Rsquared Academy

Introduction to ggplot2

Agenda

- explore graphics systems in R
- understand grammar of graphics
- understand geoms
- explore aesthetics to modify geoms
- use facets for sub plots
- change coordinate system

Why visualize data?

- explore and explain patterns and trends
- detect anomalies in data
- make sense of large amount of data efficiently and in time
- easy to communicate and share insights from data

Graphics Systems

- Base R
- ggplot2
- lattice

Libraries

```r
library(ggplot2)
library(readr)
library(descriptr)
library(gridExtra)
```

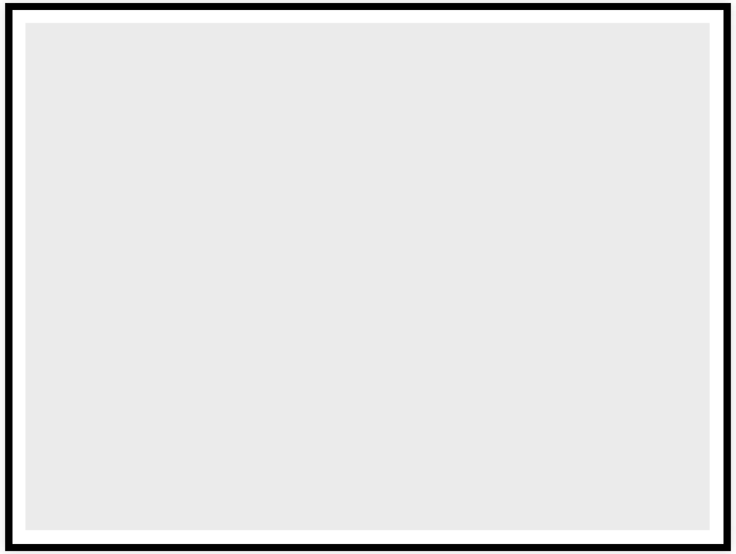
Grammar of Graphics

Grammar of graphics is a formal system for building plots. The core idea is that any plot can be uniquely described as a combination of

- a dataset
- a geom
- a set of mappings
- a statistic
- a position adjustment
- a coordinate system
- a faceting scheme

Coordinate System

```
ggplot()
```

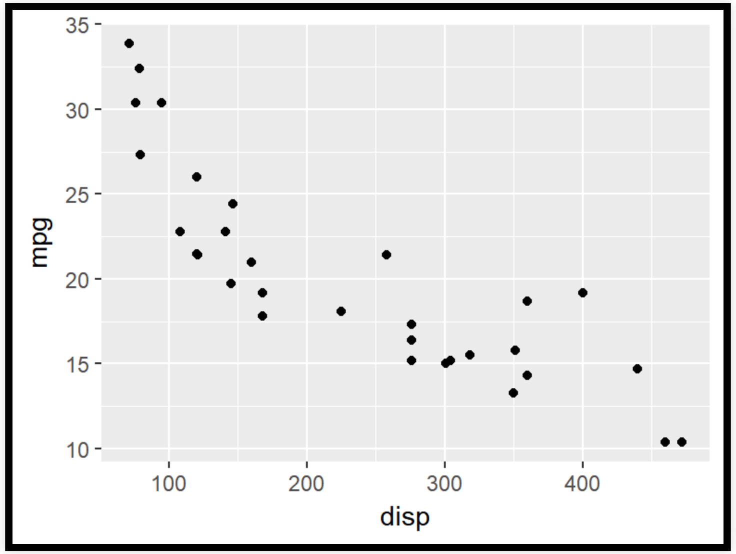


Data Set

```
ggplot(data = mtcarz)
```

Geoms

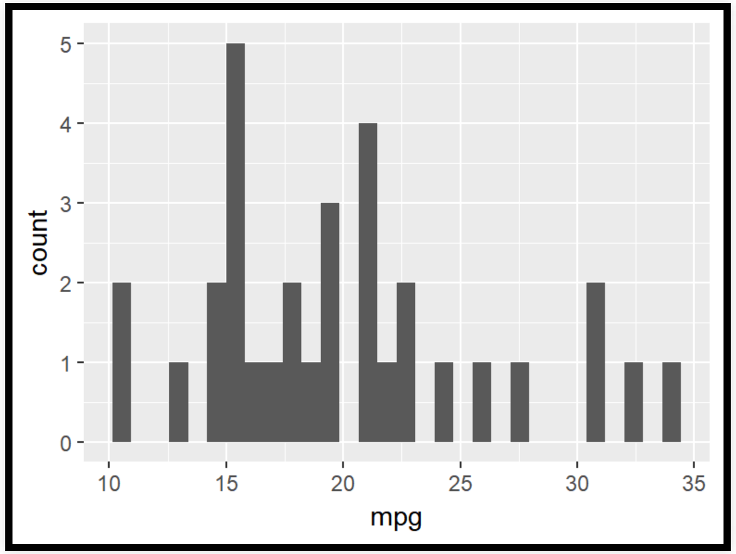
```
ggplot(data = mtcarz) +
  geom_point(mapping = aes(x = disp, y = mpg))
```



Geoms

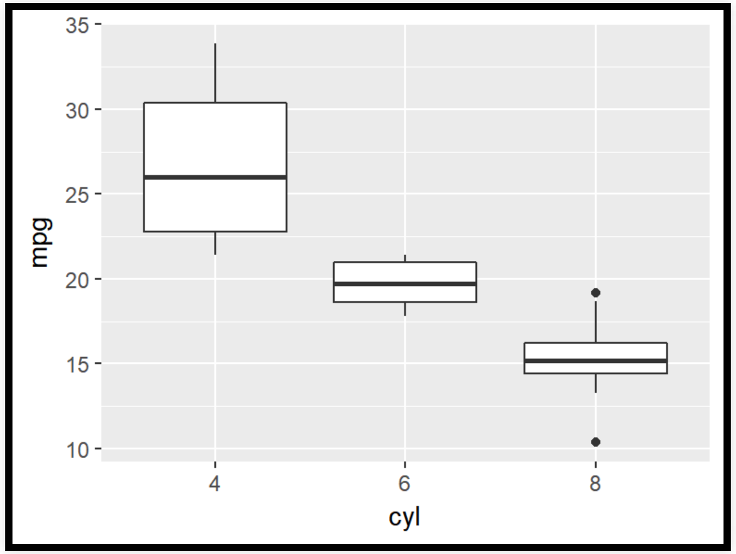
```
ggplot(data = mtcarz) +
  geom_histogram(mapping = aes(x = mpg))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



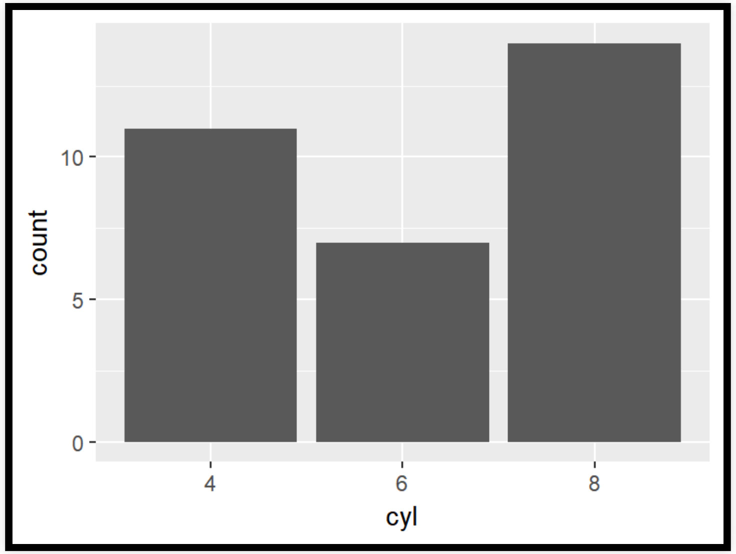
Geoms

