Introduction to Data Journalism

ggplot2 part 2

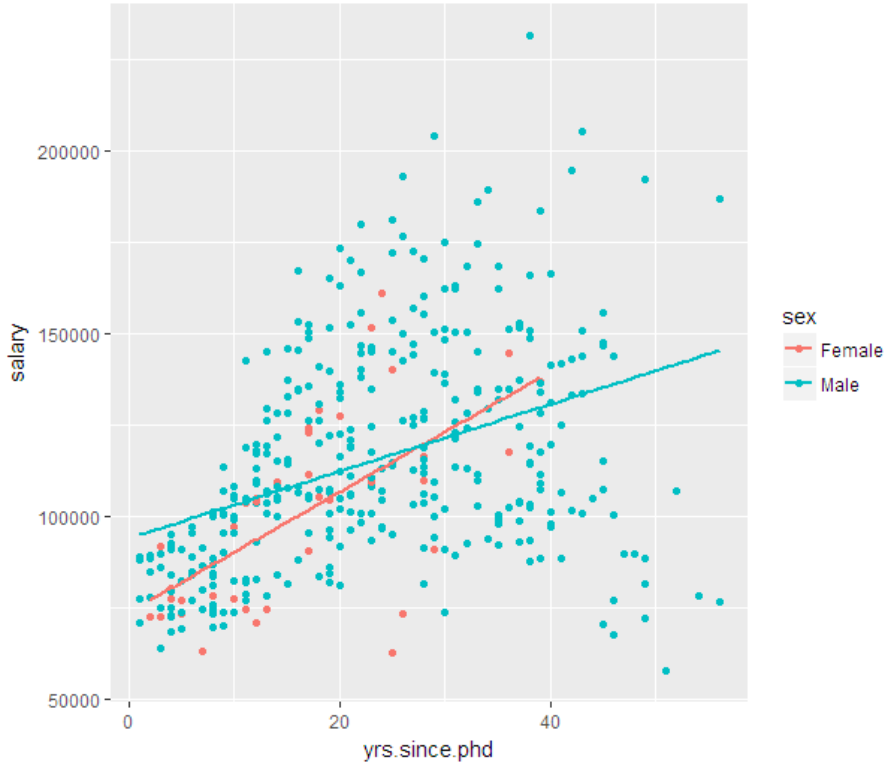
Grammar of Graphics

- data: an R data frame
- coordinate system: 2-D space data projected onto (e.g. cartesiona coordinates, polar coordinates, map projections)
- geoms: type of geometric objects that represent data (e.g. points, lines, bars)
- aesthetics: visual characteristics that represent data (e.g. position, size, color, shape, transparency, fill)
- scales: for each aesthetic, how visual characteristic is converted to display values
- stats: statistical transformations that summarize data (e.g., counts, means, trend lines)
- facets: how data is split into subsets and displayed as small multiples

Thanks to Dawn Koffman ggplot2 presentation

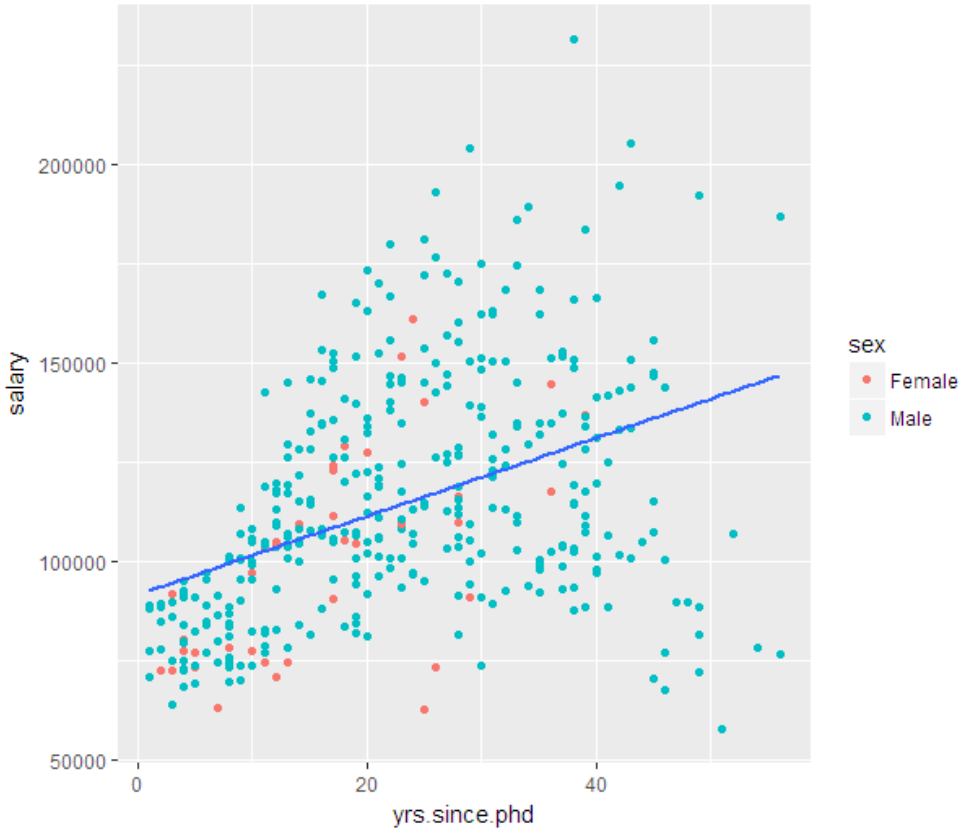
Aesthetics in ggplot( ) vs geom_xxx( )

```
library(ggplot2)
data(Salaries, package="car")
ggplot(data=Salaries, aes(x=yrs.since.phd, y=salary, color=sex )) +
  geom_point() +
  geom_smooth(method="lm", se=FALSE)
```



