ggplot2: An Introduction

Arun Rangarajan

2016-11-30

Contents

1 Background

- R has different plotting systems:
  - base
  - grid
  - lattice
  - ggplot2
- ggplot2 - based on the good parts of other systems
- gg = Grammar of Graphics
- Based on Leland Wilkinson's "The Grammar of Graphics"
- Developed in R by Hadley Wickham

2 What is special about ggplot2?

- Deep underlying grammar (just like language grammar)
- Based on a simple set of core principles
- Good defaults
- Lots of options for customization
- Takes care of a lot of details (like legends, axes grids)
- Plots are made iteratively
- Easy to learn

3 Parts of a ggplot

- Data
- Aesthetics (aes)
  - Color
  - Shape
  - Size
- Geometric Objects (geoms)
  - Points
  - Lines
  - Polygons
- Statistical Transformations (stats)
  - Histograms
  - Boxplots
- Legends and Axes
- Coordinate system (coord)

  - Cartesian
  - Polar
- Faceting (aka latticing/trellising)
- Theme
  - Background color
  - Font size

4 Installation and Loading

- ggplot2 is not shipped with base R, so:

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

- Load the library before use:

```
library(ggplot2)
```

5 Data Set to Explore

- `ggplot2` requires the data be provided in a data frame
- Example data frame to explore: `mtcars`

```
dim(mtcars)
```

```
## [1] 32 11
```

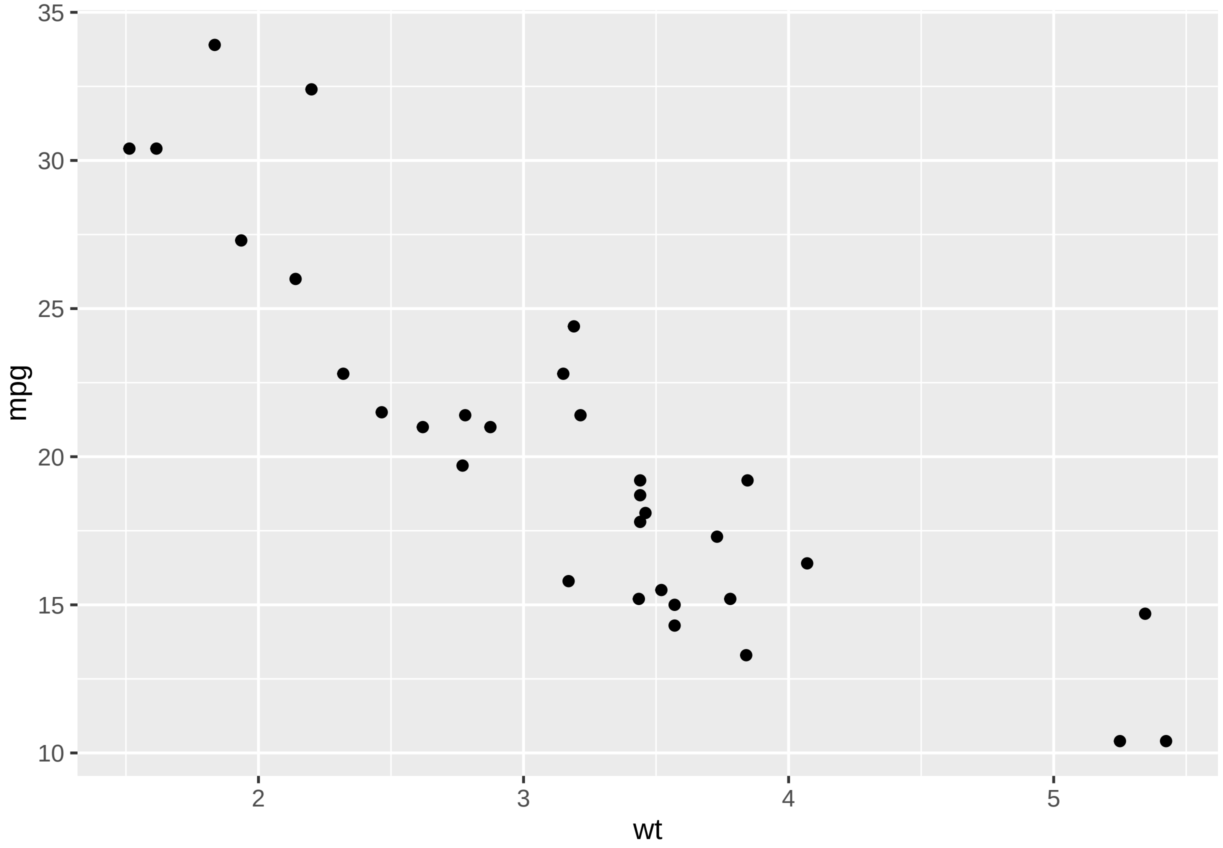
```
head(mtcars)
```

```
##                    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
```

6 Our first ggplot

- ggplot2 uses the `+` symbol to chain functions

```
library(ggplot2)
ggplot(data=mtcars, aes(x=wt, y=mpg)) +
  geom_point()
```



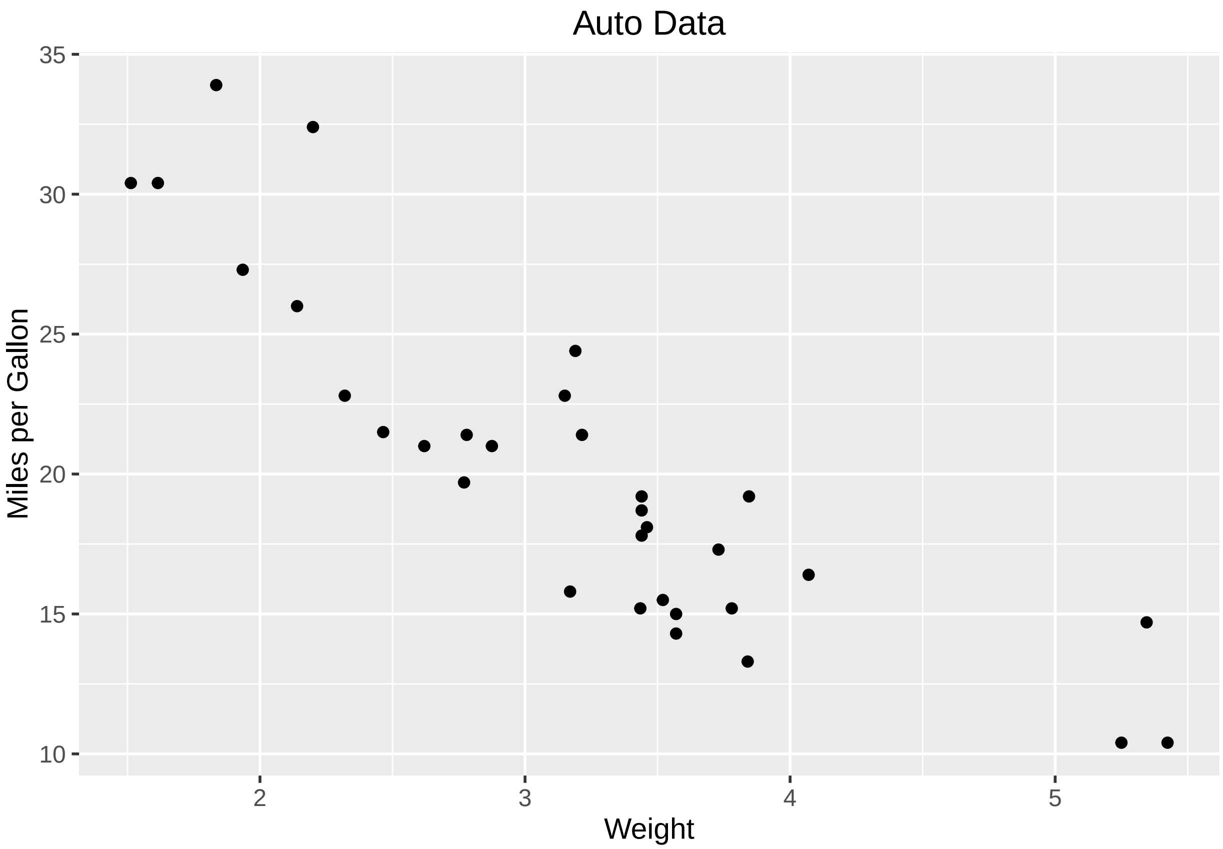
7 Parts of our first ggplot

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

- `ggplot()` function initializes the plot
- Data frame specified in `data` param
- Variables specified in `aes` (aesthetics) param
  - x-axis is `wt`
  - y-axis is `mpg`
- `geom` specifies the visual output (`geom_point()` here produces scatter plot)