```r
ggplot(data = mtcarz) +
  geom_boxplot(mapping = aes(x = cyl, y = mpg))
```



Geoms

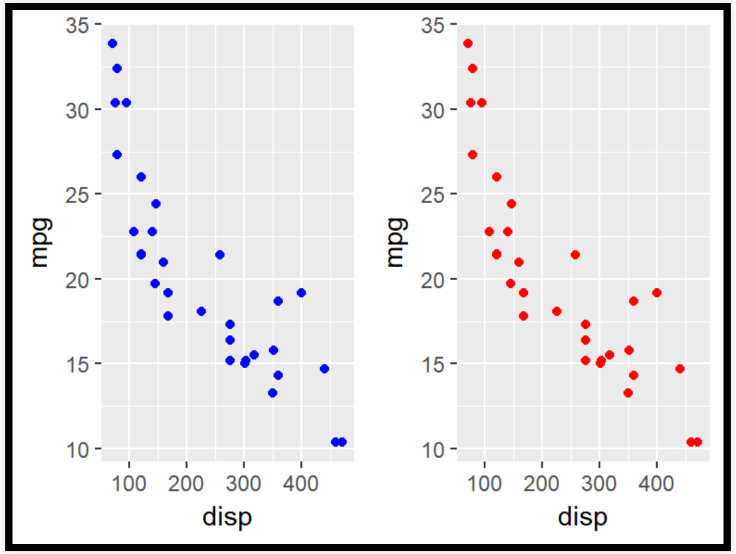
```
ggplot(data = mtcarz) +
  geom_bar(mapping = aes(x = cyl))
```



Aesthetics

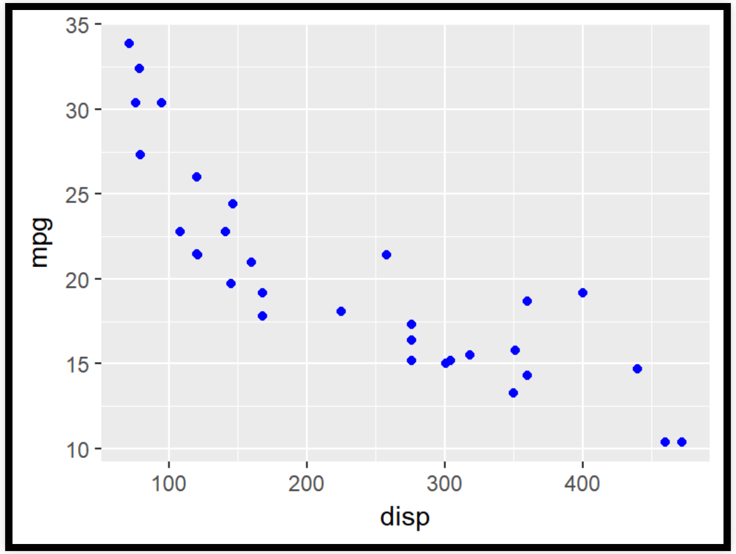
- color
- shape
- size
- background color
- transparency
- line type

Color

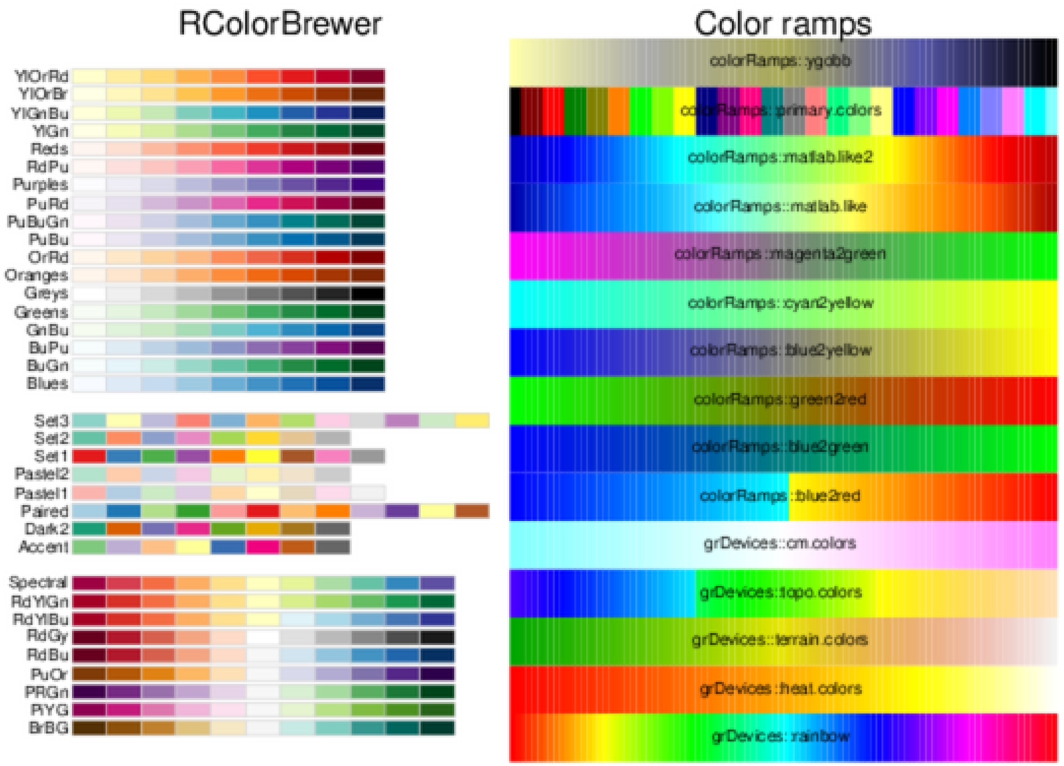

35
30
25
20
15
10
mpg
100
200
300
400
disp
35
30
25
20
15
10
mpg
100
200
300
400
disp

Color Hex Code

