

An introduction to ggplot2 with R

Geovisualization

Joaquin Cavieres

1. Using the “ggplot2” package in R

To start with the “ggplot2” package in R, first you have to install it by running the command:

```
#install.packages("ggplot2")
```

Once installed, load the ggplot2 package into your R session using the command

```
library(ggplot2)
```

and that is! now you can use the “ggplot2” package for data visualization.

2. The `ggplot()` function.

The basic arguments of the `ggplot()` function are:

- Data: The data set that you want to plot.
- Geometries \Rightarrow `geom_` function: The geometric shapes that will represent the data.
- Aesthetics \Rightarrow `aes()`: Aesthetics of the geometric and statistical objects, such as position, color, size, shape, and transparency
- Scales \Rightarrow `scale_`: Maps between the data and the aesthetic dimensions, such as data range to plot width or factor values to colors.
- Statistical transformations \Rightarrow `stat_`: Statistical summaries of the data, such as quantiles, fitted curves, and sums.
- Coordinate system \Rightarrow `coord_`: The transformation used for mapping data coordinates into the plane of the data rectangle.
- Facets \Rightarrow `facet_`: The arrangement of the data into a grid of plots.
- Visual themes \Rightarrow `theme()`: The overall visual defaults of a plot, such as background, grids, axes, default typeface, sizes and colors.

The “mtcars” data

We will use the “mtcars” dataset in R. The data was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models).

```
data(mtcars)
```

We can take a look at the first six rows of the dataset by using the `head()` function:

```
head(mtcars)
```

```
##           mpg  cyl  disp  hp drat   wt  qsec vs am gear carb
## Mazda RX4      21.0    6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag  21.0    6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710      22.8    4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive  21.4    6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7    8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant         18.1    6  225 105 2.76 3.460 20.22  1  0    3    1
```

We also can use the `summary()` function to quickly summarize each variable in the dataset:

```
summary(mtcars)
```