8 Adding title and axes labels

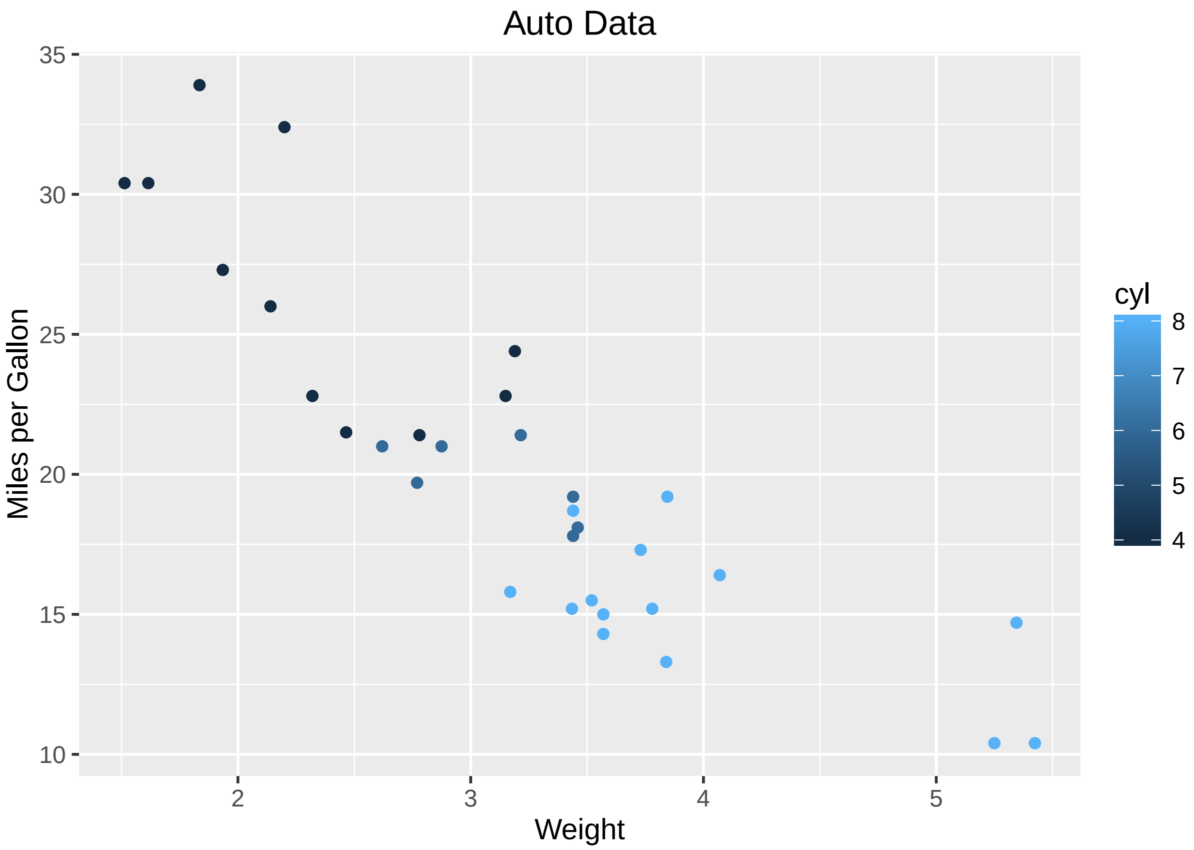
```r
ggplot(data=mtcars, aes(x=wt, y=mpg)) +
  geom_point() +
  labs(title="Auto Data", x="Weight", y="Miles per Gallon")
```



9 Adding colors to points

- Color the points based on cyl column

```
ggplot(data=mtcars, aes(x=wt, y=mpg)) +
  geom_point(aes(color=cyl)) +
  labs(title="Auto Data", x="Weight", y="Miles per Gallon")
```

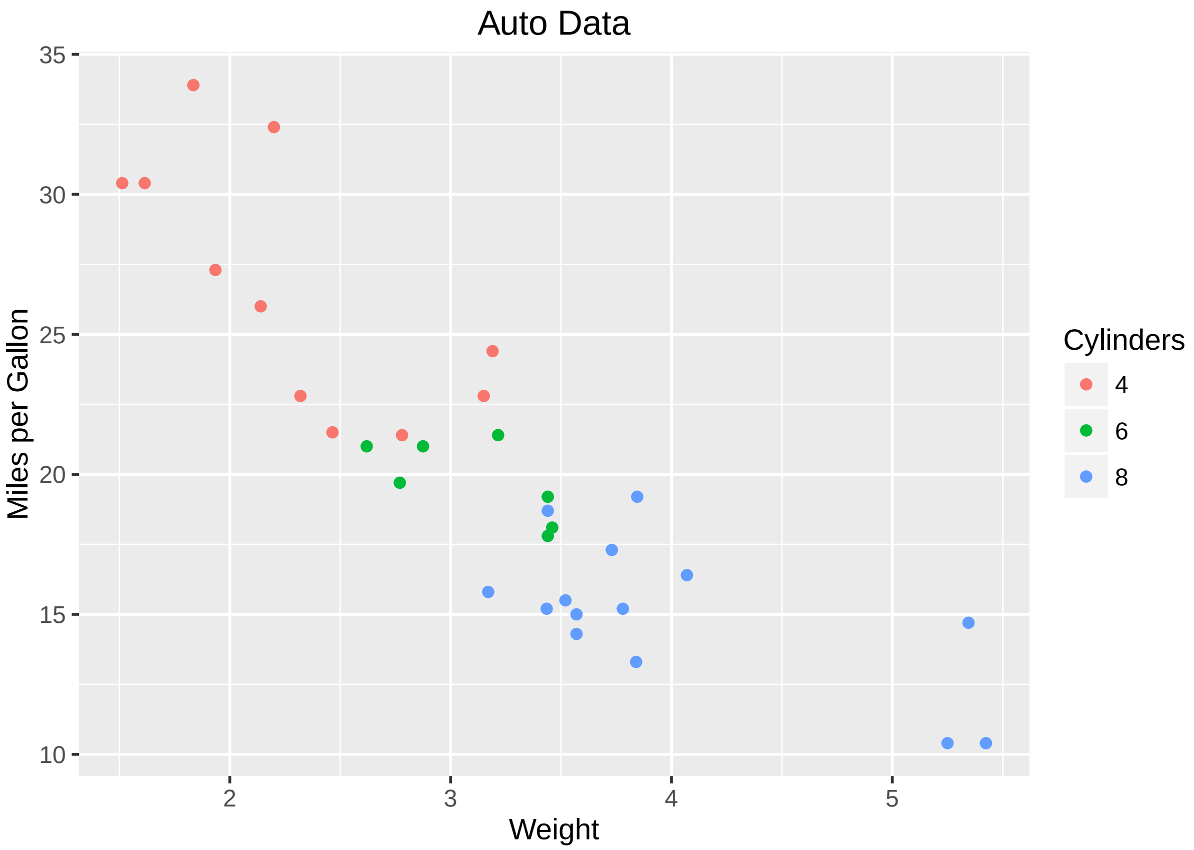


- `cyl` is a numeric field, so continuous colors

10 Adding colors to points (cont...)

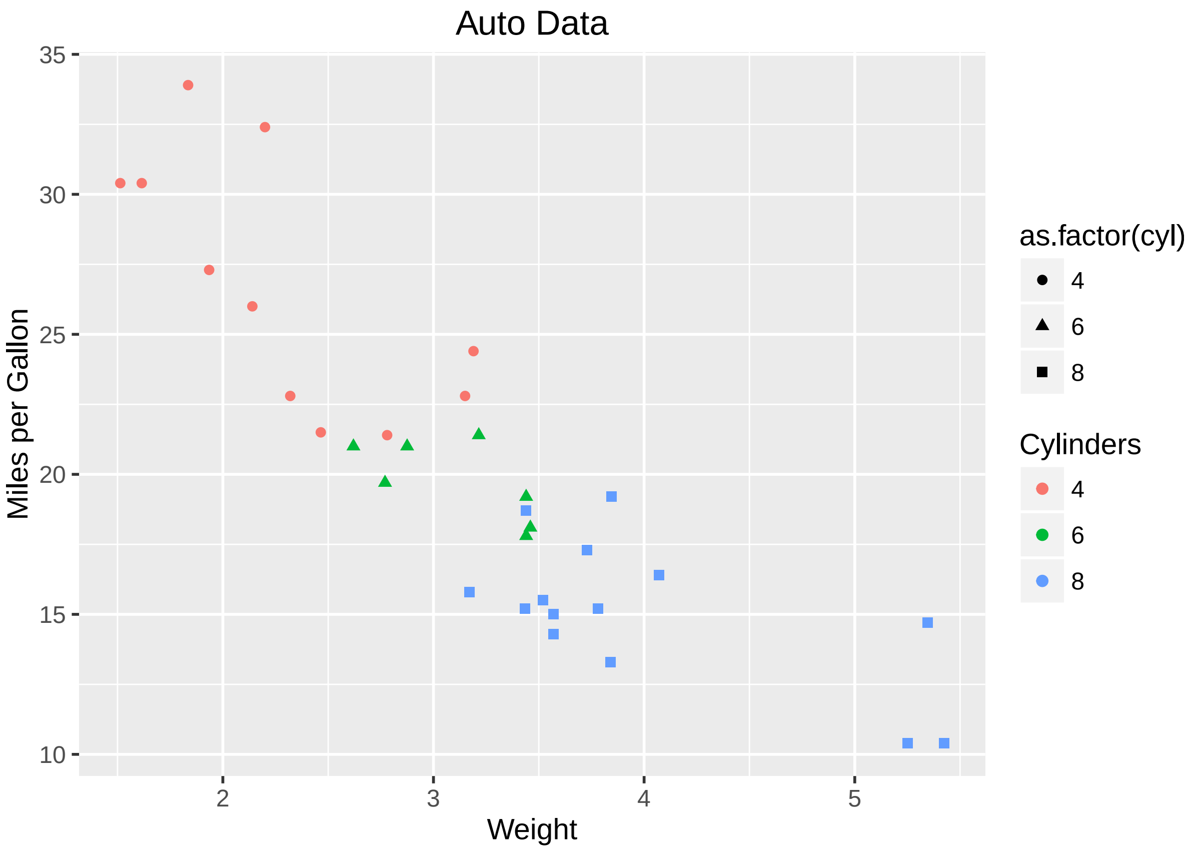
- Convert `cyl` to factor

```
ggplot(data=mtcars, aes(x=wt, y=mpg)) +
  geom_point(aes(color=as.factor(cyl))) +
  labs(title="Auto Data", x="Weight", y="Miles per Gallon",
       color="Cylinders")
```