```
ggplot(data = mtcarz) +
  geom_point(mapping = aes(x = disp, y = mpg), color = "#0000ff")
```

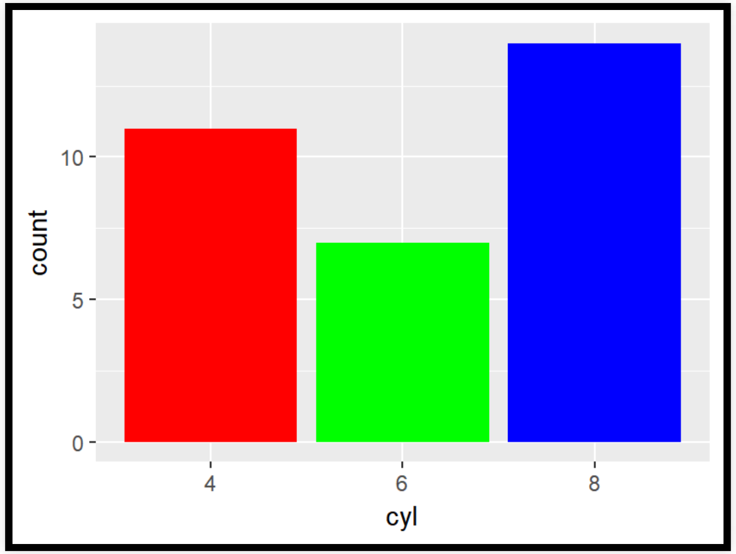


Color Palette

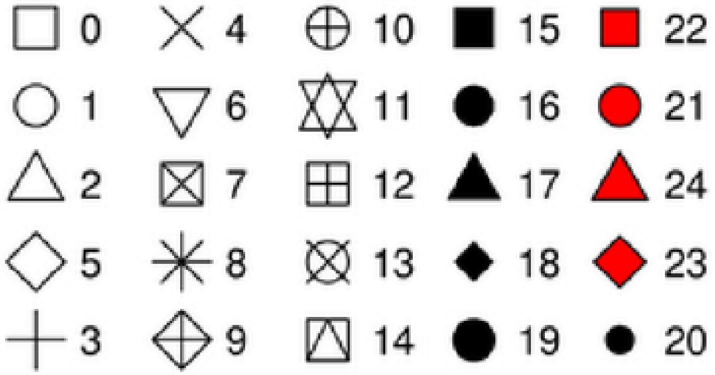


Color Palette

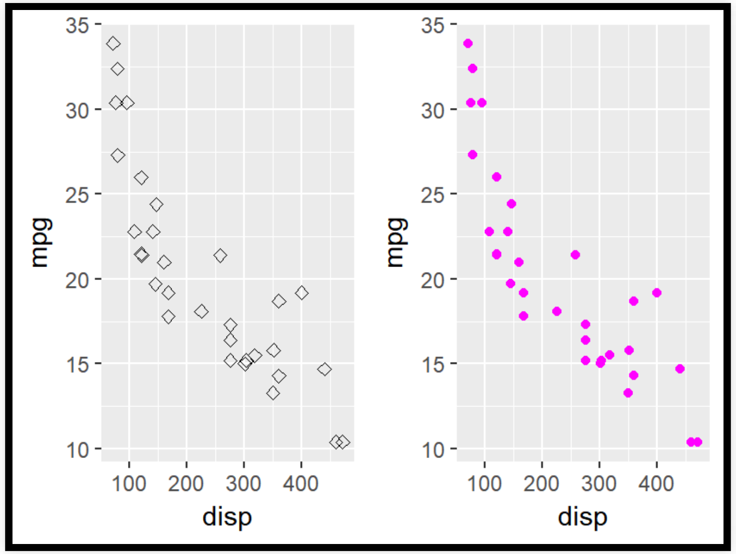
```
ggplot(mtcarz) +
  geom_bar(aes(x = cyl), fill = rainbow(3))
```



Shape

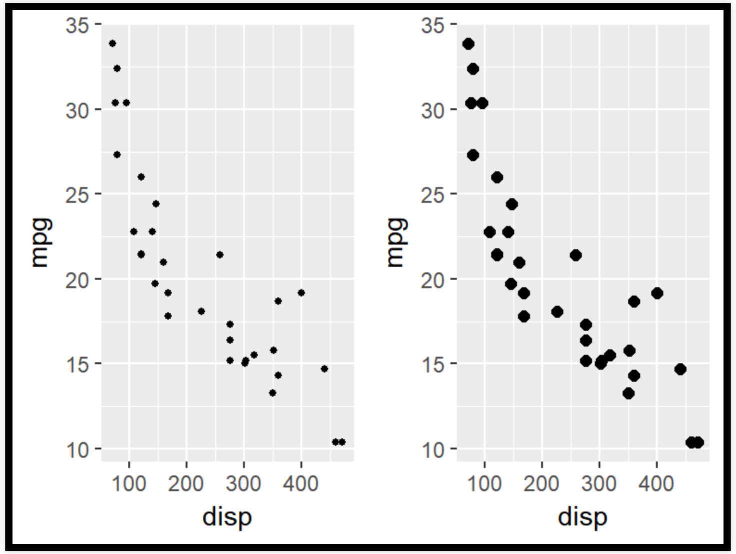


Shape

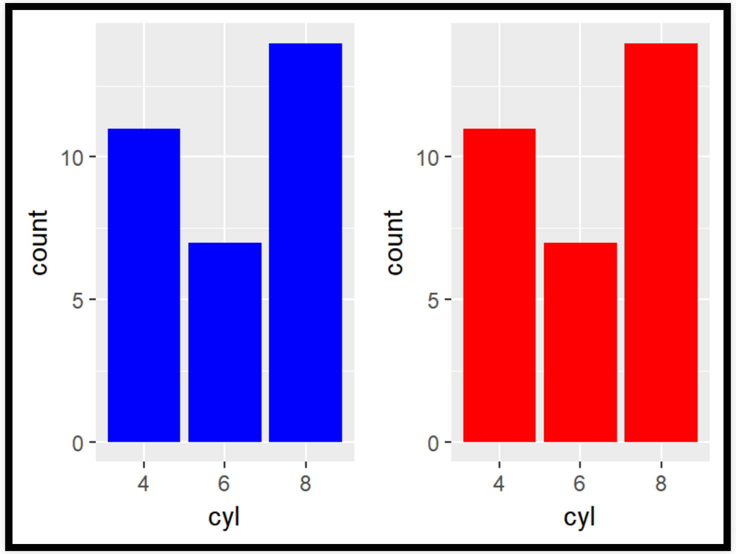


35
30
25
20
15
10
mpg
100
200
300
400
disp
35
30
25
20
15
10
mpg
100
200
300
400
disp

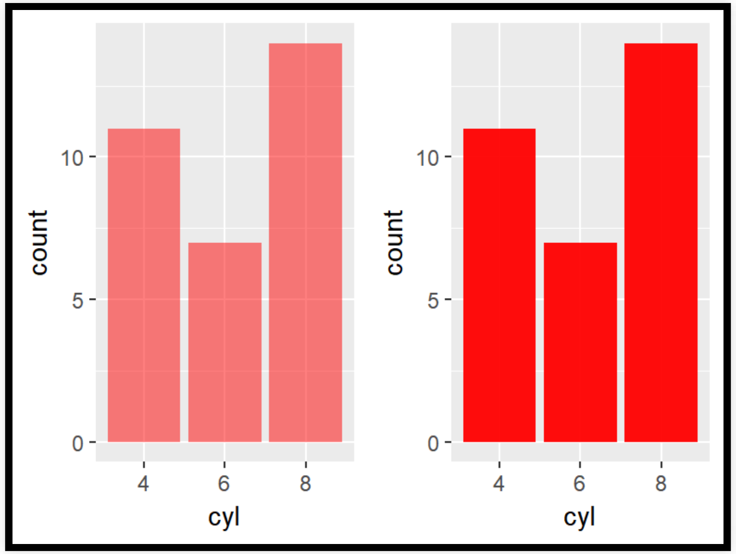
Size


35
30
25
20
15
10
mpg
100
200
300
400
disp
35
30
25
20
15
10
mpg
100
200
300
400
disp

Background Color

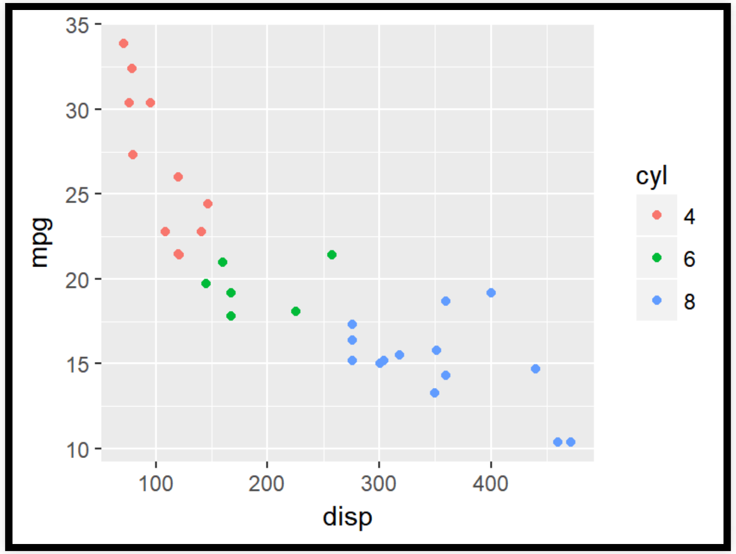


Transparency



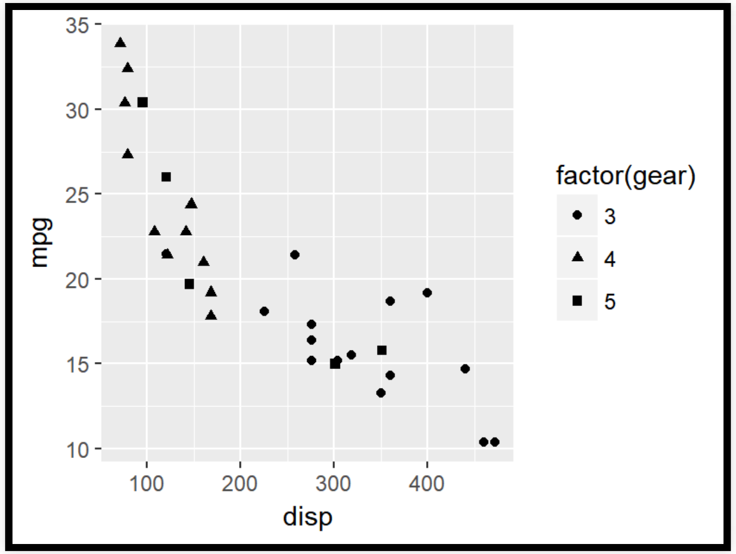
Map Color

```
ggplot(data = mtcarz) +
  geom_point(aes(x = disp, y = mpg, color = cyl))
```



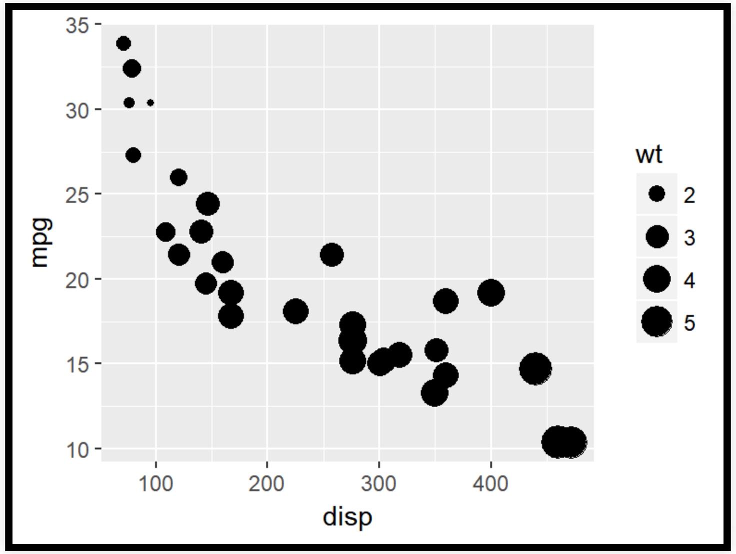
Map Shape