Aesthetics in ggplot( ) vs geom_xxx( )

```
ggplot(data=Salaries, aes(x=yrs.since.phd, y=salary )) +
  geom_point(aes(color=sex)) +
  geom_smooth(method="lm", se=FALSE)
```

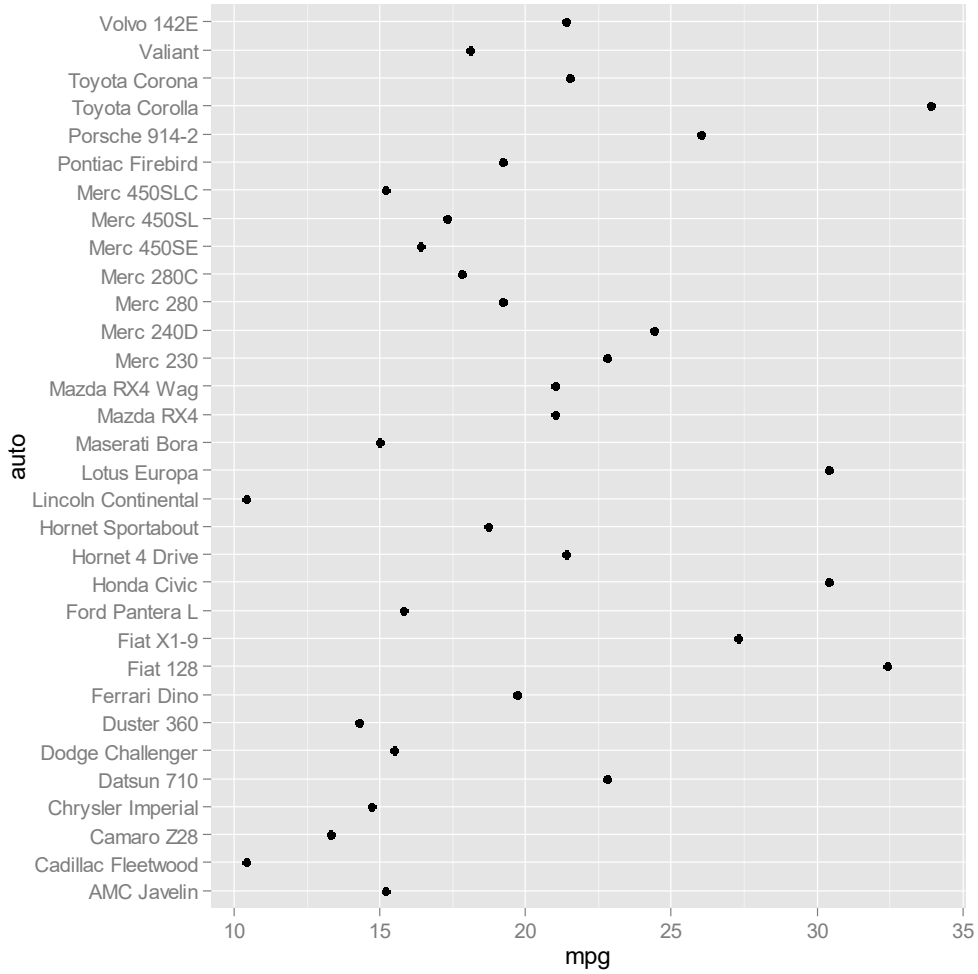


Geoms

geom_abline	Reference lines: horizontal, vertical, and diagonal
geom_bar	Bars charts
geom_bin2d	Heatmap of 2d bin counts
geom_blank	Draw nothing
geom_boxplot	A box and whiskers plot (in the style of Tukey)
geom_contour	2d contours of a 3d surface
geom_count	Count overlapping points
geom_density	Smoothed density estimates
geom_density_2d	Contours of a 2d density estimate
geom_dotplot	Dot plot
geom_errorbarh	Horizontal error bars
geom_hex	Hexagonal heatmap of 2d bin counts
geom_freqpoly	Histograms and frequency polygons