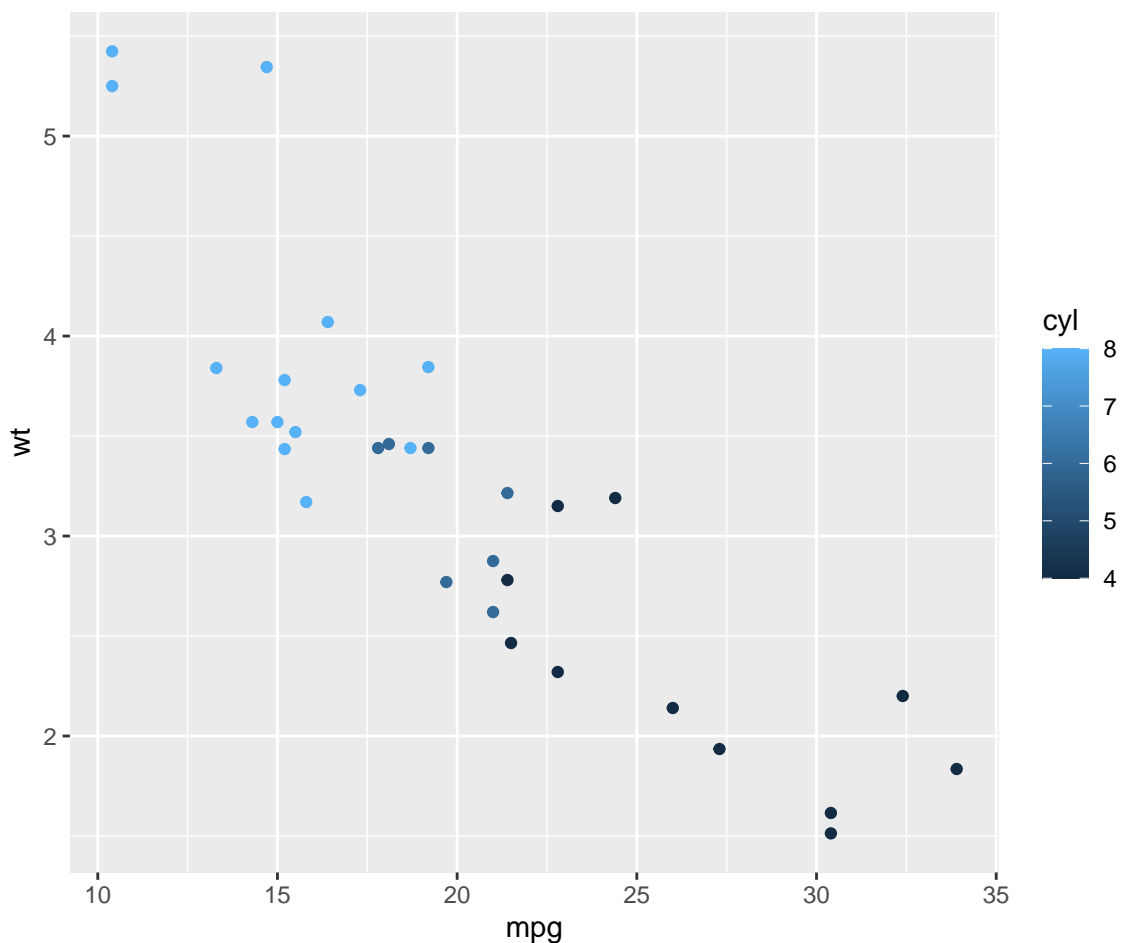
```
##           mpg           cyl           disp           hp
##  Min.      :10.40   Min.      :4.000   Min.      : 71.1   Min.      : 52.0
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5
##  Median :19.20   Median :6.000   Median :196.3   Median :123.0
##  Mean     :20.09   Mean     :6.188   Mean     :230.7   Mean     :146.7
##  3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0
##  Max.     :33.90   Max.     :8.000   Max.     :472.0   Max.     :335.0
##           drat           wt           qsec           vs
##  Min.      :2.760   Min.      :1.513   Min.      :14.50   Min.      :0.0000
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000
##  Median :3.695   Median :3.325   Median :17.71   Median :0.0000
##  Mean     :3.597   Mean     :3.217   Mean     :17.85   Mean     :0.4375
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000
##  Max.     :4.930   Max.     :5.424   Max.     :22.90   Max.     :1.0000
##           am           gear           carb
##  Min.      :0.0000   Min.      :3.000   Min.      :1.000
##  1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000
##  Median :0.0000   Median :4.000   Median :2.000
##  Mean     :0.4062   Mean     :3.688   Mean     :2.812
##  3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000
##  Max.     :1.0000   Max.     :5.000   Max.     :8.000
```

How can we know the dimension of the data? how can we know the names of the variables in the dataset?

3. Creating a basic plot with the `ggplot()` function

We have to use the `ggplot()` function to specify the dataset as the first argument, and then we can add a geometric object to the plot using the `geom_point()` function like this:

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



the aesthetic mappings were done using the `aes()` function, using the variables `x`, `y`, `color`.

3.1. Customize the plot

We can customize our plot by adding additional layers or modifying aesthetics. For example, we can use the `labs()` to add labels to the axes or change the plot title using the `ggtitle()` function.

```
ggplot(data = mtcars) +  
  geom_point(aes(x = mpg, y = wt, color = cyl)) +  
  labs(title = "mpg vs weight", x = "Miles per gallon", y = "Weight") +  
  scale_color_manual(values = c("red", "blue", "green"))
```

To solve this, we have to convert our grouping variable to the factor class using the `factor()` function.