```
ggplot(data = mtcarz) +
  geom_point(aes(x = disp, y = mpg, shape = factor(gear)))
```



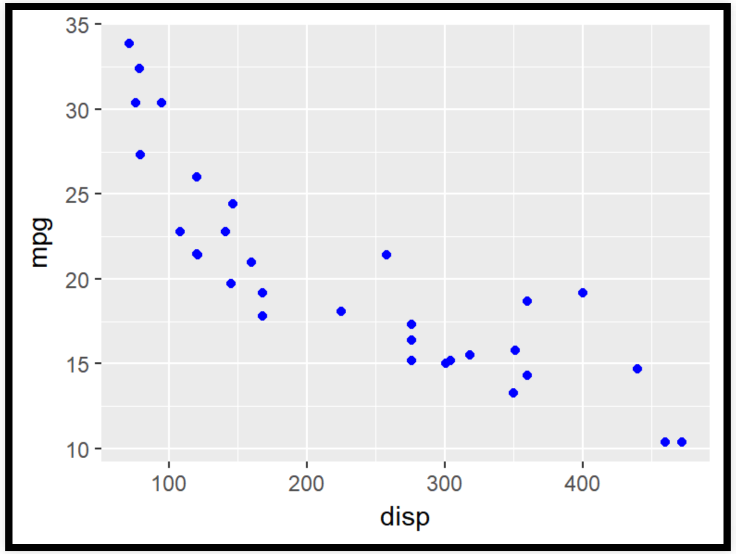
Map Size

```
ggplot(data = mtcarz) +
  geom_point(aes(x = disp, y = mpg, size = wt))
```



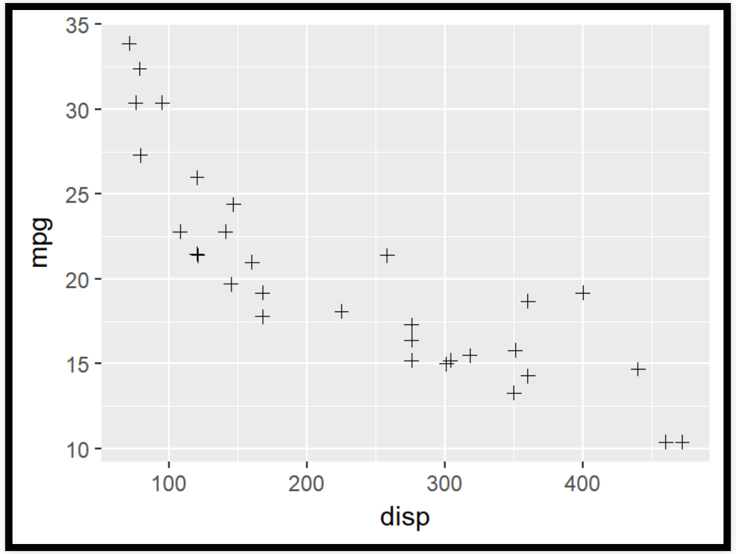
Color = blue

```
ggplot(data = mtcarz) +
  geom_point(aes(x = disp, y = mpg), color = "blue")
```



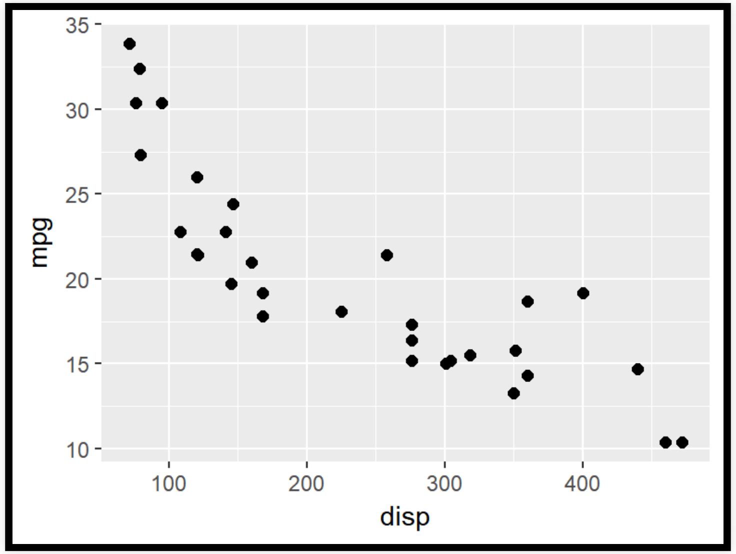
Shape = 3

```
ggplot(data = mtcarz) +
  geom_point(aes(x = disp, y = mpg), shape = 3)
```

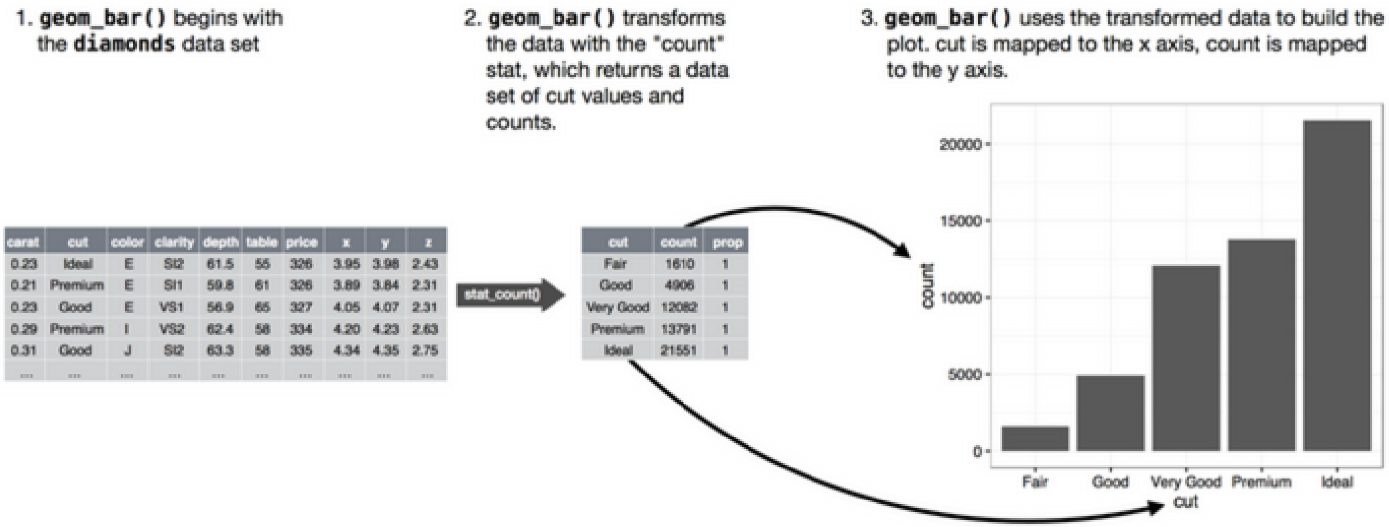


Size = 2