geom_jitter	Jittered points
geom_crossbar	Vertical intervals: lines, crossbars & errorbars
geom_map	Polygons from a reference map
geom_path	Connect observations
geom_point	Points
geom_polygon	Polygons
geom_qq	A quantile-quantile plot
geom_quantile	Quantile regression
geom_ribbon	Ribbons and area plots
geom_rug	Rug plots in the margins
geom_segment	Line segments and curves
geom_smooth	Smoothed conditional means
geom_spoke	Line segments parameterised by location, direction and distance
geom_label	Text
geom_raster	Rectangles
geom_violin	Violin plot

Dot plot

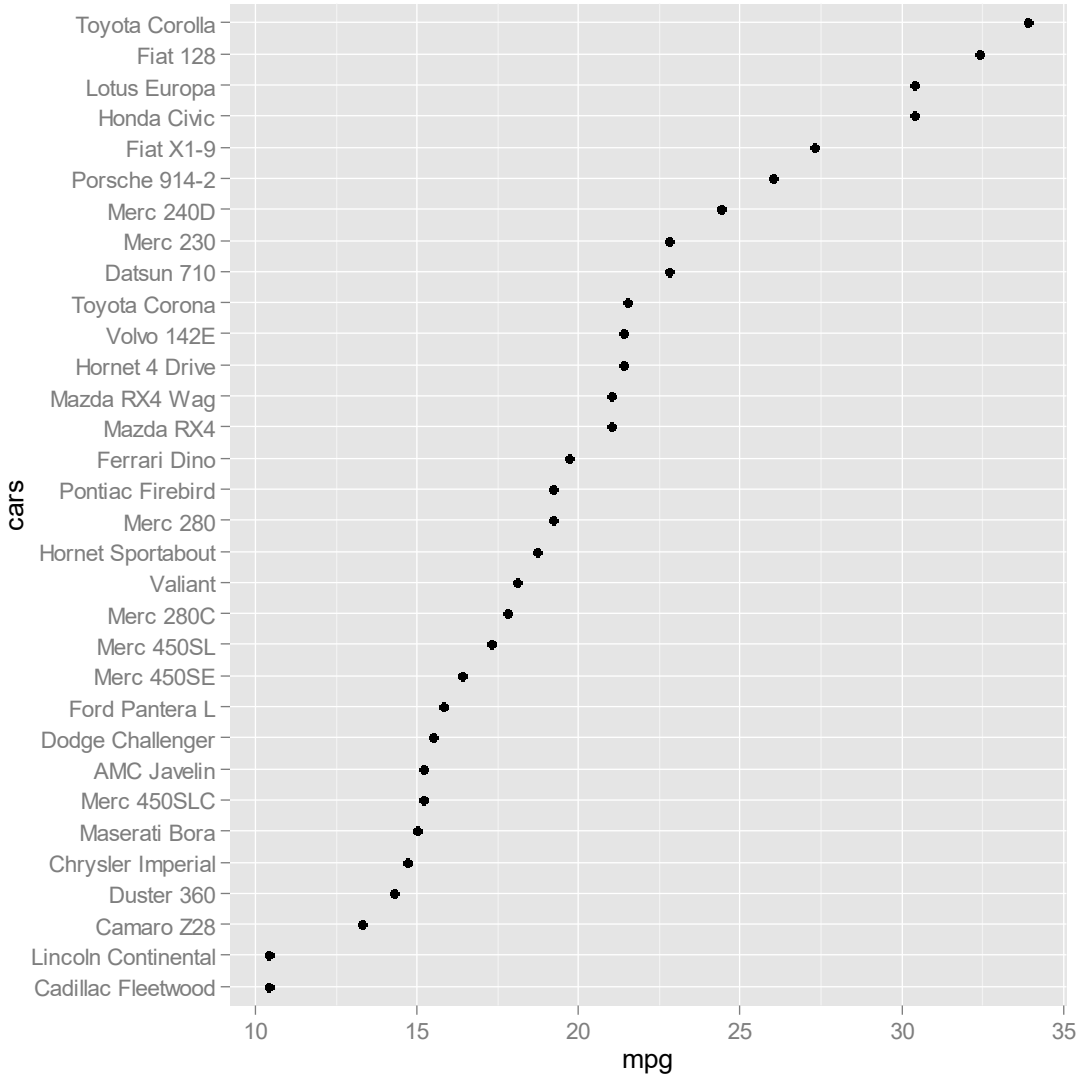
```
df <- mtcars
df$cars <- row.names(df)

ggplot(data=df,
   aes(x=mpg, y=cars)) +
   geom_point()
```