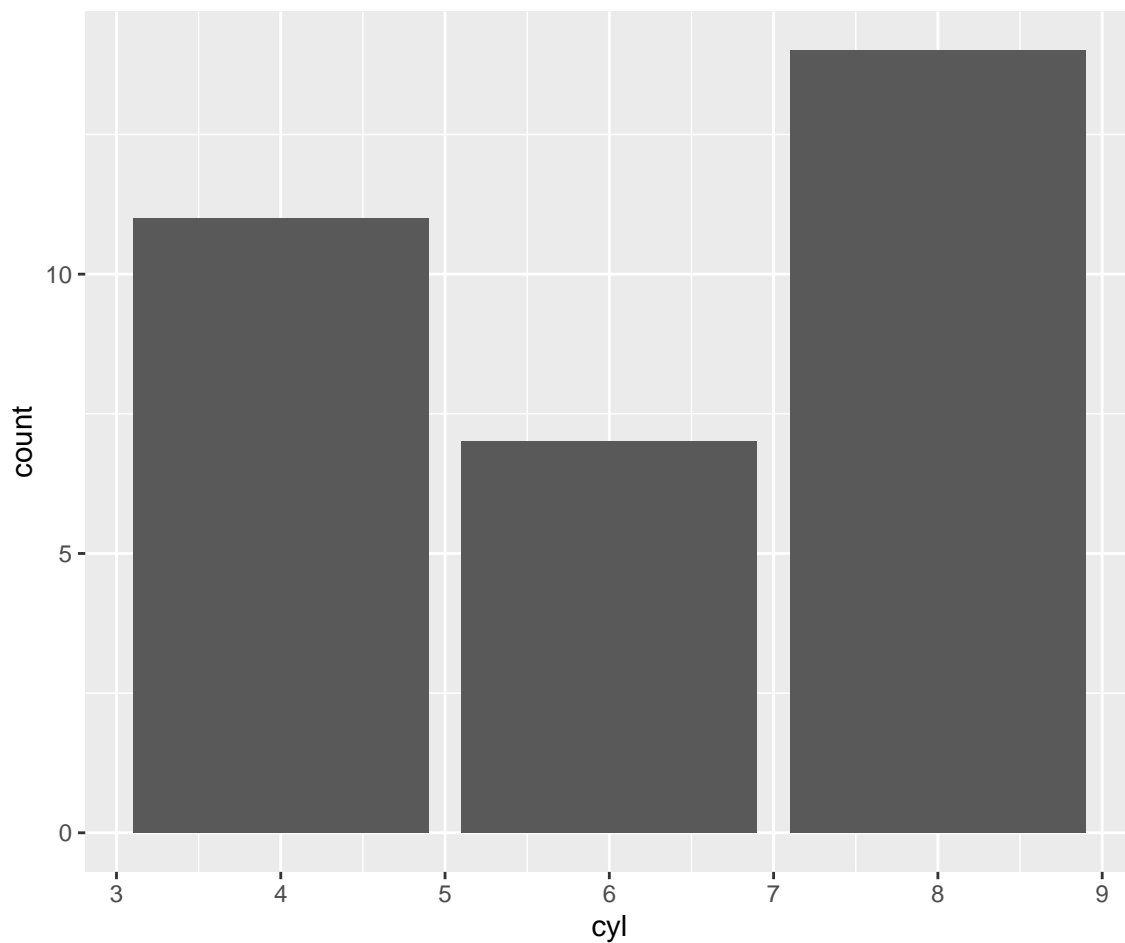
```
ggplot(data = mtcars) +  
  geom_point(aes(x = mpg, y = wt, color = factor(cyl))) +  
  labs(title = "mpg vs weight", x = "Miles per gallon", y = "Weight") +  
  scale_color_manual(values = c("red", "blue", "green"))
```