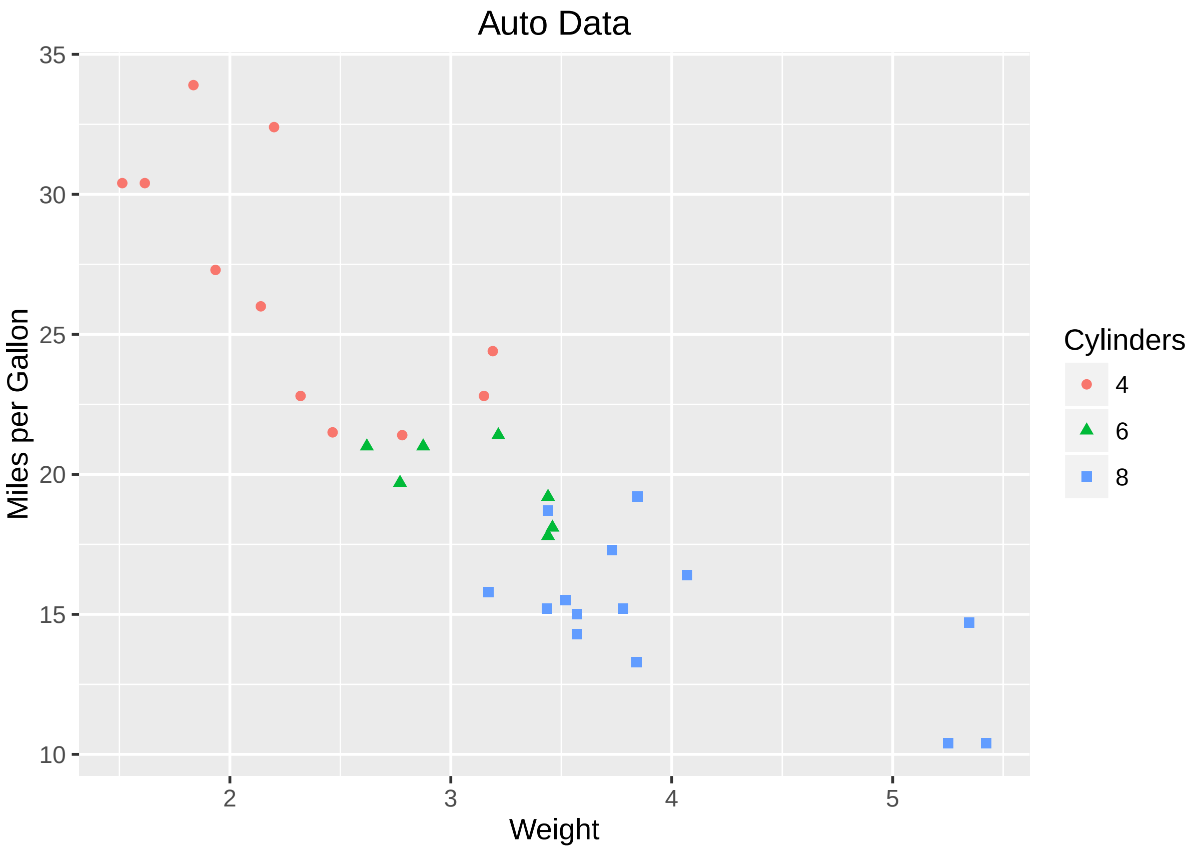
11 Changing the shapes

```
ggplot(data=mtcars, aes(x=wt, y=mpg)) +
  geom_point(aes(color=as.factor(cyl), shape=as.factor(cyl))) +
  labs(title="Auto Data", x="Weight", y="Miles per Gallon",
       color="Cylinders")
```



12 Fix the legend

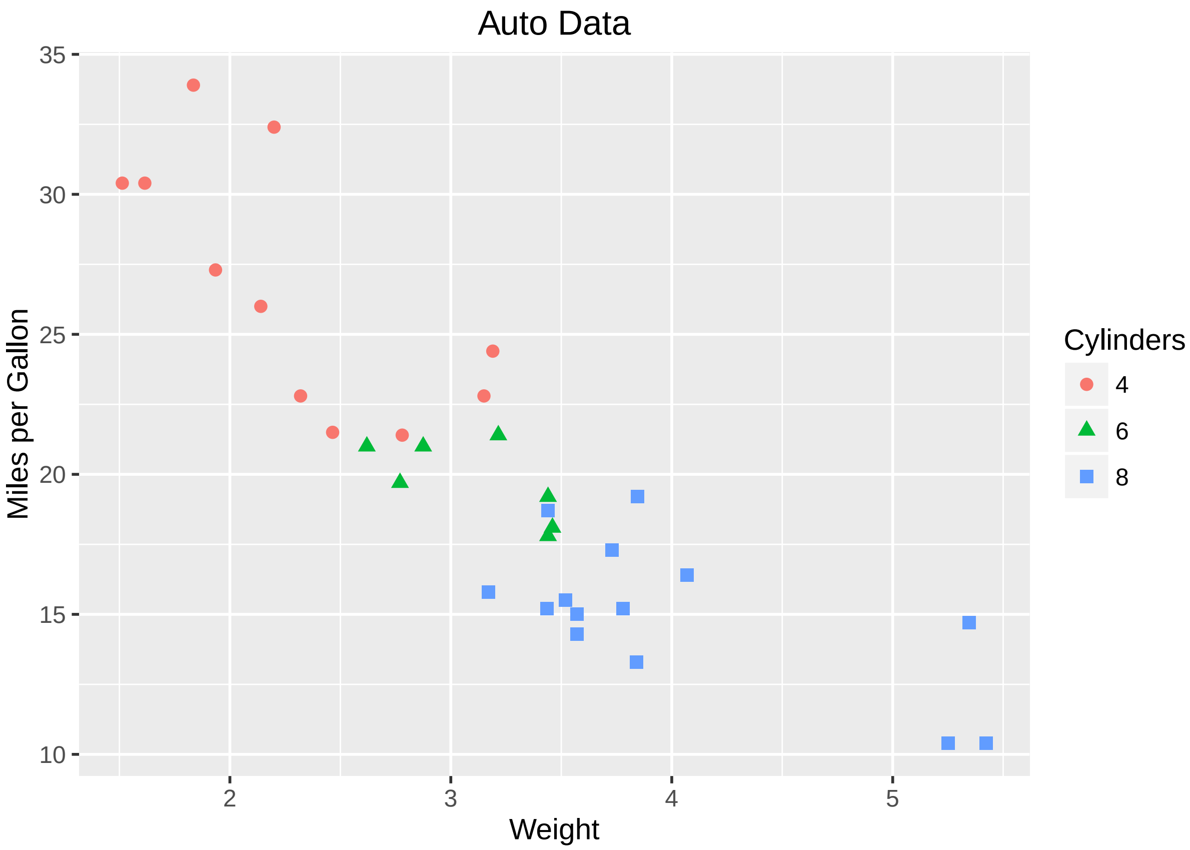
```
ggplot(data=mtcars, aes(x=wt, y=mpg)) +
  geom_point(aes(color=as.factor(cyl), shape=as.factor(cyl))) +
  labs(title="Auto Data", x="Weight", y="Miles per Gallon",
       color="Cylinders", shape="Cylinders")
```



13 Increase the point size

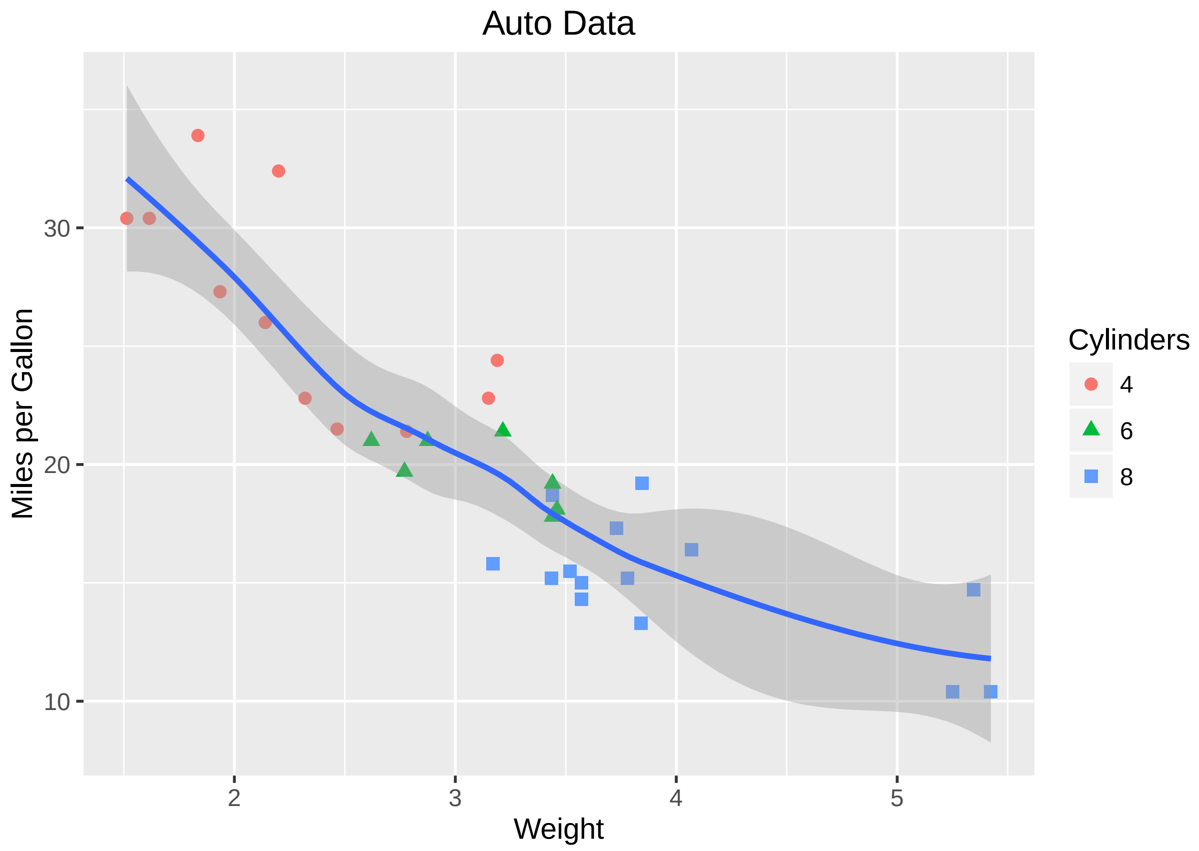
- Do this outside `aes`

```
ggplot(data=mtcars, aes(x=wt, y=mpg)) +
  geom_point(aes(color=as.factor(cyl), shape=as.factor(cyl)), size=2) +
  labs(title="Auto Data", x="Weight", y="Miles per Gallon",
       color="Cylinders", shape="Cylinders")
```