```
ggplot(data = mtcarz) +
  geom_point(aes(x = disp, y = mpg), size = 2)
```



Transform



1. **geom_bar()** begins with the **diamonds** data set

2. **geom_bar()** transforms the data with the "count" stat, which returns a data set of cut values and counts.

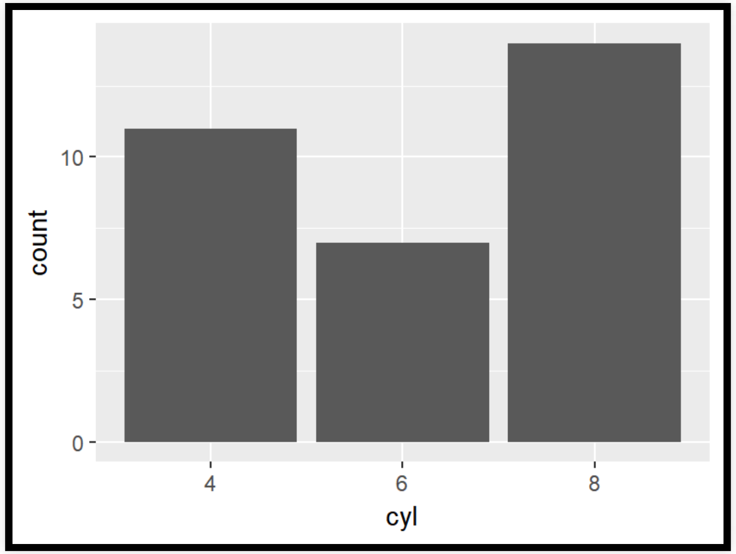
3. **geom_bar()** uses the transformed data to build the plot. cut is mapped to the x axis, count is mapped to the y axis.

carat	cut	color	clarity	depth	table	price	x	y	z
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31
0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31
0.29	Premium	I	VS2	62.4	58	334	4.20	4.23	2.63
0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75
...	...	...	...	...	...	...	...	...	...

cut	count	prop
Fair	1610	1
Good	4906	1
Very Good	12082	1
Premium	13791	1
Ideal	21551	1

Bar chart

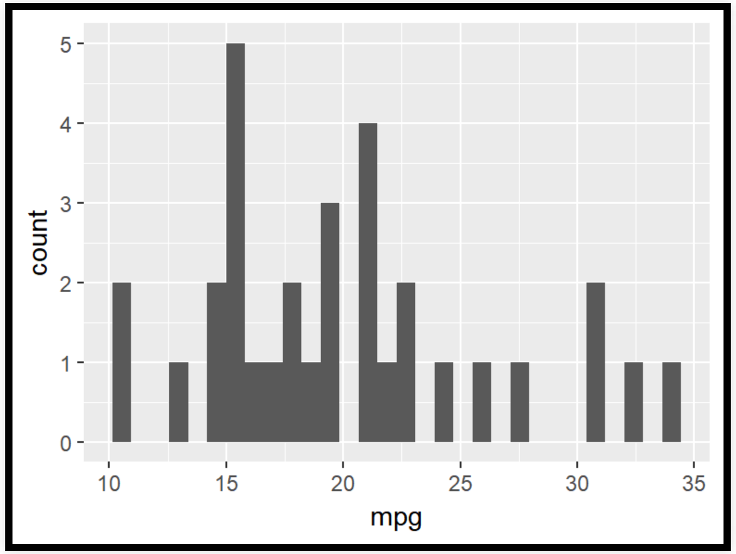
```
ggplot(data = mtcarz) +
  stat_count(mapping = aes(x = cyl))
```



Histograms

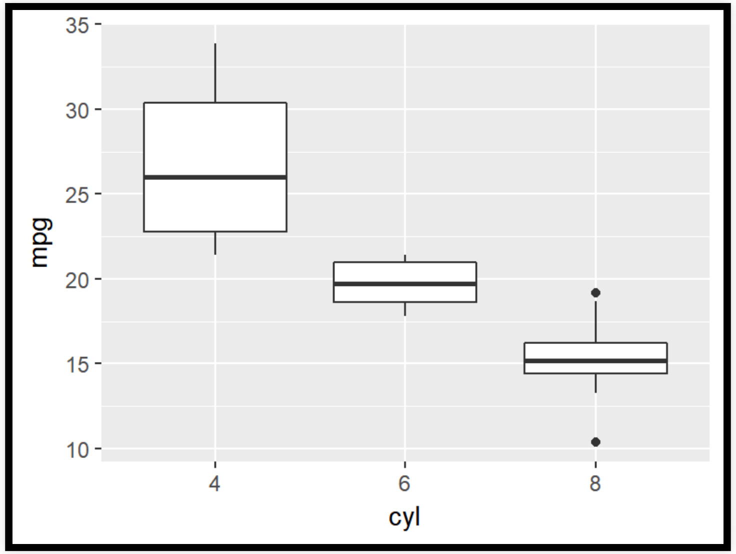
```
ggplot(data = mtcarz) +
  stat_bin(mapping = aes(x = mpg))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



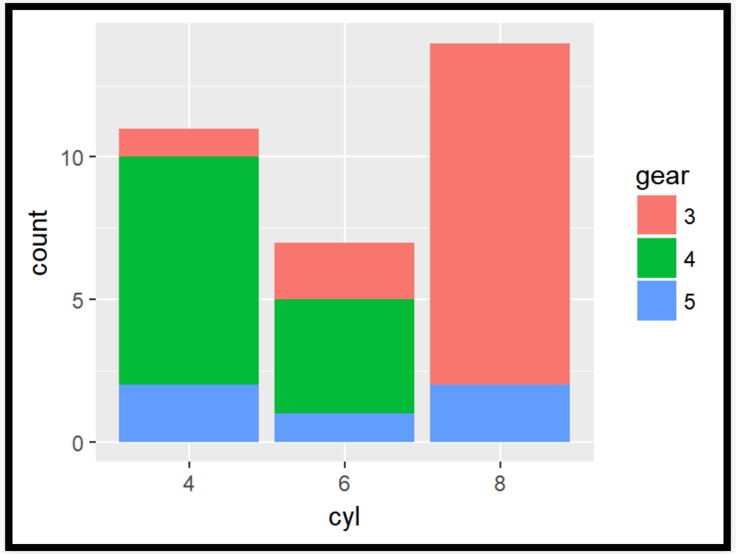
Box plots

```r
ggplot(data = mtcarz) +
  stat_boxplot(mapping = aes(x = cyl, y = mpg))
```



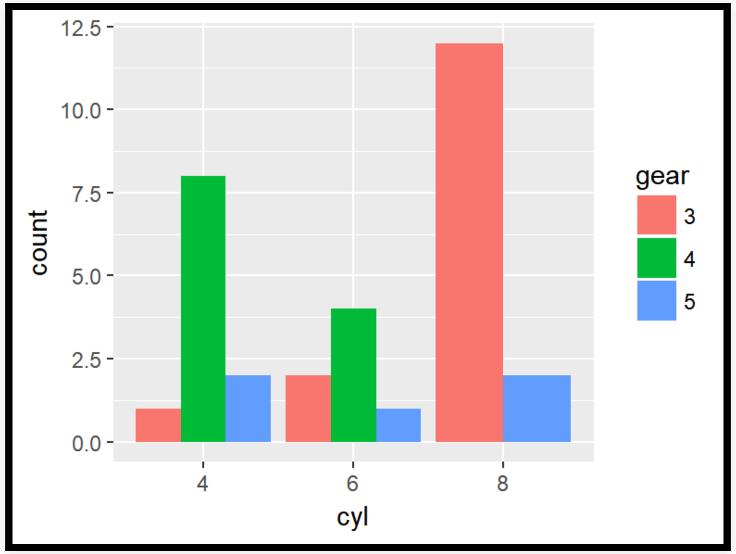
Position

```r
ggplot(data = mtcarz) +
  geom_bar(mapping = aes(x = cyl, fill = gear))
```



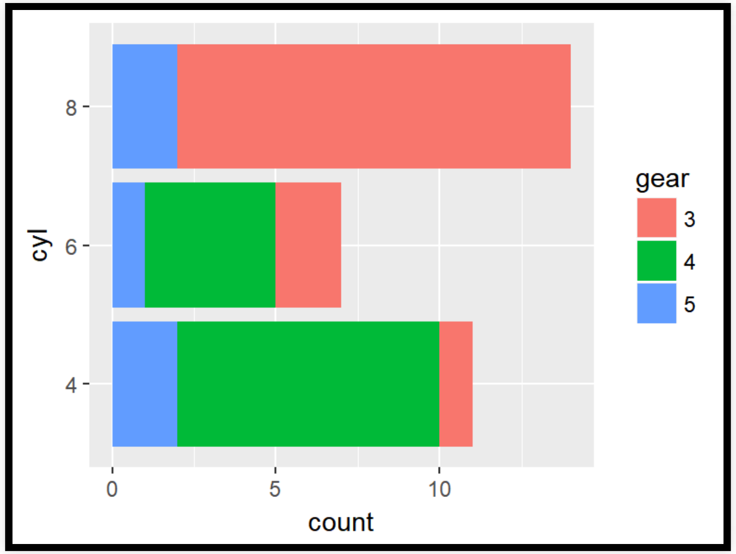
Position

```
ggplot(data = mtcarz) +
  geom_bar(mapping = aes(x = cyl, fill = gear), position = "dodge")
```



Flip Coordinates

```r
ggplot(data = mtcarz) +
  geom_bar(mapping = aes(x = cyl, fill = gear)) +
  coord_flip()
```

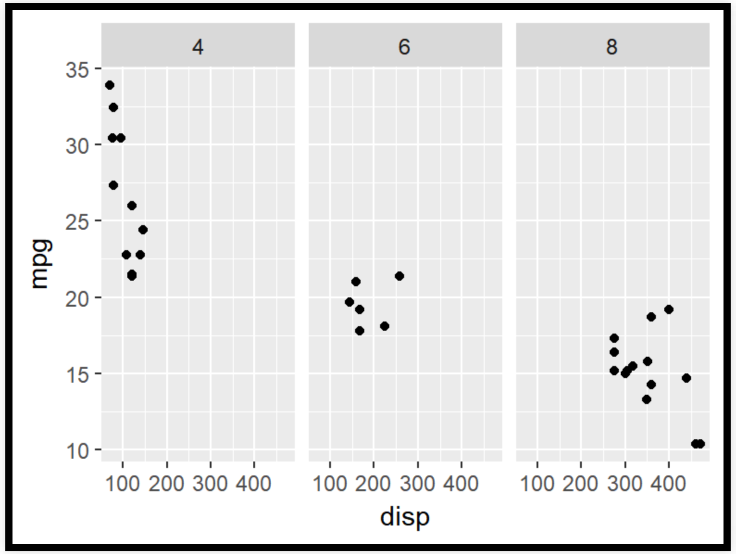


Facets

- split the plot into sub plots
- each subplot displays a subset of the data
- use `facet_wrap()` to create sub plots based on a single variable
- use `facet_grid()` to create sub plots based on a two variable