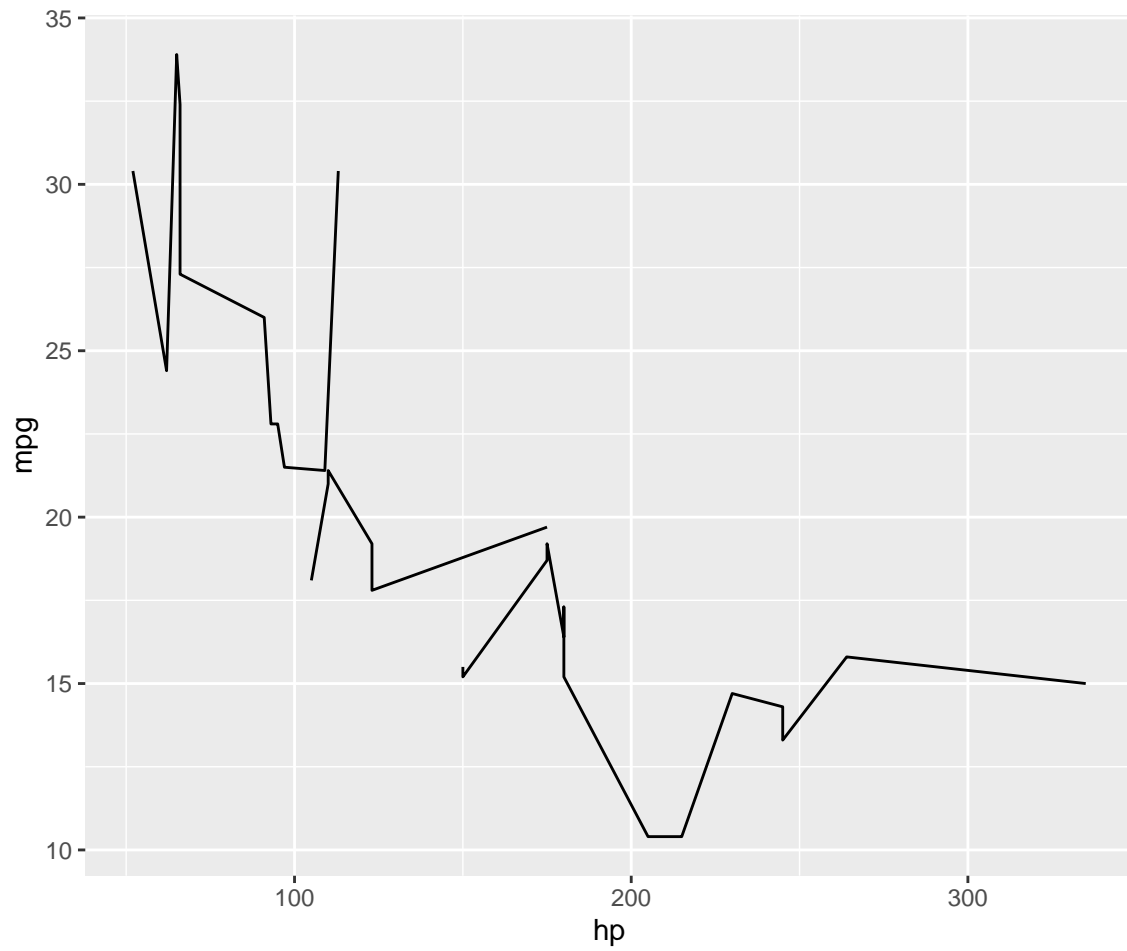
3.2. Exploring more type of plots

There are other types of plots, as for example: bar charts, line plots, histograms, or box plots. The corresponding geometries for those plots are: `geom_bar()`, `geom_line()`, `geom_histogram()`, or `geom_boxplot()`.