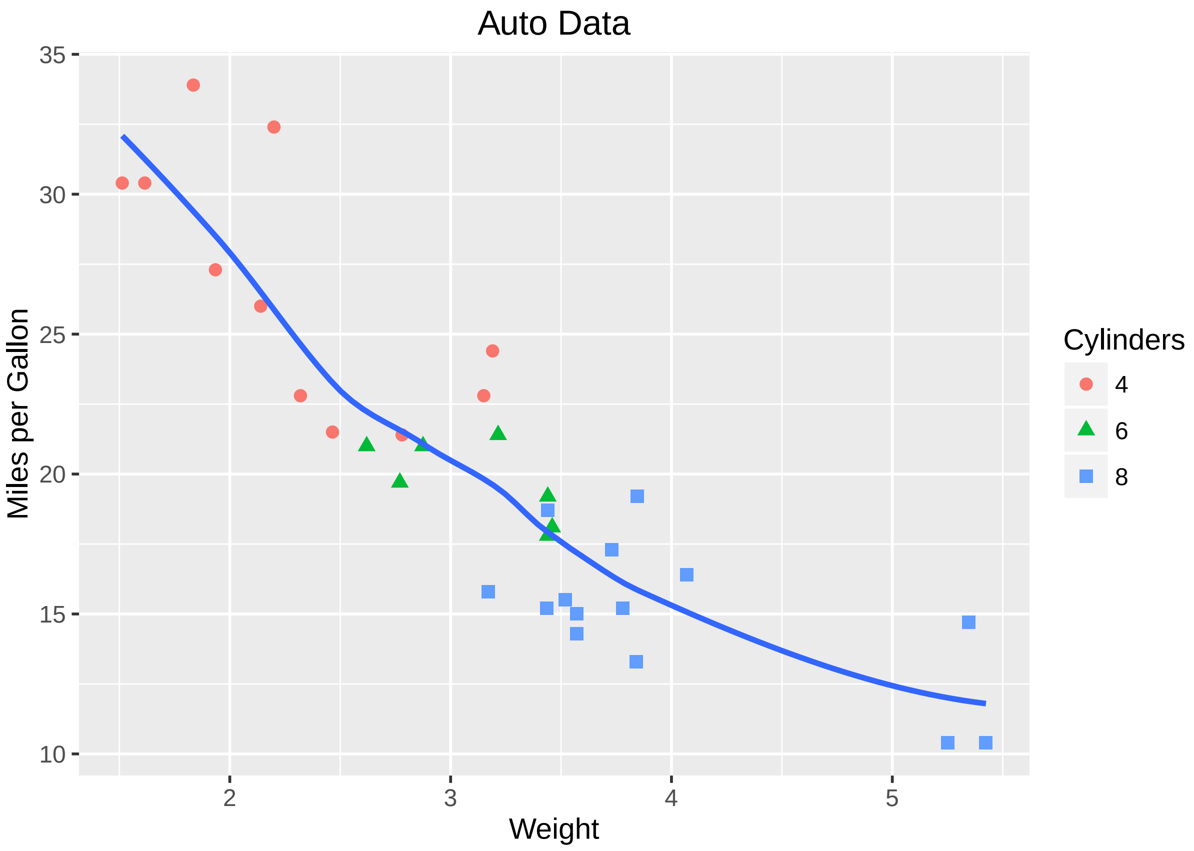
14 Adding a loess smoother

```
ggplot(data=mtcars, aes(x=wt, y=mpg)) +
  geom_point(aes(color=as.factor(cyl), shape=as.factor(cyl)), size=2) +
  geom_smooth() +
  labs(title="Auto Data", x="Weight", y="Miles per Gallon",
       color="Cylinders", shape="Cylinders")
```



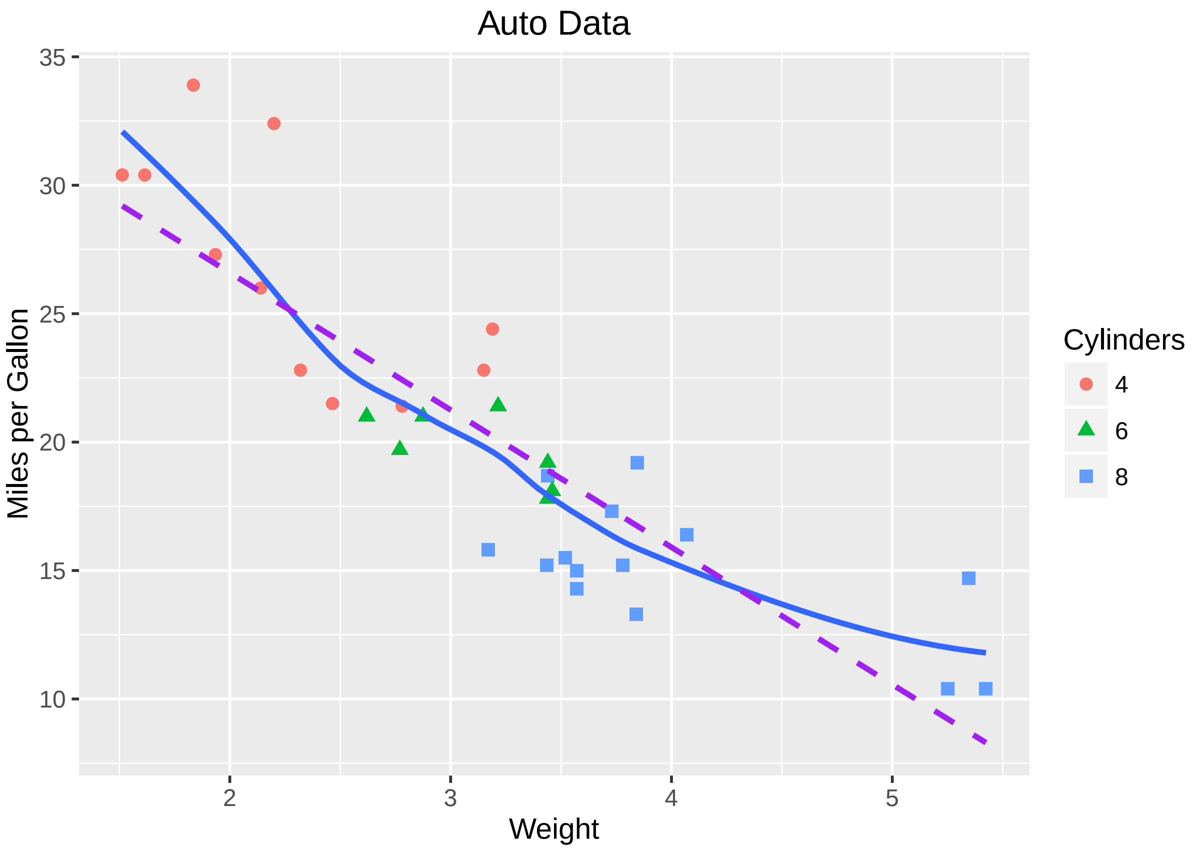
15 Remove the confidence interval

```
ggplot(data=mtcars, aes(x=wt, y=mpg)) +
  geom_point(aes(color=as.factor(cyl), shape=as.factor(cyl)), size=2) +
  geom_smooth(se=FALSE) +
  labs(title="Auto Data", x="Weight", y="Miles per Gallon",
       color="Cylinders", shape="Cylinders")
```



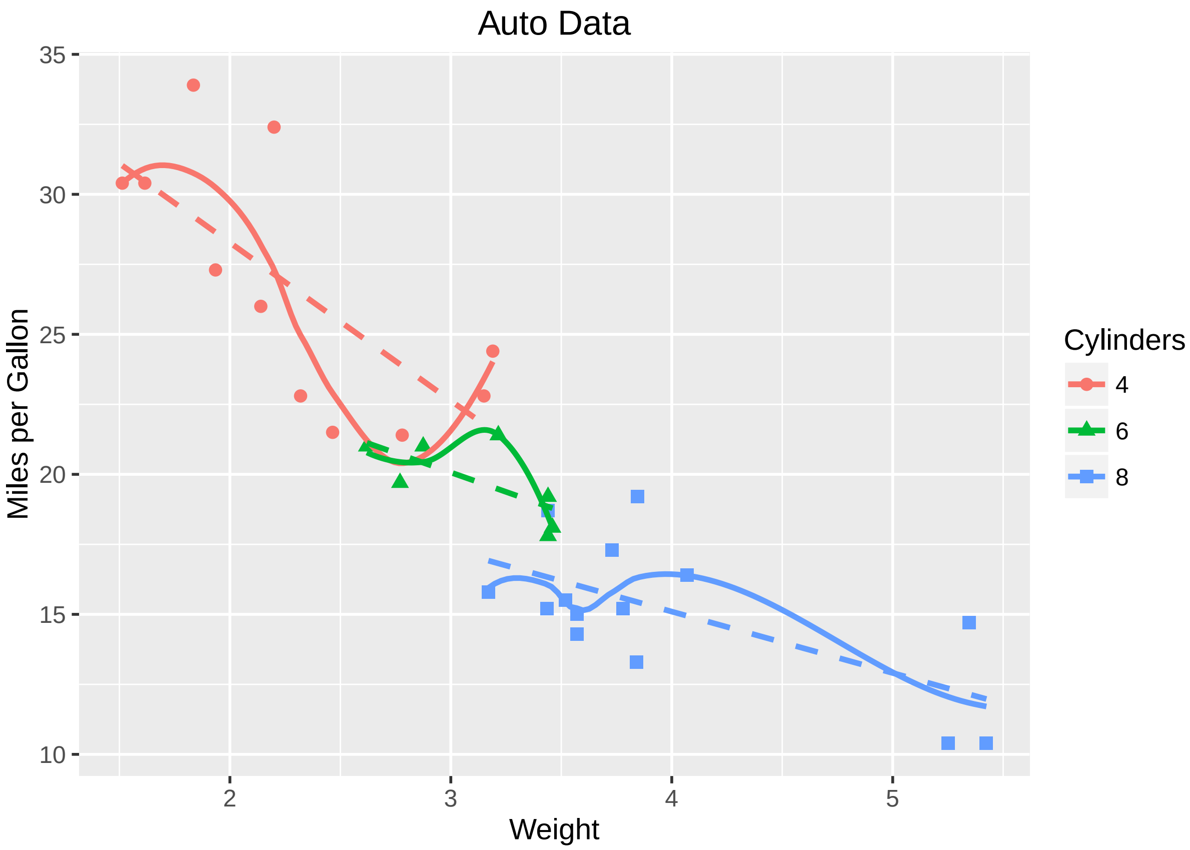
16 Add a linear fit

```
ggplot(data=mtcars, aes(x=wt, y=mpg)) +
  geom_point(aes(color=as.factor(cyl), shape=as.factor(cyl)), size=2) +
  geom_smooth(se=FALSE) +
  geom_smooth(method="lm", se=FALSE, color="purple", lty=2) +
  labs(title="Auto Data", x="Weight", y="Miles per Gallon",
       color="Cylinders", shape="Cylinders")
```