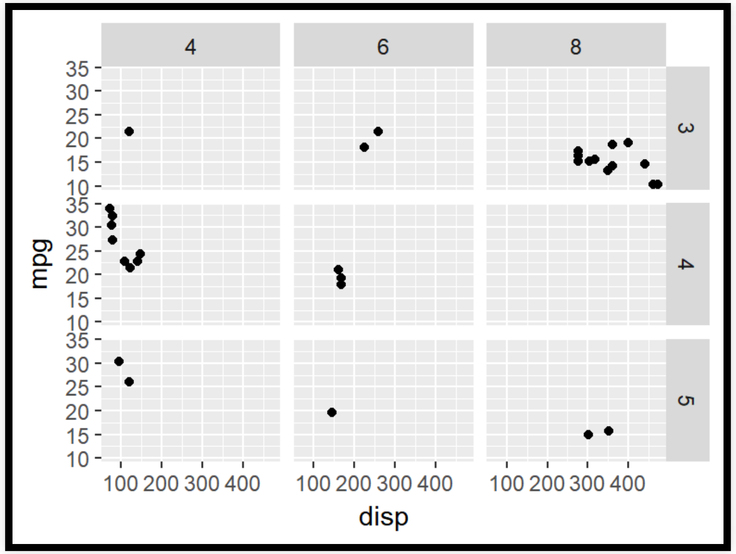
Facets

```
ggplot(data = mtcarz) +
  geom_point(mapping = aes(x = disp, y = mpg)) +
  facet_wrap(~ cyl)
```



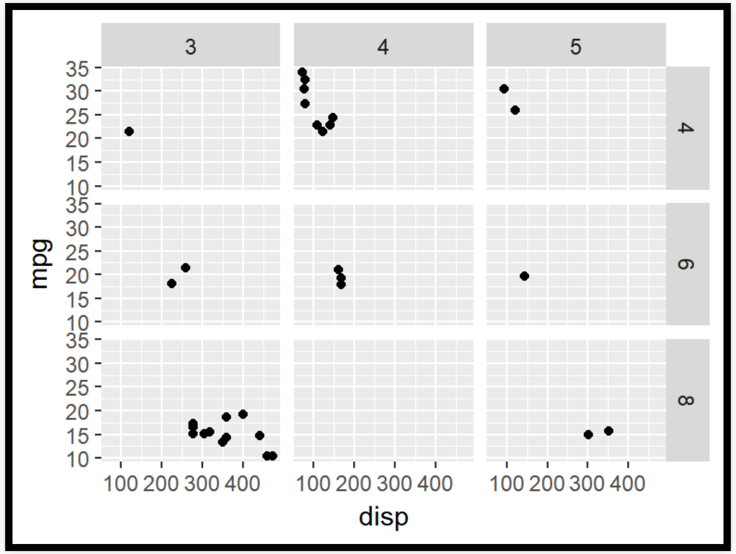
Facets

```
ggplot(data = mtcarz) +
  geom_point(mapping = aes(x = disp, y = mpg)) +
  facet_grid(gear ~ cyl)
```



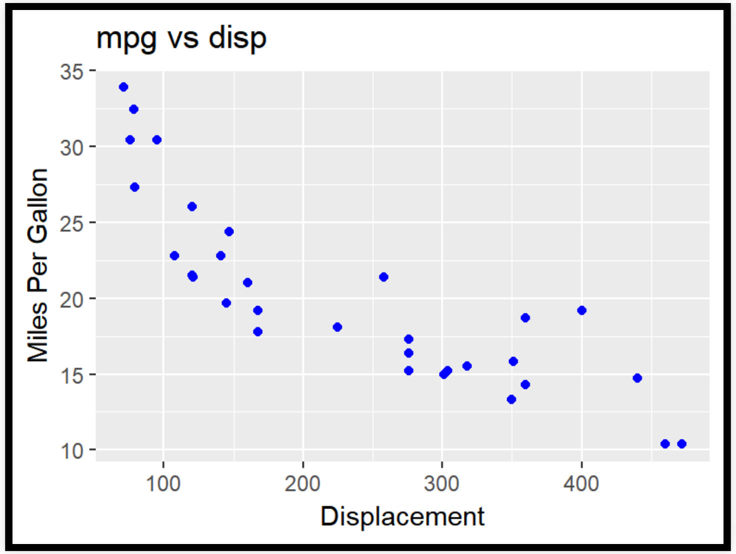
Facets

```
ggplot(data = mtcarz) +
  geom_point(mapping = aes(x = disp, y = mpg)) +
  facet_grid(cyl ~ gear)
```



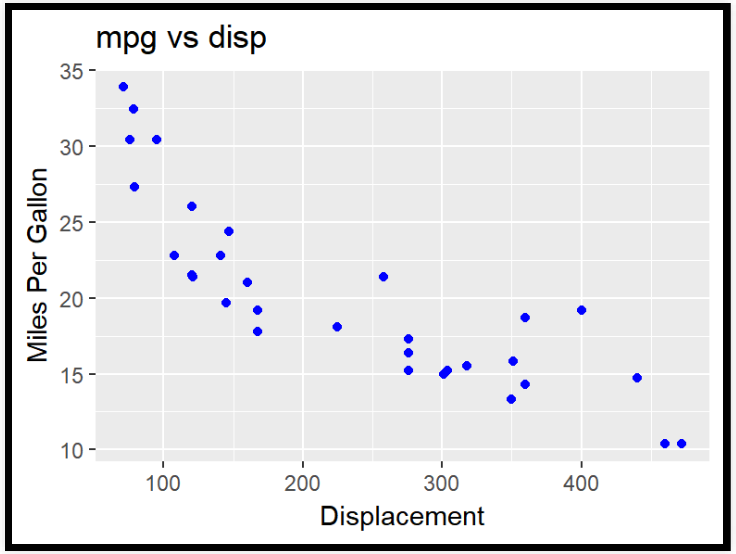
Title, Axis Labels & Limits

```
ggplot(data = mtcarz) +
  geom_point(aes(x = disp, y = mpg), color = "blue") +
  labs(title = "mpg vs disp", x = "Displacement",
       y = "Miles Per Gallon")
```



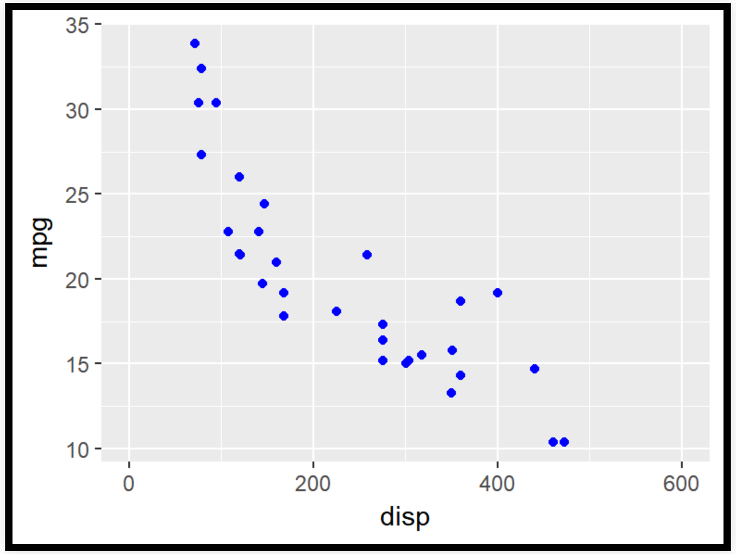
Title

```
ggplot(data = mtcarz) +
  geom_point(aes(x = disp, y = mpg), color = "blue") +
  ggtitle("mpg vs disp") + xlab("Displacement") +
  ylab("Miles Per Gallon")
```



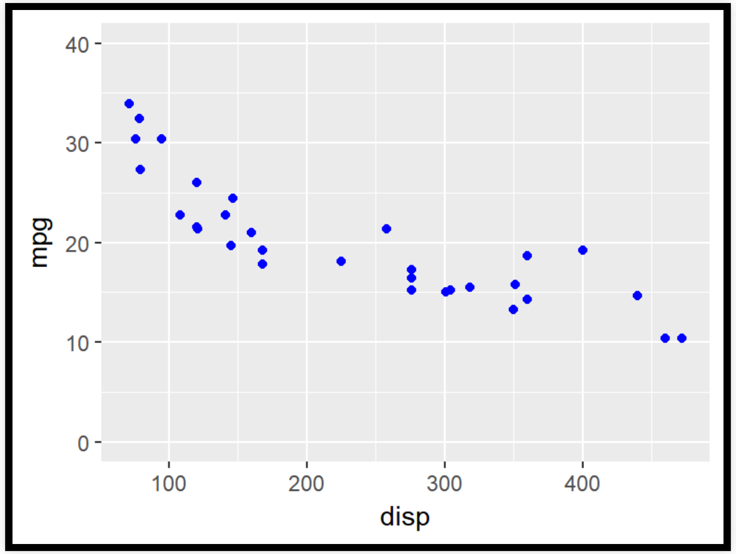
Axis Limits

```
ggplot(data = mtcarz) +
  geom_point(aes(x = disp, y = mpg), color = "blue") +
  xlim(c(0, 600))
```