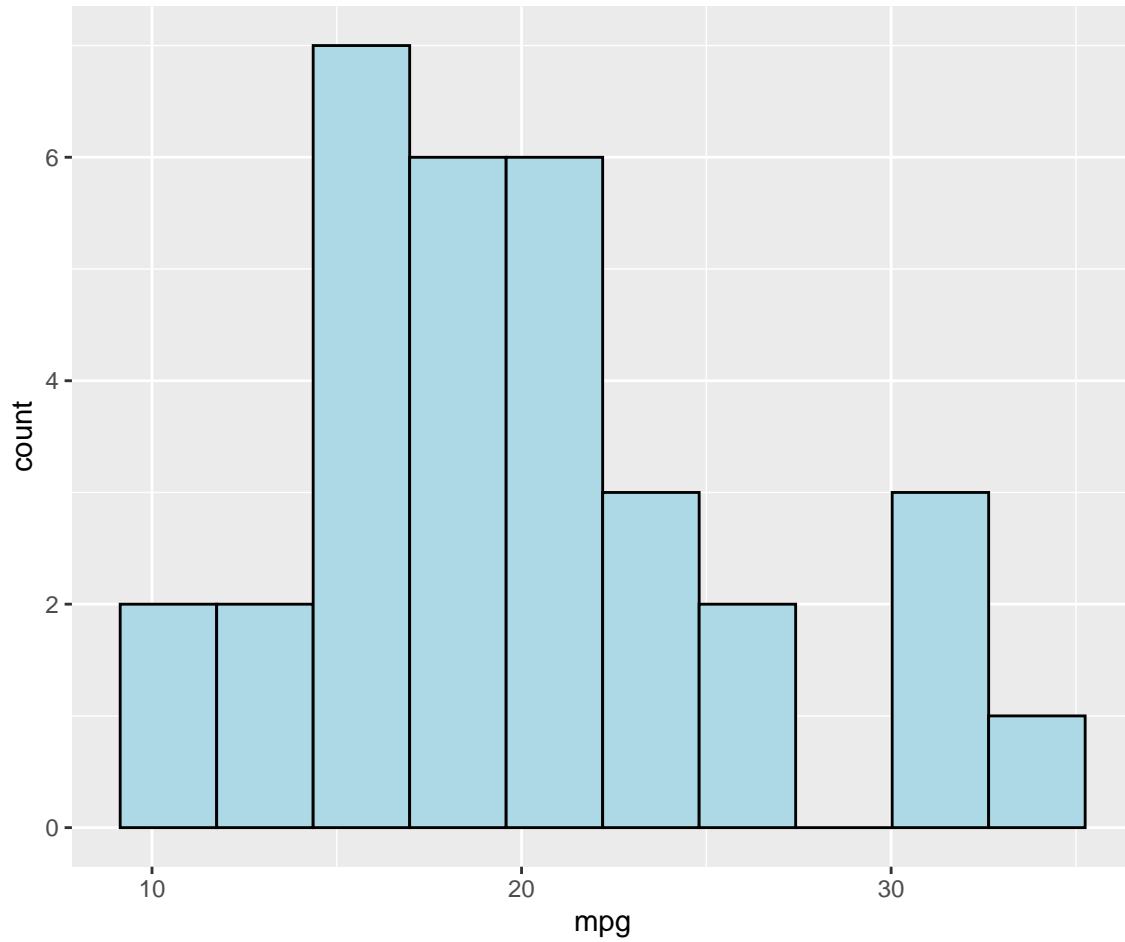
```
# Bar Chart  
ggplot(data = mtcars) +  
  geom_bar(aes(x = cyl, fill = gear), position = "dodge")
```