Sorted Dot plot

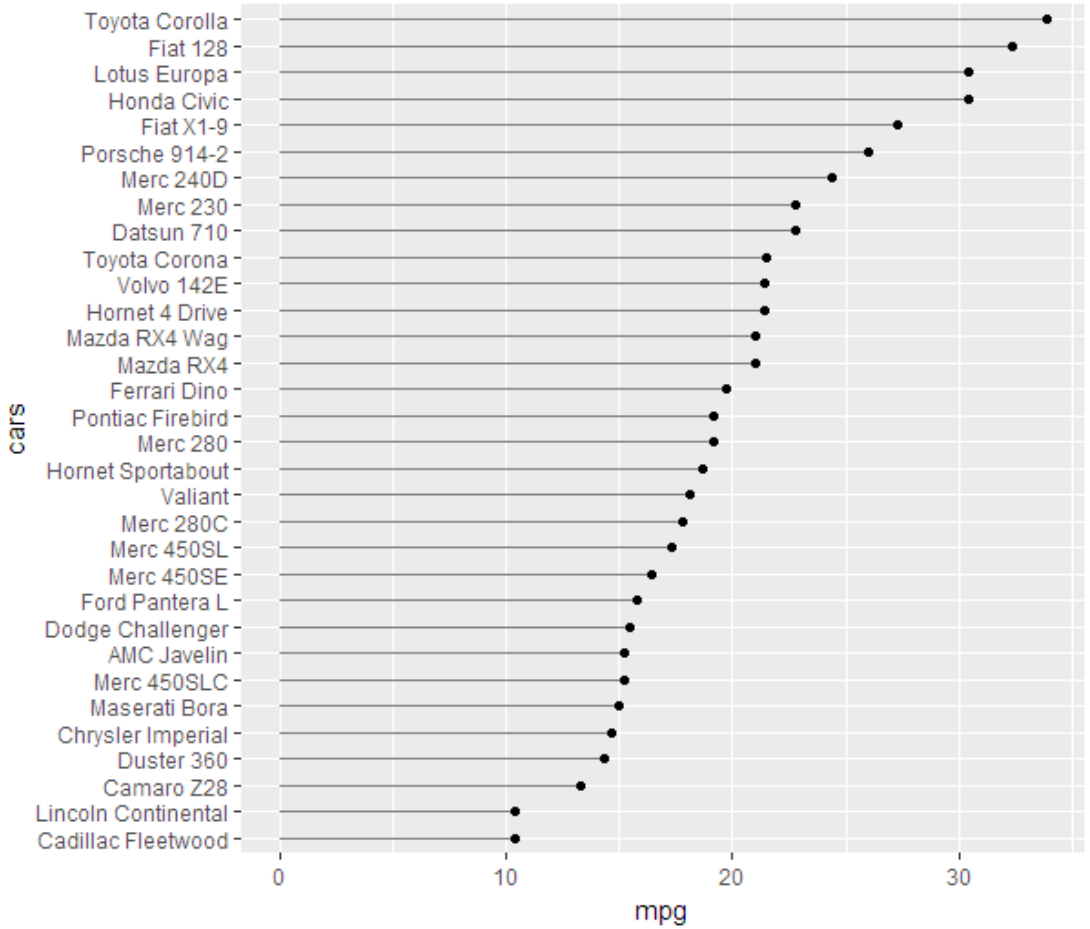
```
mtcars <- mutate(mtcars,
                 cars = row.names(mtcars),
                 cars = factor(cars, levels = cars[order(mpg)])
)

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

Sorted Dot Plot



Lollypop Plot

