=cyl), size=3) + labs(title="Horsepower vs mpg", subtitle="mtcars", y="Horsepower", x="mpg", caption="chart")

gg+ theme_classic() + geom_smooth()

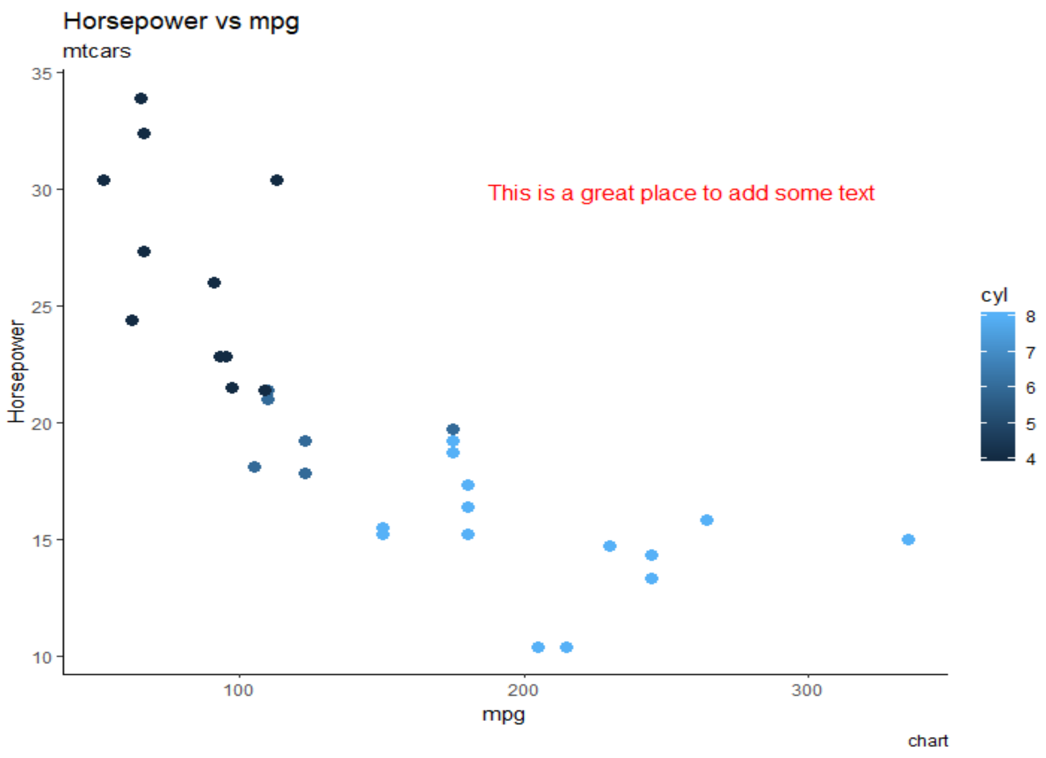


Now lets make it interactive

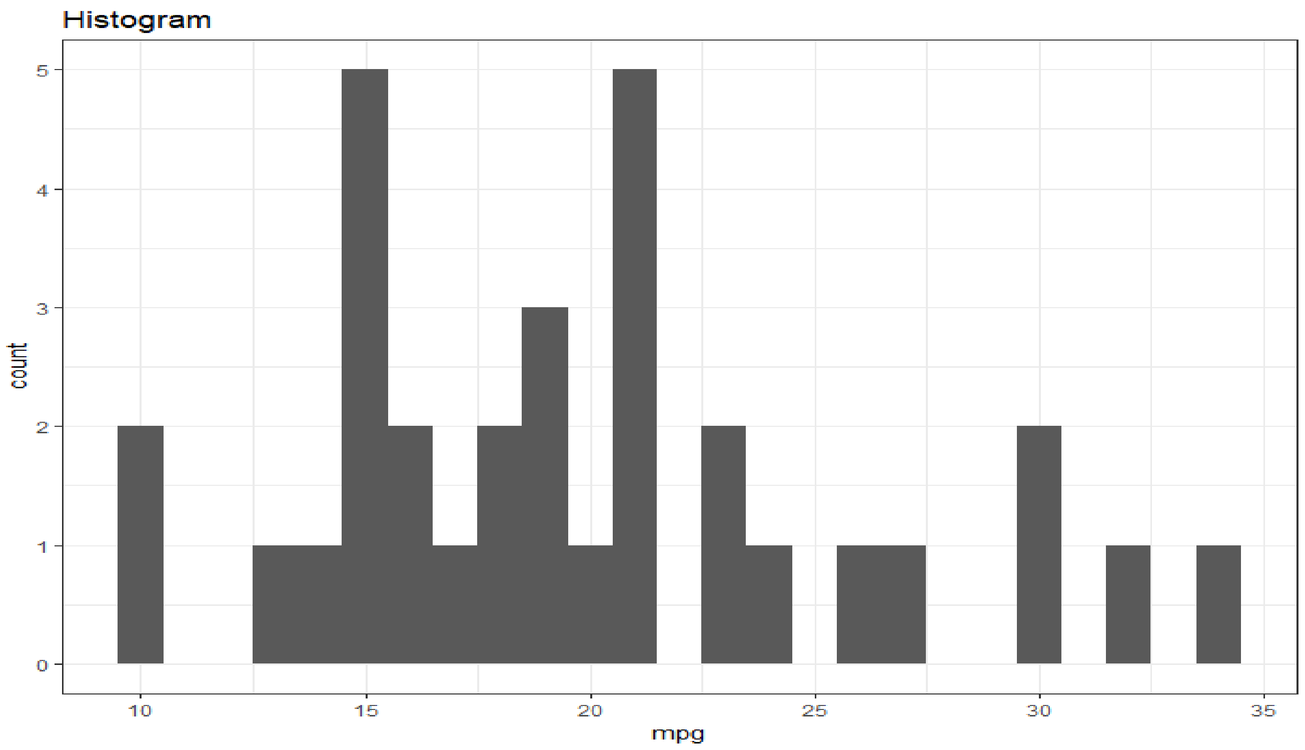
```
install.packages("plotly")
library(plotly)
ggplotly(gg)
```