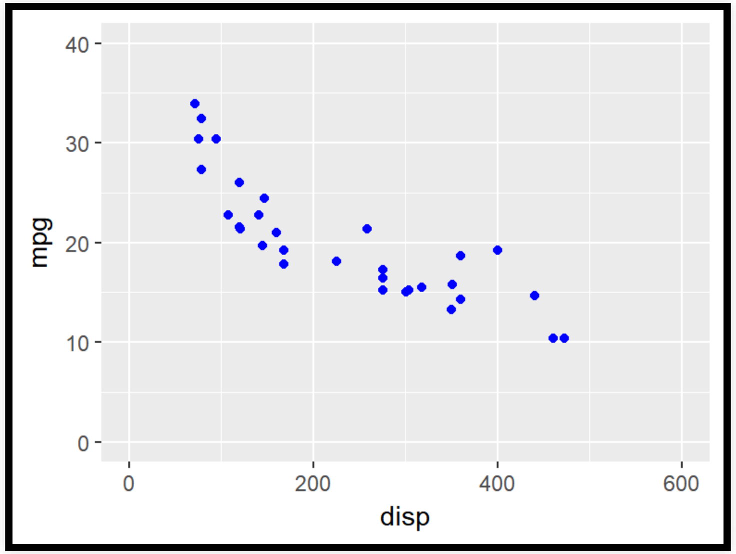
Axis Limits

```r
ggplot(data = mtcarz) +
  geom_point(aes(x = disp, y = mpg), color = "blue") +
  ylim(c(0, 40))
```



Axis Limits

```
ggplot(data = mtcarz) +
  geom_point(aes(x = disp, y = mpg), color = "blue") +
  expand_limits(x = c(0, 600), y = c(0, 40))
```

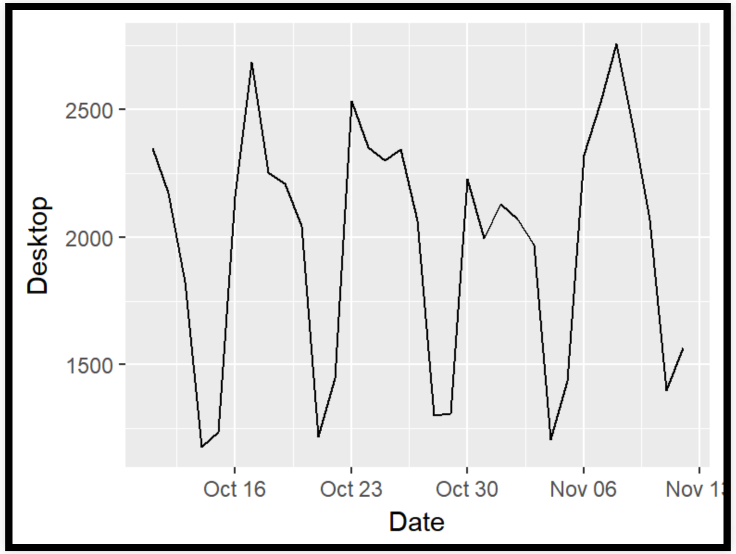


Line Chart: Data

```r
device <- read_csv("data/users_device.csv",
                   col_types = list(col_date(format = "%m/%d/%y"),
                                    col_integer(), col_integer(),
                                    col_integer()))
```

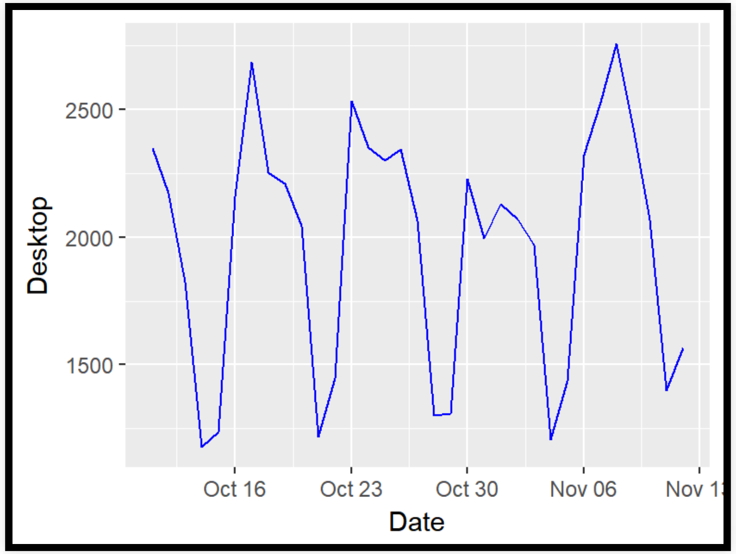
Line Chart

```
ggplot(data = device) +
  geom_line(mapping = aes(x = Date, y = Desktop))
```



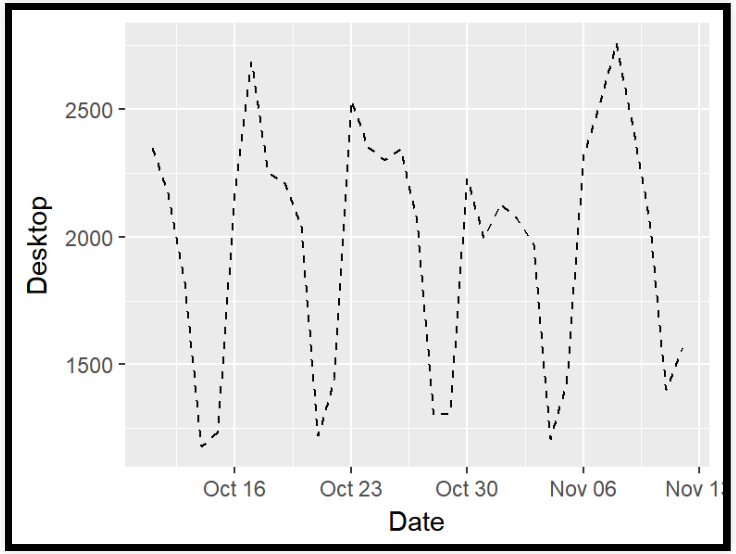
Line Color

```
ggplot(data = device) +
  geom_line(mapping = aes(x = Date, y = Desktop), color = "blue")
```



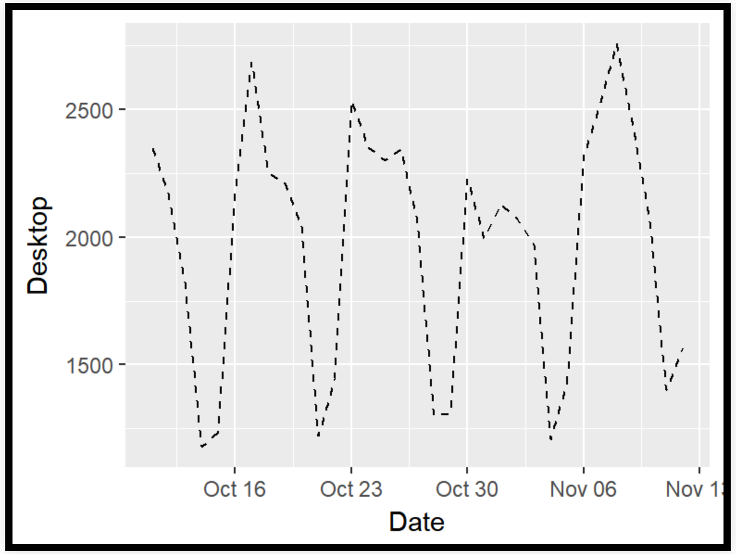
Line Type

```
ggplot(data = device) +
  geom_line(mapping = aes(x = Date, y = Desktop), linetype = 2)
```



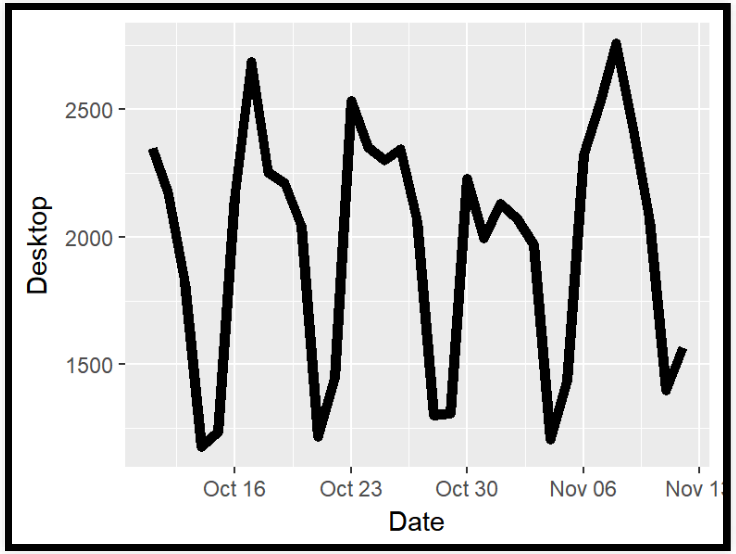
Line Type

```r
ggplot(data = device) +
  geom_line(mapping = aes(x = Date, y = Desktop), linetype = "dashed")
```