```
ggplot(data=mtcars,
  aes(x=mpg, y=cars)) +
  geom_segment(
     aes(x = 0,
         y = cars,
         xend = mpg,
         yend = cars),
      color = "grey50") +
  geom_point()
```



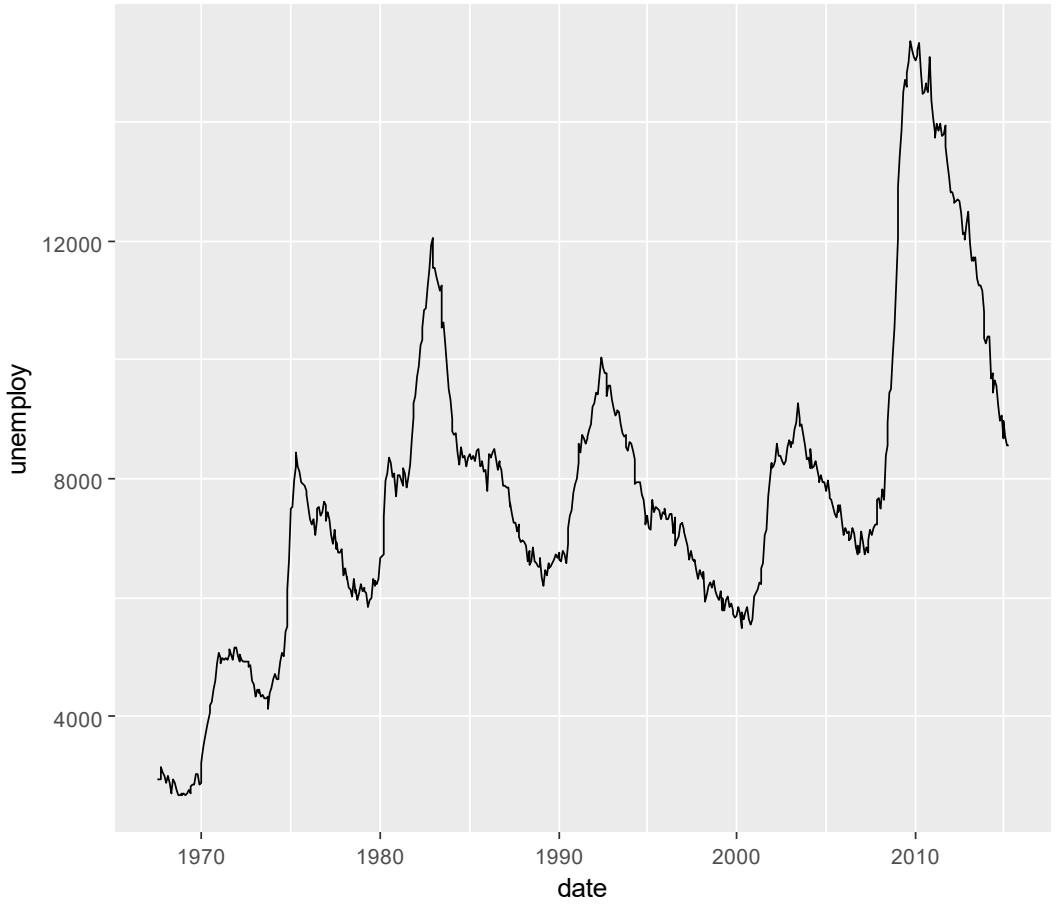
Changing point shapes

```
+ geom_point(shape = 15)
```

for 21-25 you
can control both
the fill and the border

Line charts