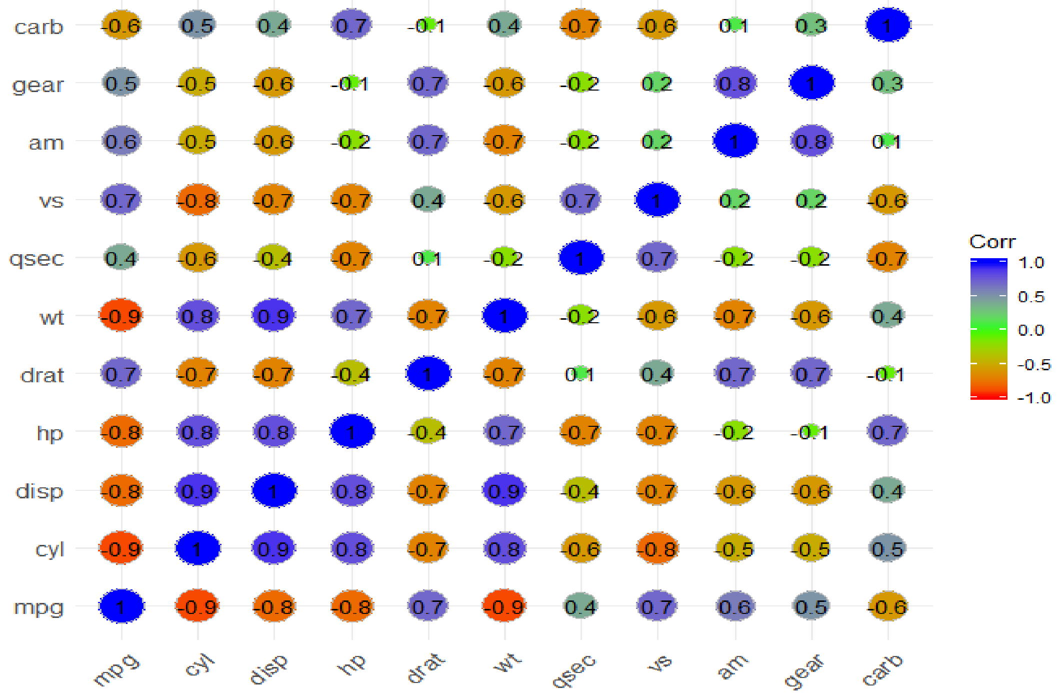
```
# Add annotation to the chart
library(grid)
text <- "This is a great place to add some text"
grob <- grid.text(text, x=0.7, y=0.8, gp=gpar(col="red", fontsize=12))
gg + theme_classic() + annotation_custom(grob)
```



```
#Draw a histogram
g3 <- ggplot(mtcars, aes(mpg))
g3+geom_histogram(binwidth=1)+labs(title="Histogram")
```



```
install.packages("ggcorrplot")
library(ggcorrplot)
data(mtcars)
corr <- round(cor(mtcars), 1)
ggcorrplot(corr, method="circle", colors=c("red", "green", "blue"), lab=TRUE)
```


Corr
mpg
cyl
disp
hp
drat
wt
qsec
vs
am
gear
carb