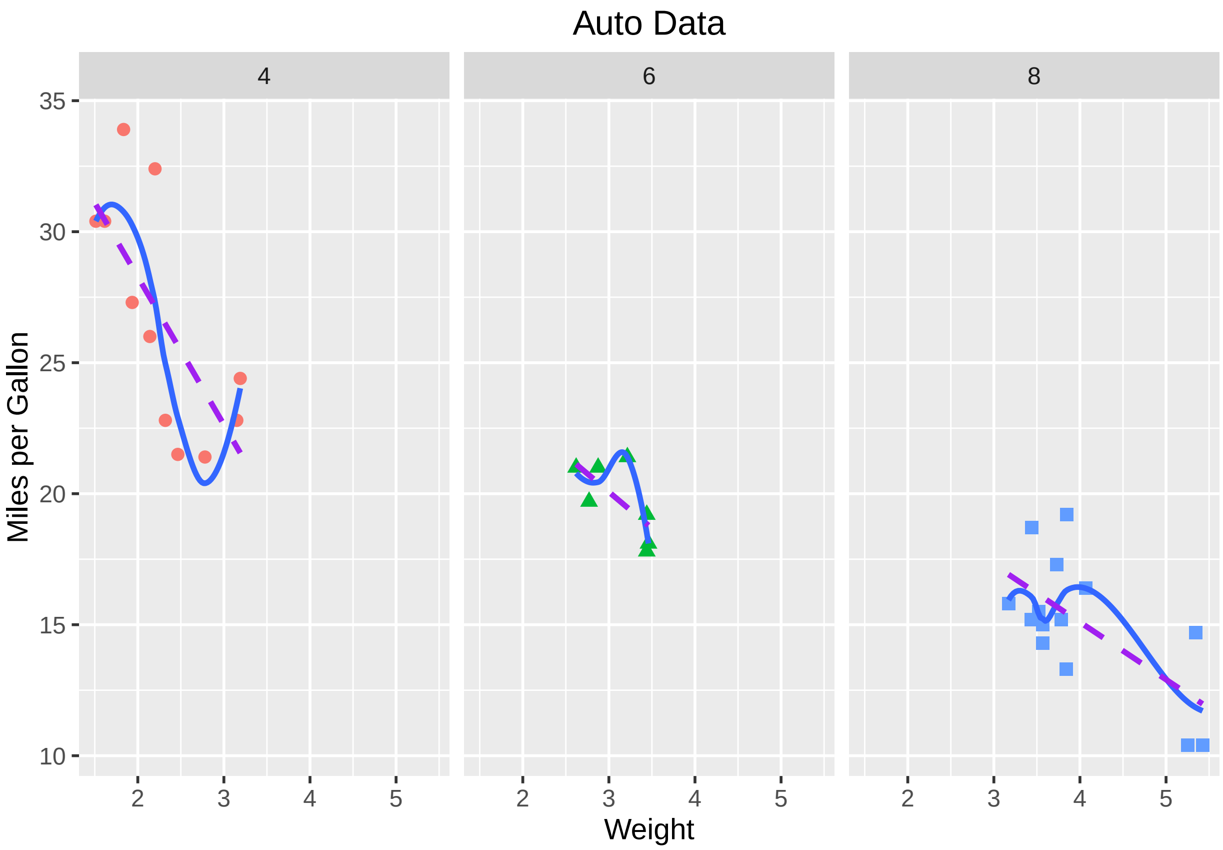
17 One fit per group

```
ggplot(data=mtcars, aes(x=wt, y=mpg)) +
  geom_point(aes(color=as.factor(cyl), shape=as.factor(cyl)), size=2) +
  geom_smooth(se=FALSE, aes(group=as.factor(cyl),
                            color=as.factor(cyl))) +
  geom_smooth(method="lm", se=FALSE, lty=2,
              aes(group=as.factor(cyl),
                  color=as.factor(cyl))) +
  labs(title="Auto Data", x="Weight", y="Miles per Gallon",
       color="Cylinders", shape="Cylinders")
```



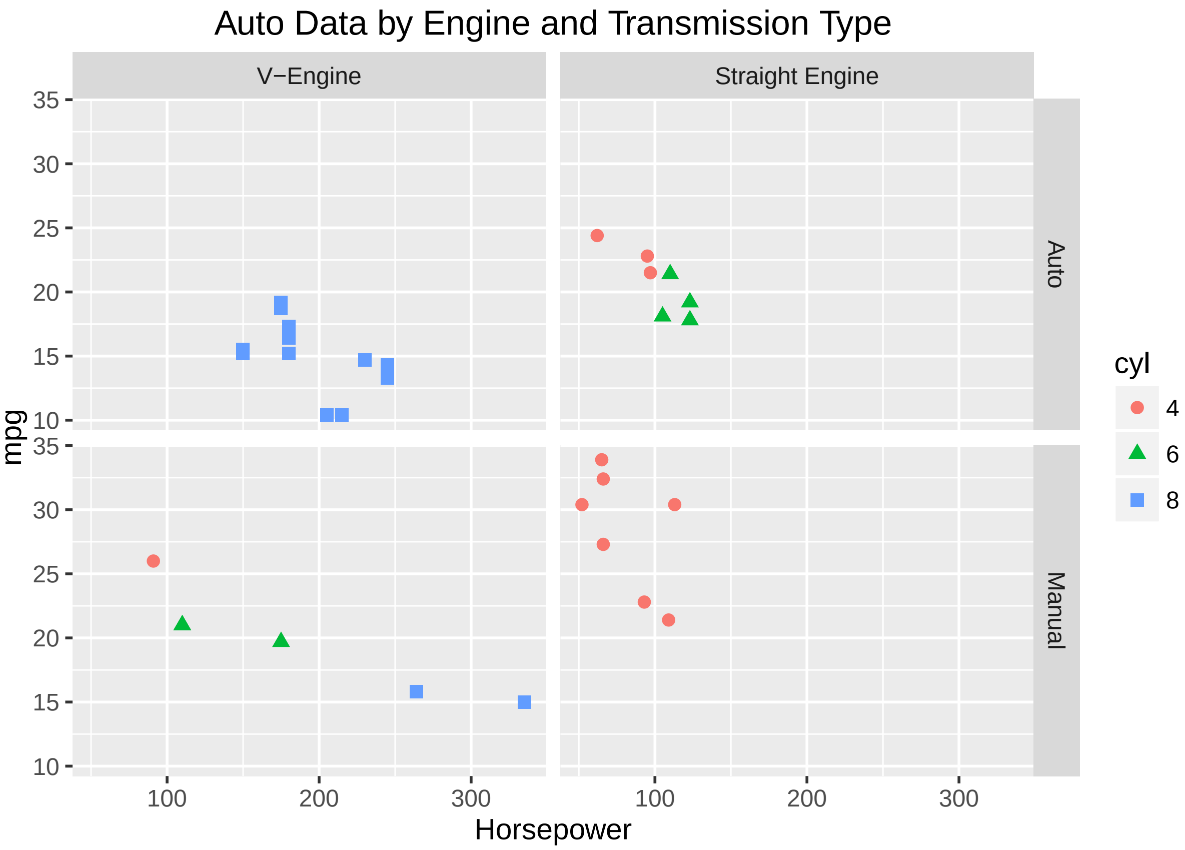
18 Separate the groups with faceting

```
ggplot(data=mtcars, aes(x=wt, y=mpg)) +
  geom_point(aes(color=as.factor(cyl), shape=as.factor(cyl)), size=2) +
  geom_smooth(se=FALSE, aes(group=as.factor(cyl))) +
  geom_smooth(method="lm", se=FALSE, color="purple", lty=2,
              aes(group=as.factor(cyl))) +
  labs(title="Auto Data", x="Weight", y="Miles per Gallon") +
  # remove the legend
  theme(legend.position="none") +
  facet_wrap(~cyl)
```



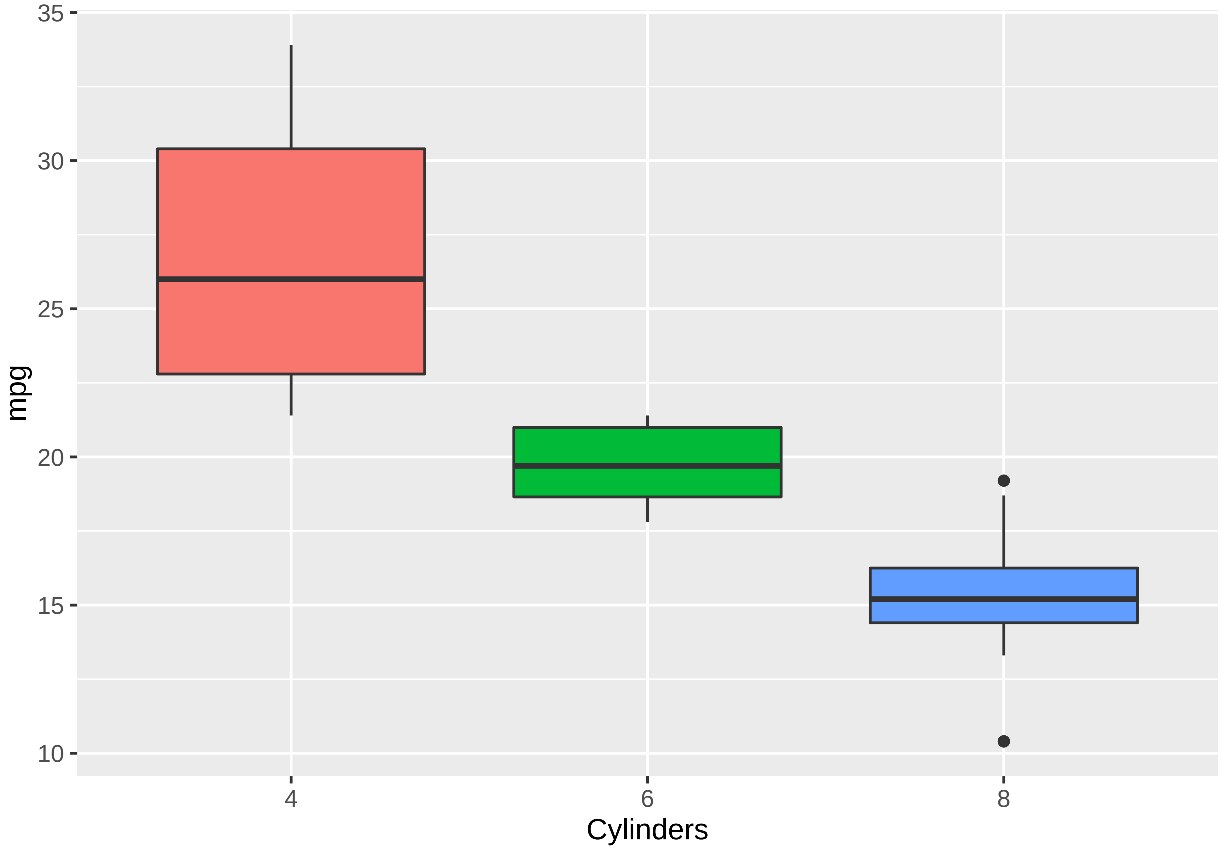
19 More Faceting

```
mtcars$am <- factor(mtcars$am, levels=c(0, 1), labels=c("Auto", "Manual"))
mtcars$vs <- factor(mtcars$vs, levels=c(0, 1), labels=c("V-Engine", "Straight Engine"))
mtcars$cyl <- factor(mtcars$cyl)
ggplot(mtcars, aes(x=hp, y=mpg, color=cyl, shape=cyl)) +
  geom_point(size=2) +
  facet_grid(am ~ vs) +
  labs(title="Auto Data by Engine and Transmission Type", x="Horsepower", y="mpg")
```