```
ggplot(economics, aes(date, unemploy)) + geom_line()
```

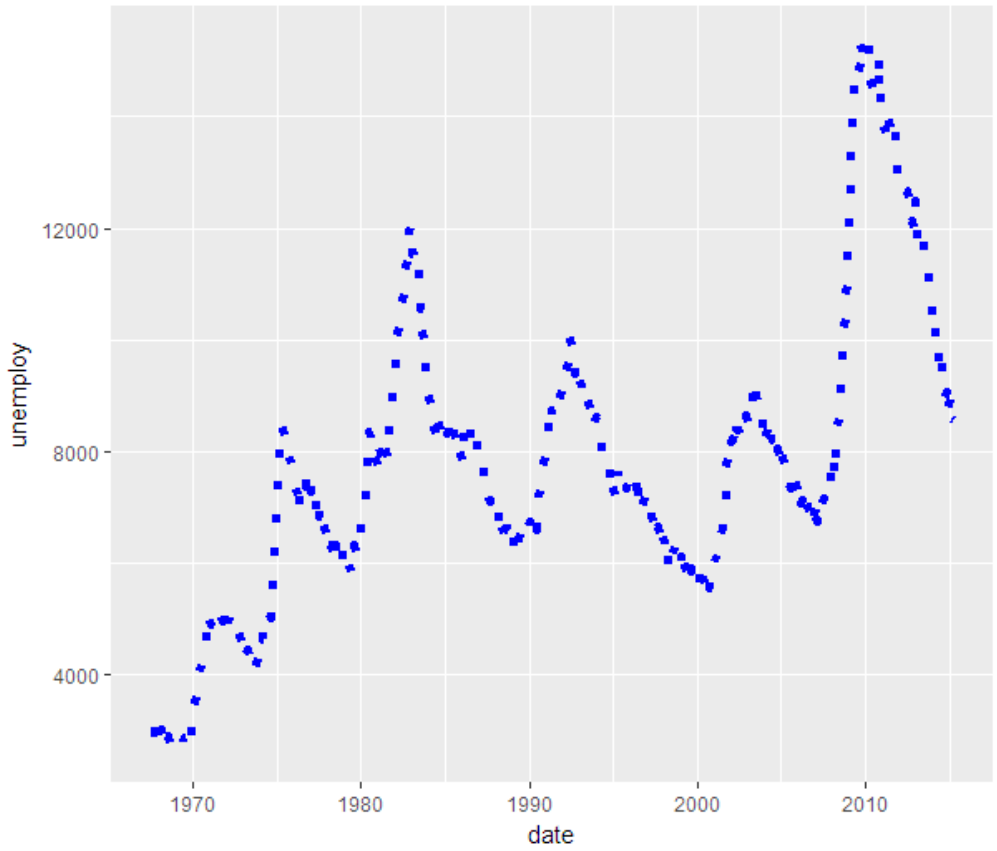


Line charts

Changing the linetype

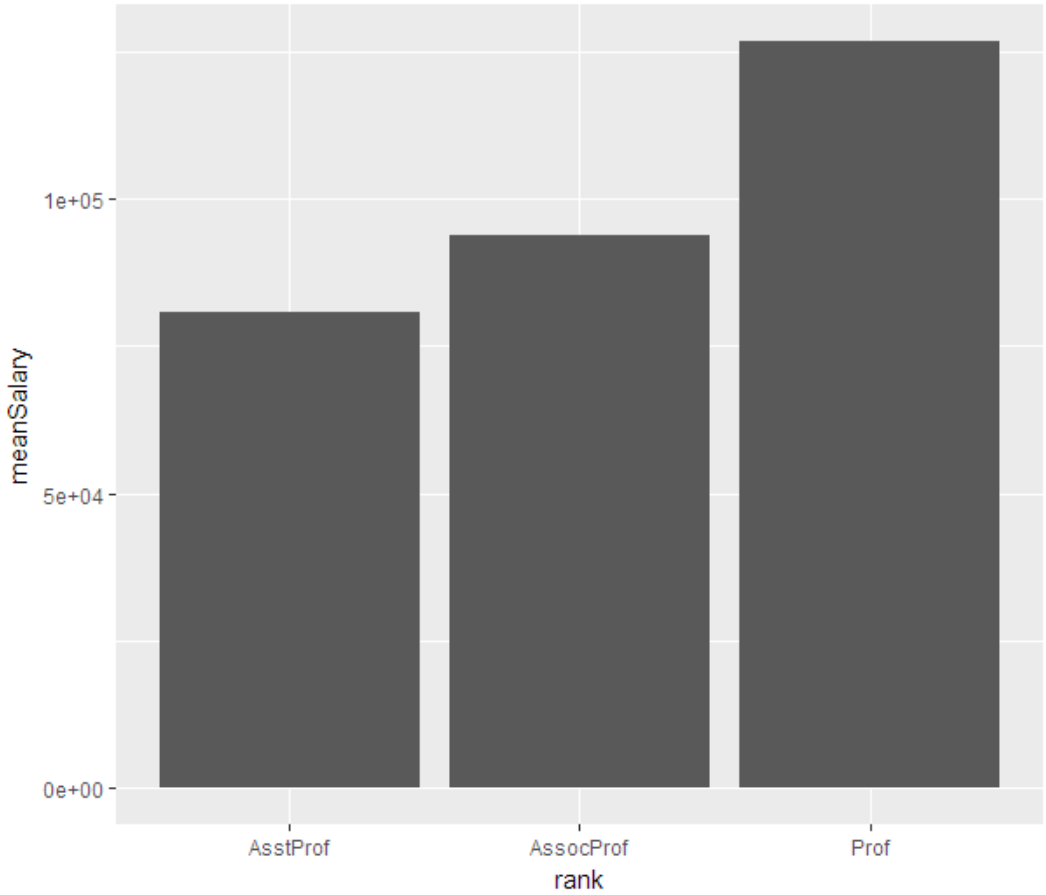
Line charts