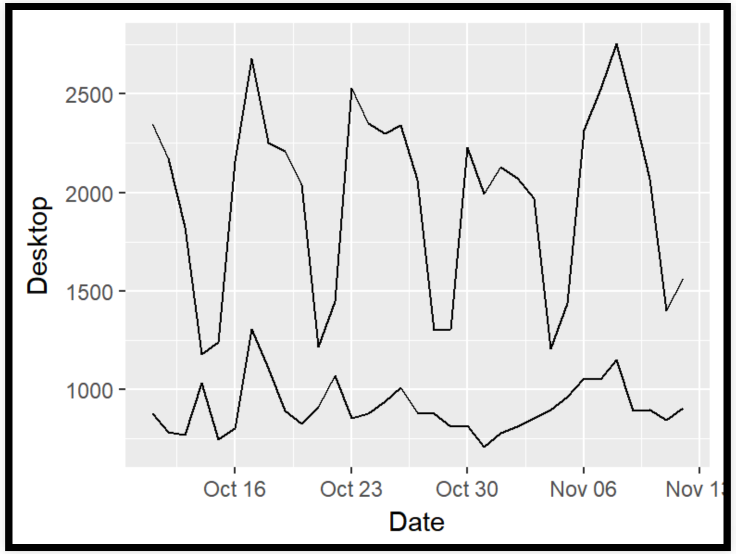
Line Width

```
ggplot(data = device) +
  geom_line(mapping = aes(x = Date, y = Desktop), size = 2)
```



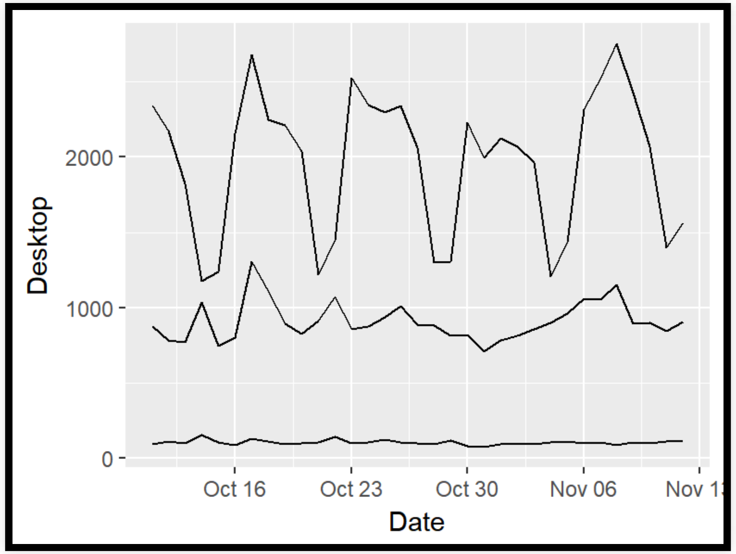
Multiple Lines

```
ggplot(data = device) +
  geom_line(mapping = aes(x = Date, y = Desktop)) +
  geom_line(mapping = aes(x = Date, y = Mobile))
```



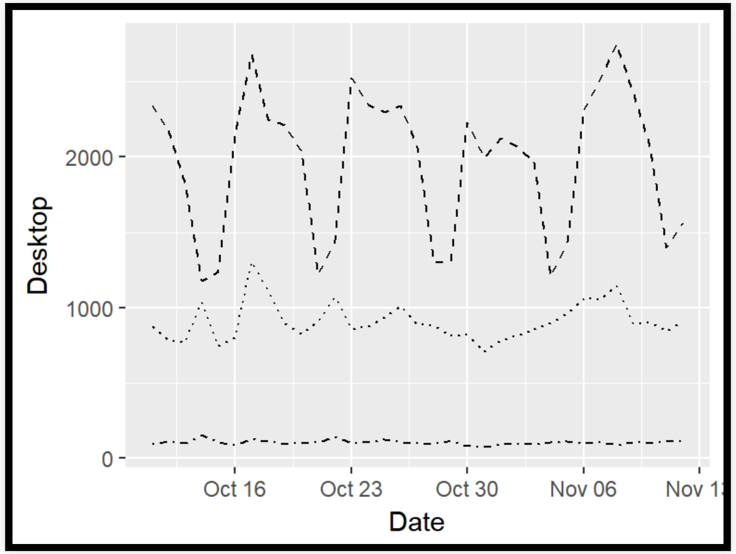
Multiple Lines

```
ggplot(data = device) +
  geom_line(mapping = aes(x = Date, y = Desktop)) +
  geom_line(mapping = aes(x = Date, y = Mobile)) +
  geom_line(mapping = aes(x = Date, y = Tablet))
```



Multiple Lines

```
ggplot(data = device) +
  geom_line(mapping = aes(x = Date, y = Desktop), linetype = 2) +
  geom_line(mapping = aes(x = Date, y = Mobile), linetype = 3) +
  geom_line(mapping = aes(x = Date, y = Tablet), linetype = 4)
```

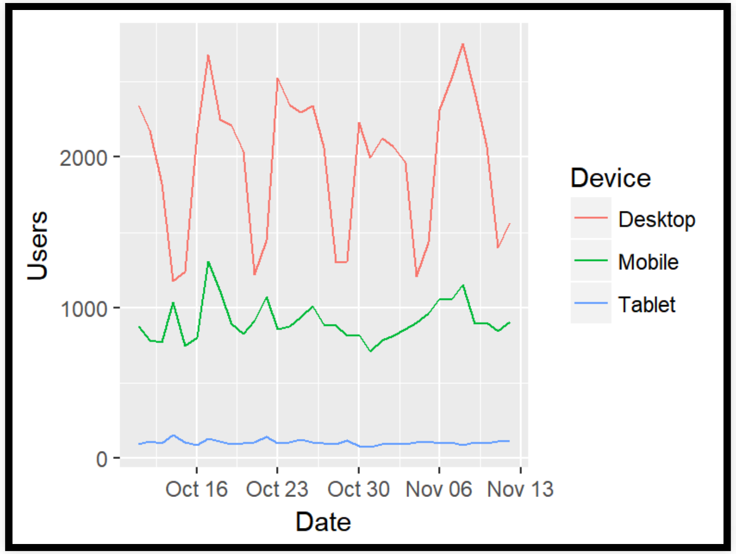


Line Chart: Data

```
tidy_device <- read_csv("data/tidy_users.csv",
                        col_types = list(col_date(format = "%m/%d/%y"),
                                         col_character(), col_integer())
```

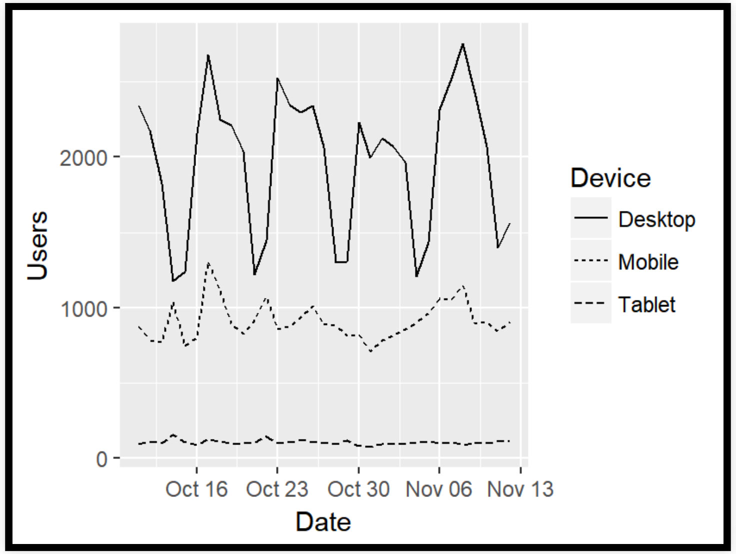
Multiple Lines

```
ggplot(data = tidy_device) +
  geom_line(mapping = aes(x = Date, y = Users, group = Device,
                          color = Device))
```



Multiple Lines

```
ggplot(data = tidy_device) +
  geom_line(mapping = aes(x = Date, y = Users, group = Device,
                          linetype = Device))
```



Thank You

For more information please visit our website
www.rsquaredacademy.com