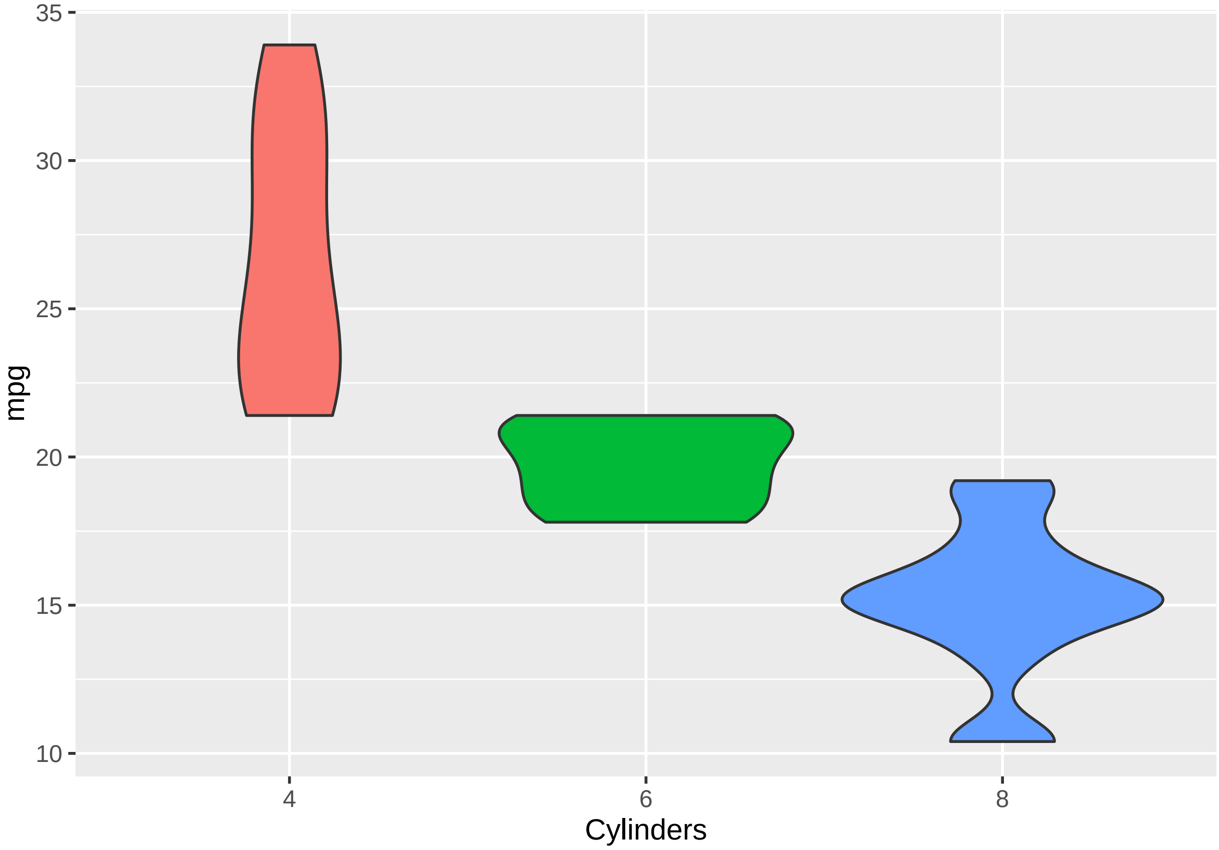
20 Boxplots

```
ggplot(mtcars) +
  geom_boxplot(aes(x=as.factor(cyl), y=mpg, fill=as.factor(cyl))) +
  labs(x="Cylinders") +
  theme(legend.position="none")
```



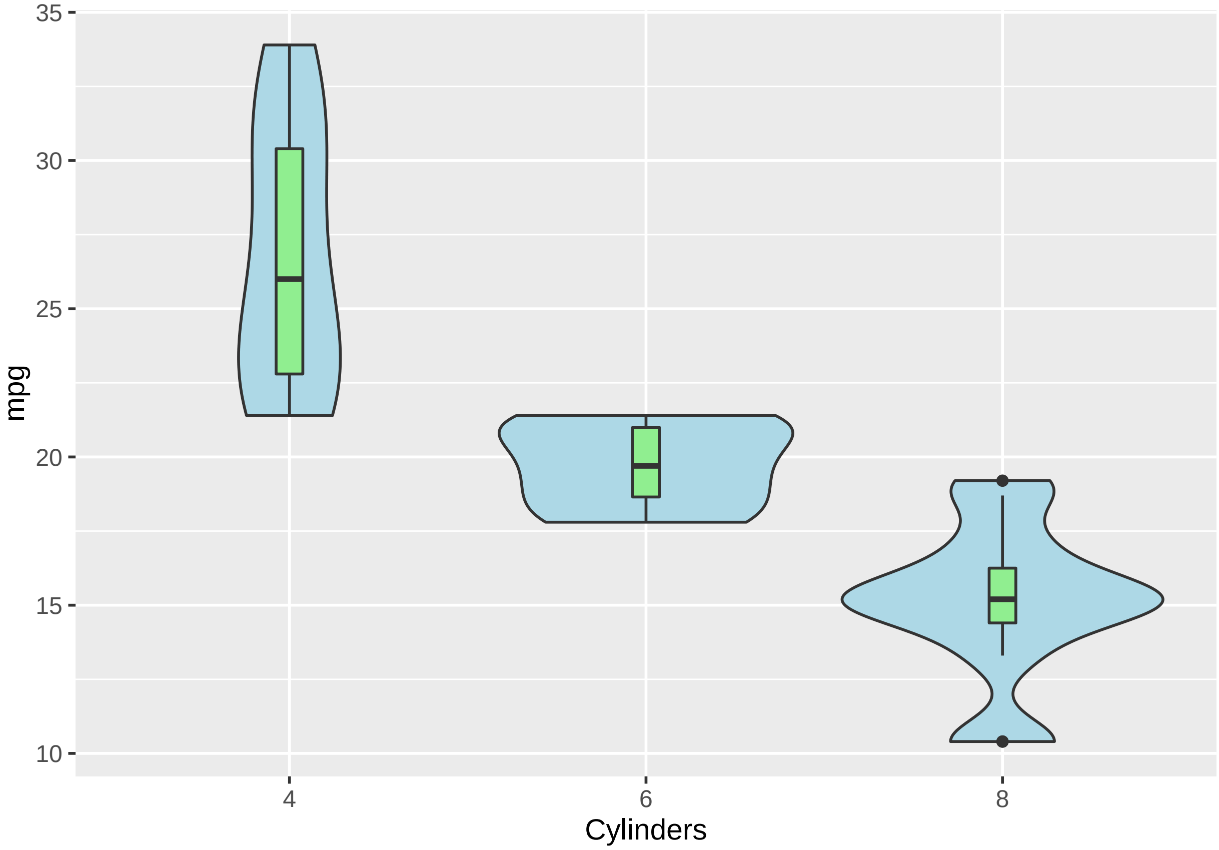
21 Violin Plots

```
ggplot(mtcars) +
  geom_violin(aes(x=as.factor(cyl), y=mpg, fill=as.factor(cyl))) +
  labs(x="Cylinders") +
  theme(legend.position="none")
```



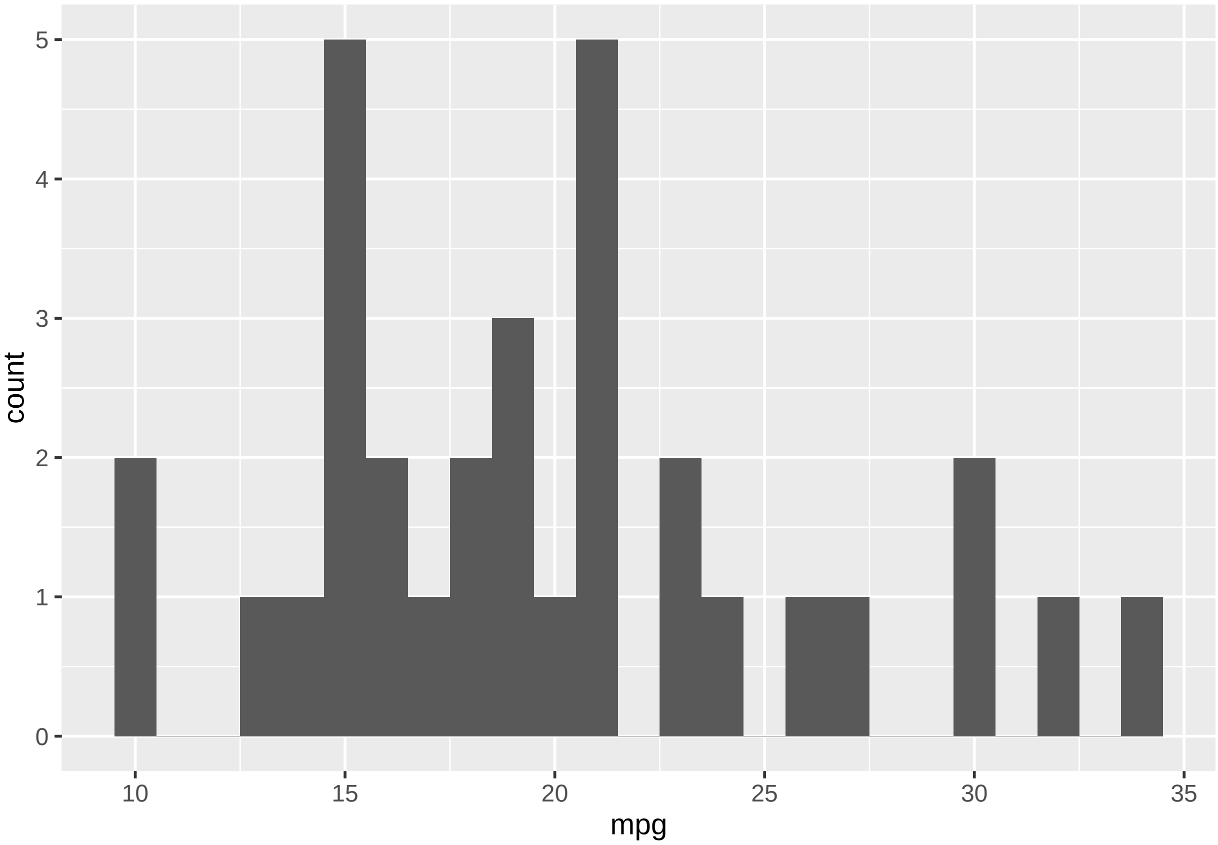
22 Combine boxplot with violin plot

```
ggplot(mtcars) +
  geom_violin(aes(x=as.factor(cyl), y=mpg), fill="lightblue") +
  geom_boxplot(aes(x=as.factor(cyl), y=mpg), fill="lightgreen", width=0.1) +
  labs(x="Cylinders") +
  theme(legend.position="none")
```