```
ggplot(economics, aes(date, unemploy)) +
geom_line(linetype="dotted", color="blue", size=1)
```



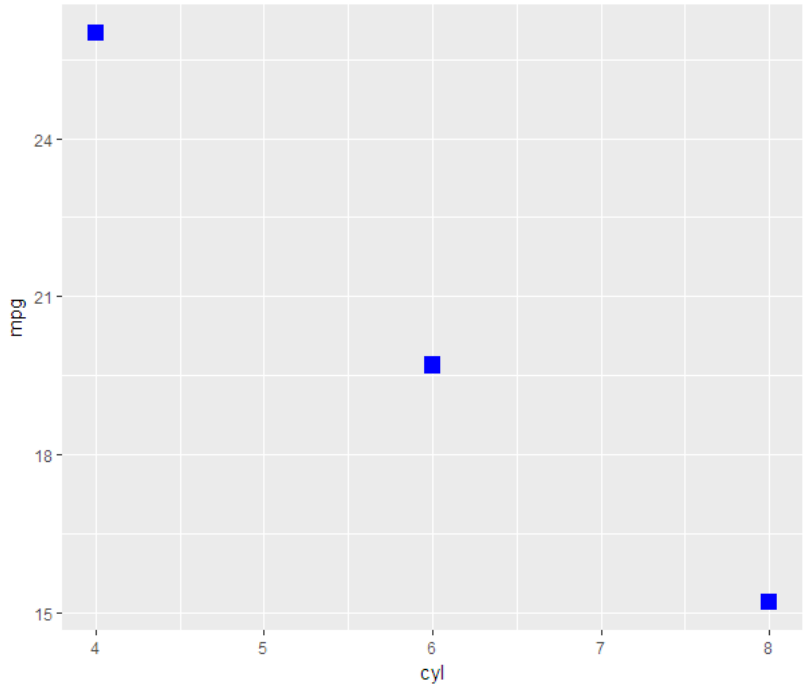
Using ggplot2 stat functions