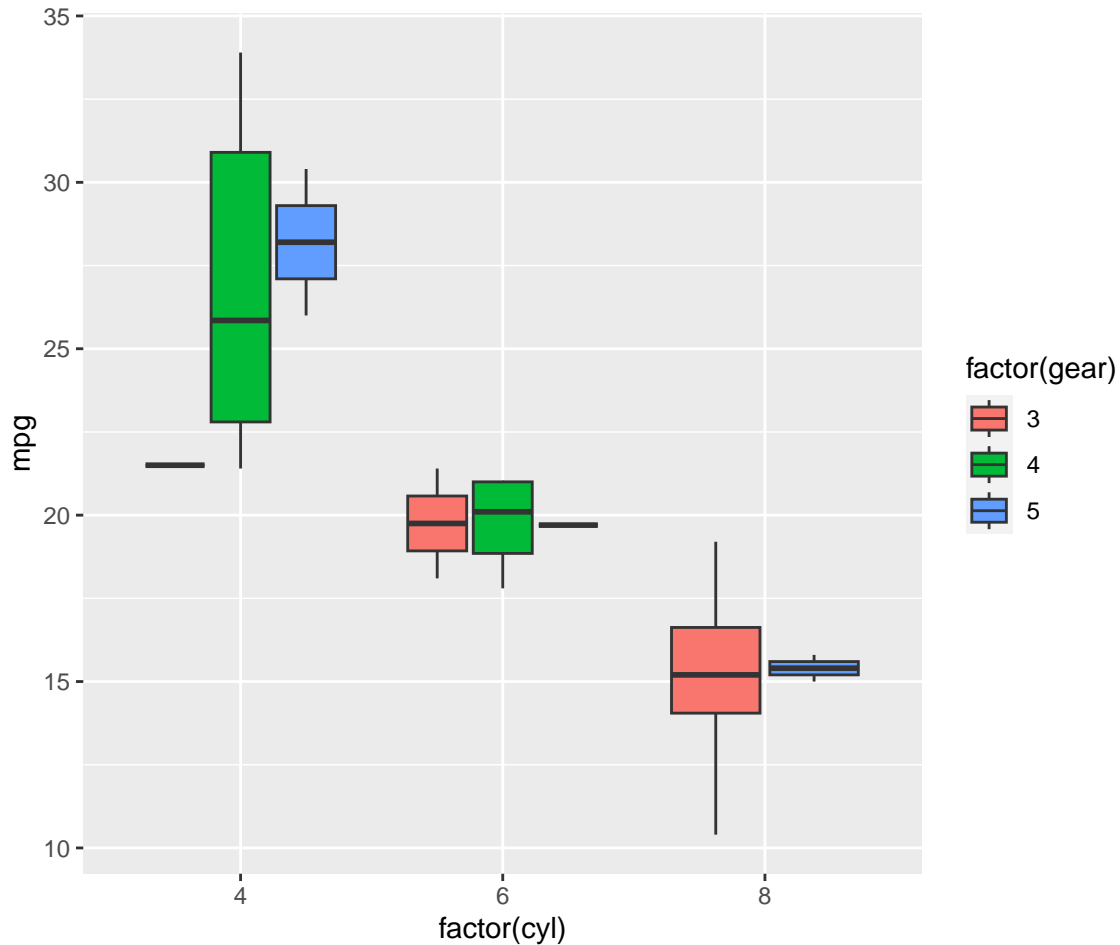
```
# Line Plot  
ggplot(data = mtcars) +  
  geom_line(aes(x = hp, y = mpg, group = cyl))
```



```
# Histogram  
ggplot(data = mtcars) +  
  geom_histogram(aes(x = mpg), bins = 10, fill = "lightblue", color = "black")
```