23 Histogram

```
ggplot(mtcars) +
  geom_histogram(aes(mpg), binwidth=1)
```

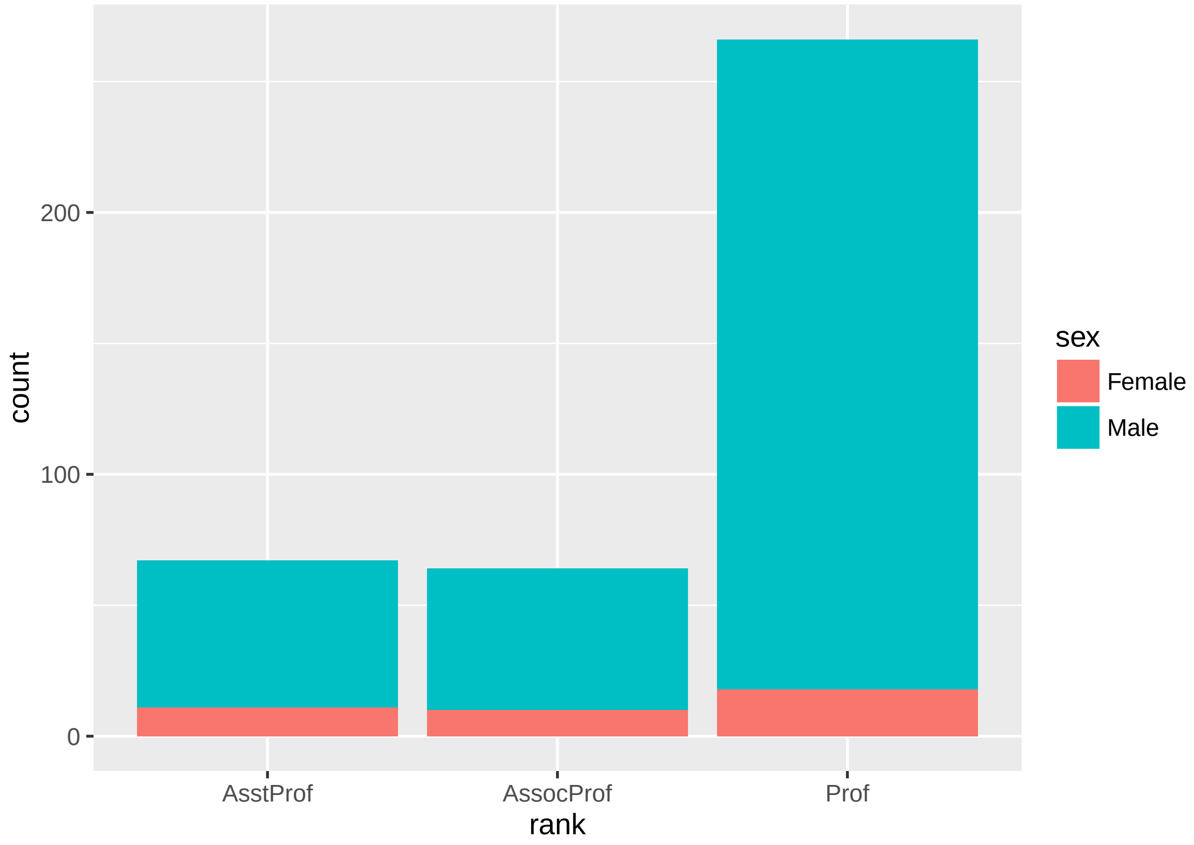


24 Bar Plots

```
library(car)
data(Salaries, package="car")
head(Salaries, 2)
```

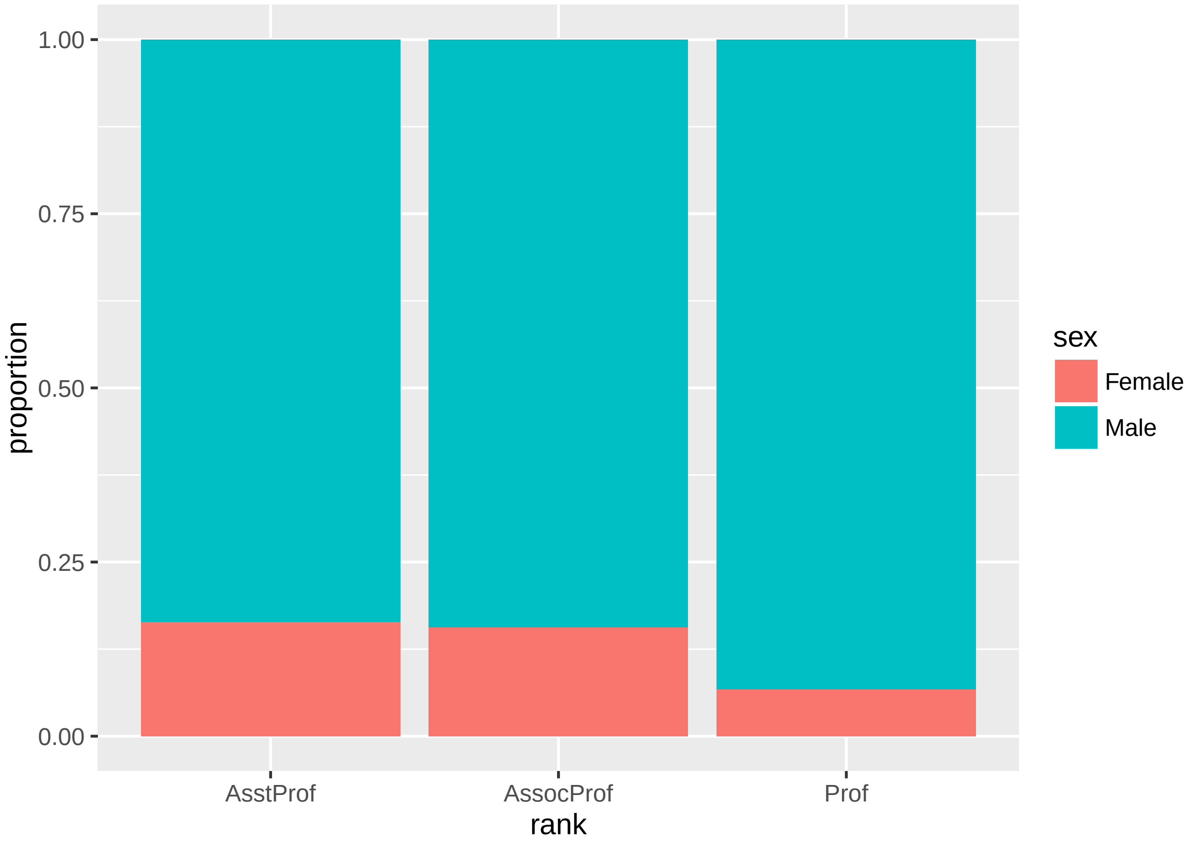
```
##   rank discipline yrs.since.phd yrs.service  sex salary
## 1 Prof          B            19          18 Male 139750
## 2 Prof          B            20          16 Male 173200
```

```
ggplot(Salaries, aes(x=rank, fill=sex)) +
  geom_bar(position="stack")
```



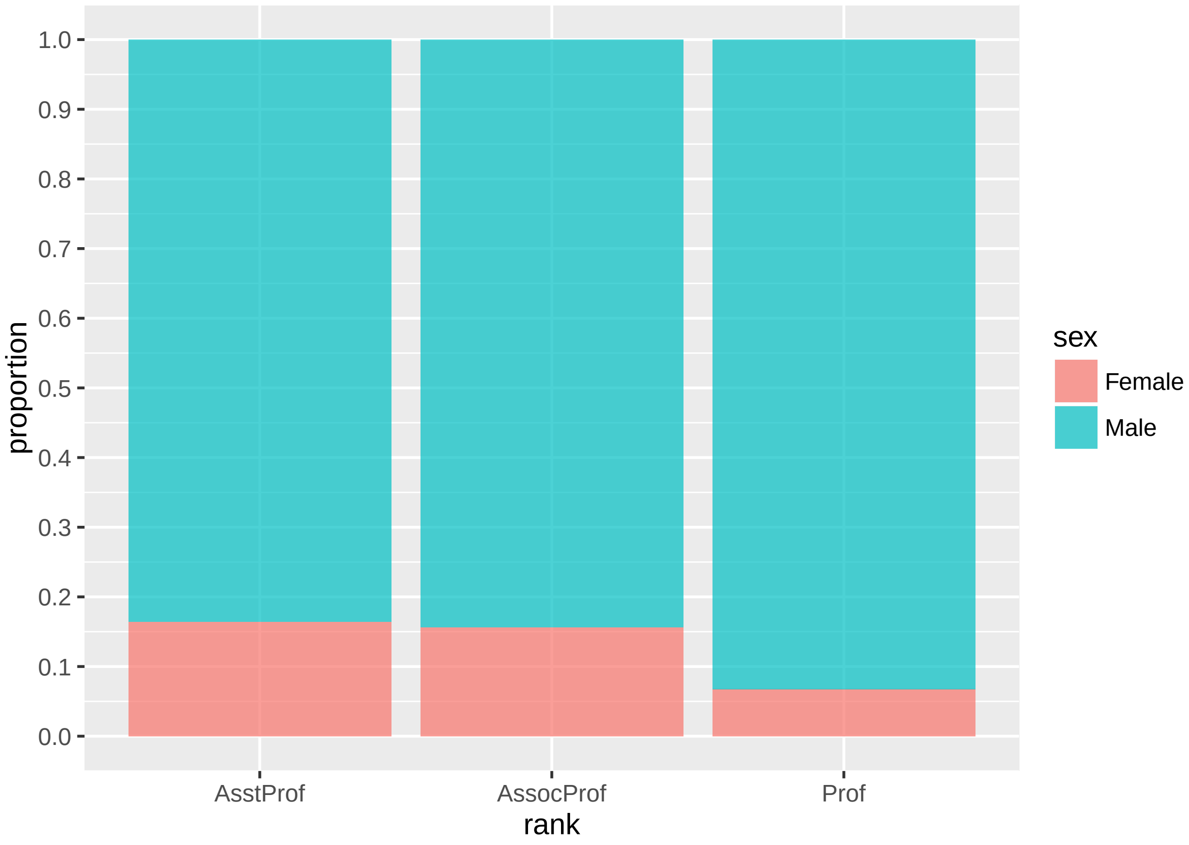
25 Filled Bar Plots

```
ggplot(Salaries, aes(x=rank, fill=sex)) +
  geom_bar(position="fill") +
  # change the default y label 'count' to 'proportion'
  labs(y="proportion")
```



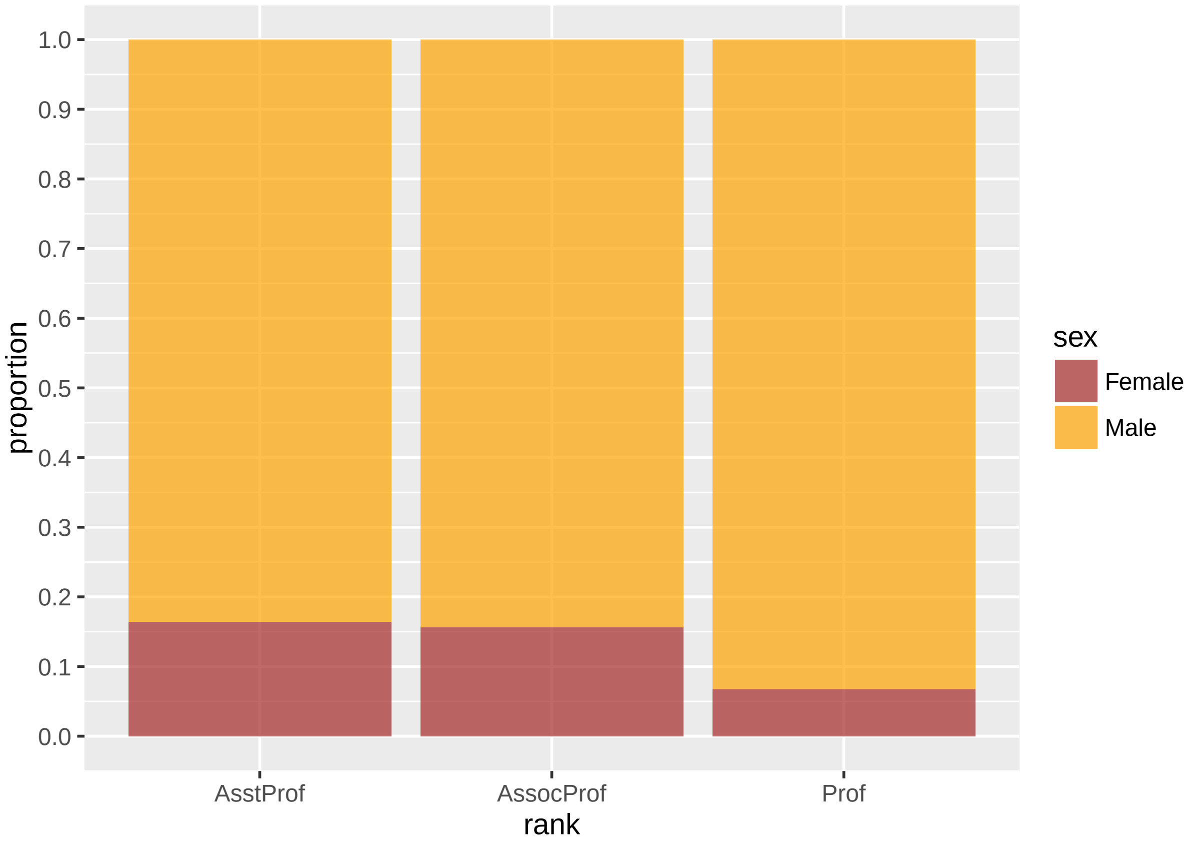
26 Changing the Y-axis breaks

```
ggplot(Salaries, aes(x=rank, fill=sex)) +
  geom_bar(position="fill", alpha=0.7) +
  labs(y="proportion") +
  scale_y_continuous(breaks=seq(0, 1, 0.1))
```



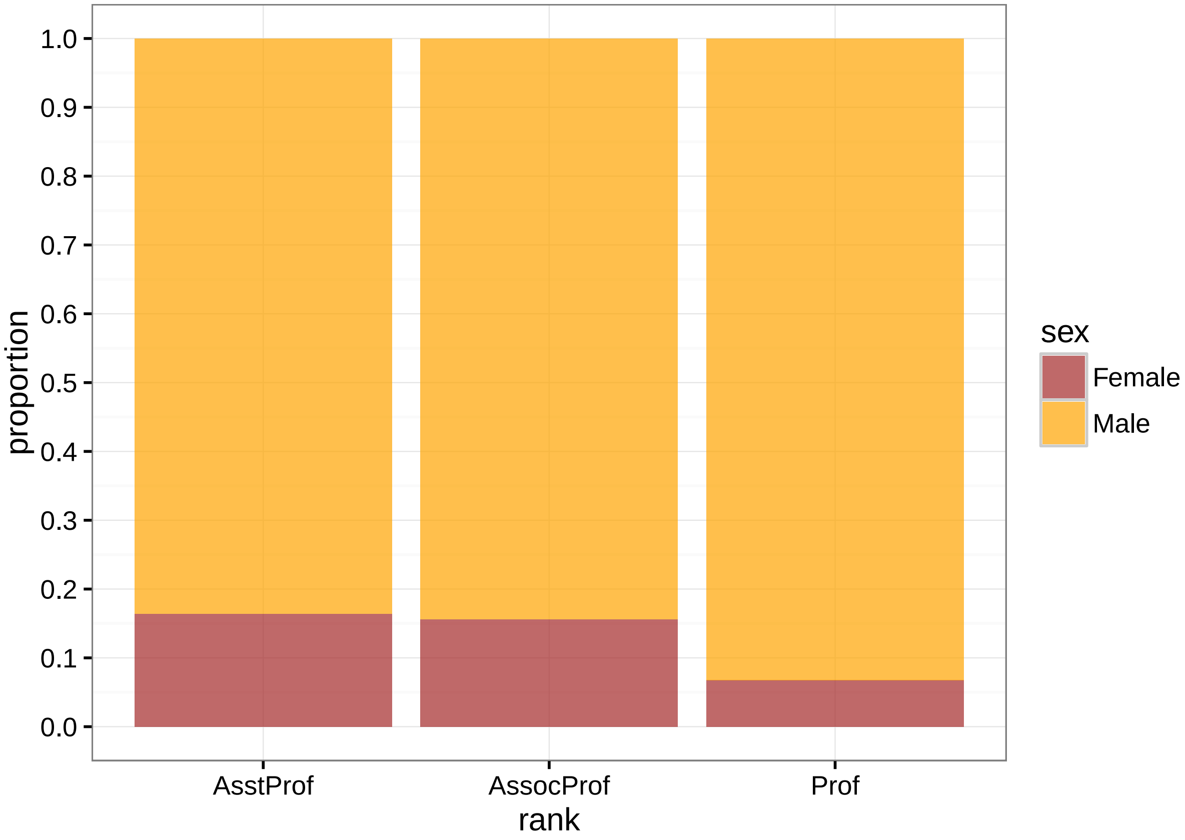
27 Manual colors

```
ggplot(Salaries, aes(x=rank, fill=sex)) +
  geom_bar(position="fill", alpha=0.7) +
  labs(y="proportion") +
  scale_y_continuous(breaks=seq(0, 1, 0.1)) +
  scale_fill_manual(values = c("brown", "orange"))
```



28 Black and White Theme

```
ggplot(Salaries, aes(x=rank, fill=sex)) +
  geom_bar(position="fill", alpha=0.7) +
  labs(y="proportion") +
  scale_y_continuous(breaks=seq(0, 1, 0.1)) +
  scale_fill_manual(values = c("brown", "orange")) +
  theme_bw()
```



29 Make ggplot interactive with plotly

```
library(plotly)
p <- ggplot(data=mtcars, aes(x=wt, y=mpg)) +
  geom_point(aes(color=as.factor(cyl), shape=as.factor(cyl)),
             size=2) +
  labs(title="Auto Data", x="Weight", y="Miles per Gallon",
       color="Cylinders", shape="Cylinders")
ggplotly(p)
```

30 Change the hover text

```
library(plotly)
mtcars$rowName <- row.names(mtcars)
p <- ggplot(data=mtcars, aes(x=wt, y=mpg)) +
  geom_point(aes(color=as.factor(cyl),
                 shape=as.factor(cyl),
                 text=paste(rowName)),
             size=3) +
  labs(title="Auto Data", x="Weight", y="Miles per Gallon",
       color="Cylinders", shape="Cylinders")
ggplotly(p)
```

Learning more

- Hadley Wickham, *ggplot2: Elegant Graphics for Data Analysis*, 2/e, Springer, 2016.
- Robert I Kabacoff, *R in action*, 2/e (ch 19), Manning, 2015.
- Roger Peng, *Coursera: Exploratory Data Analysis*
- ggplot2 docs: http://docs.ggplot2.org/current/
- plotly: https://plot.ly/ggplot2/