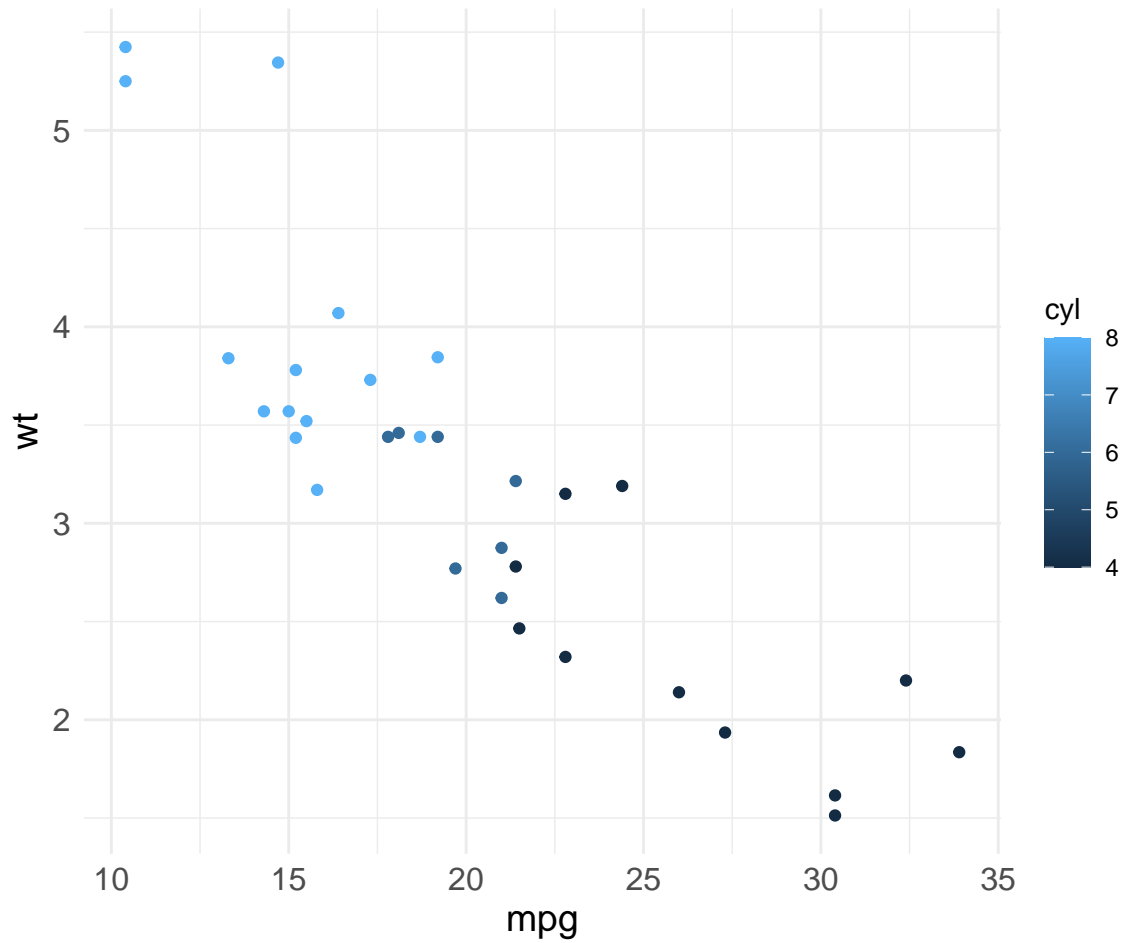
```
# Box Plot
ggplot(data = mtcars) +
  geom_boxplot(aes(x = factor(cyl), y = mpg, fill = factor(gear)))
```



3.3. Customize themes and appearance

Customize the appearance of plots using themes, which control various visual aspects. For this we can use the functions `theme_minimal()`, `theme_bw()`, or `theme_classic()` to change the overall plot style.

```
ggplot(data = mtcars) +
  geom_point(aes(x = mpg, y = wt, color = cyl)) +
  theme_minimal() +
  theme(plot.title = element_text(size = 16, face = "bold"),
        axis.text = element_text(size = 12),
        axis.title = element_text(size = 14))
```

3.4. Save and export the plot

Once your plot represent the idea that you wanted to show, then you can save it as an image file using the `ggsave()` function, for example doing the following:

```
ggsave("plot.png", width = 6, height = 4, dpi = 300)
```

4. Easily data visualization with the “esquisse” package

The “esquisse” package provides an intuitive and accessible way to create “ggplot2” visualizations through its graphical user interface. It simplifies the process of creating plots and allows users to experiment with different aesthetics and plot types without extensive coding knowledge.

```
# devtools::install_github("dreamRs/esquisse")  
# library(esquisse)
```

[See the examples in the R code.](#)

References

- r-statistics.co: <http://r-statistics.co/>
- ggplot2: Elegant Graphics for Data Analysis. <https://ggplot2-book.org/>.