```
ggplot(data=Salaries, aes(x=rank, y=salary)) +
stat_summary(geom="bar", fun.y=mean)
```



Using stat_summary

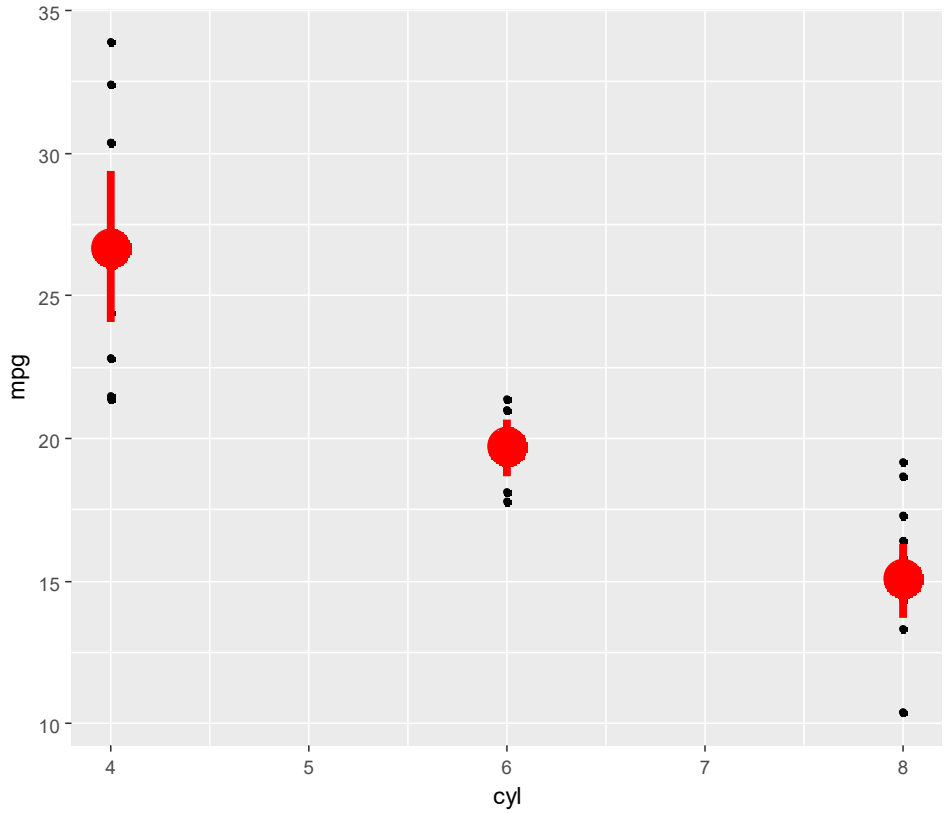
```
ggplot(mtcars, aes(cyl, mpg)) +
stat_summary(fun.y = "median", color = "blue", size = 4,
                    geom = "point", shape=15)
```



Using stat_summary

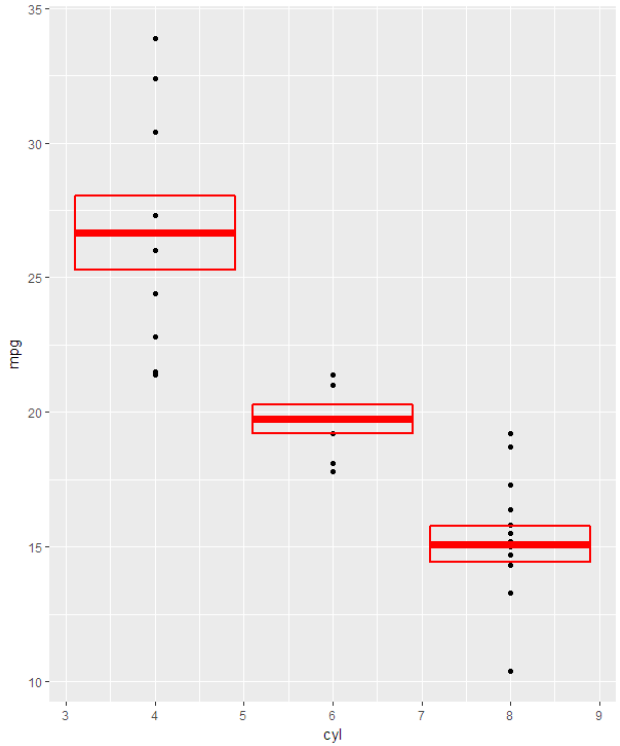
```
d <- ggplot(mtcars, aes(cyl, mpg)) + geom_point()
d + stat_summary(fun.data = "mean_cl_boot", geom="pointrange",
                 colour = "red", size = 2)
```

also try
mean_se
median_hilow



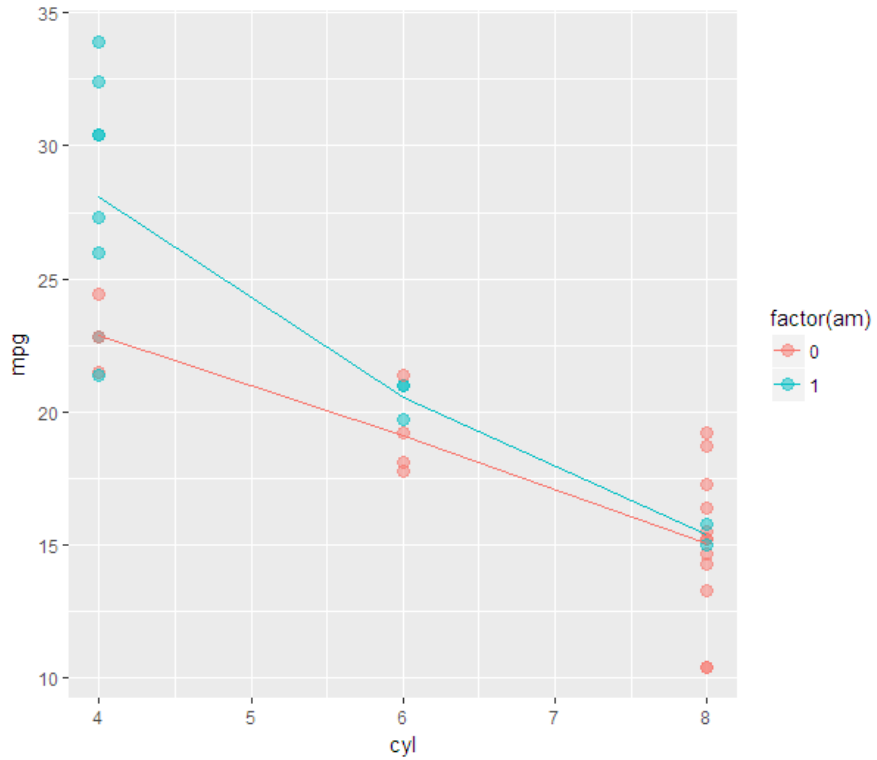
Using stat_summary

```
d <- ggplot(mtcars, aes(cyl, mpg)) + geom_point()
d + stat_summary(fun.data = "mean_se", geom="crossbar",
                 colour = "red", size = 2)
```



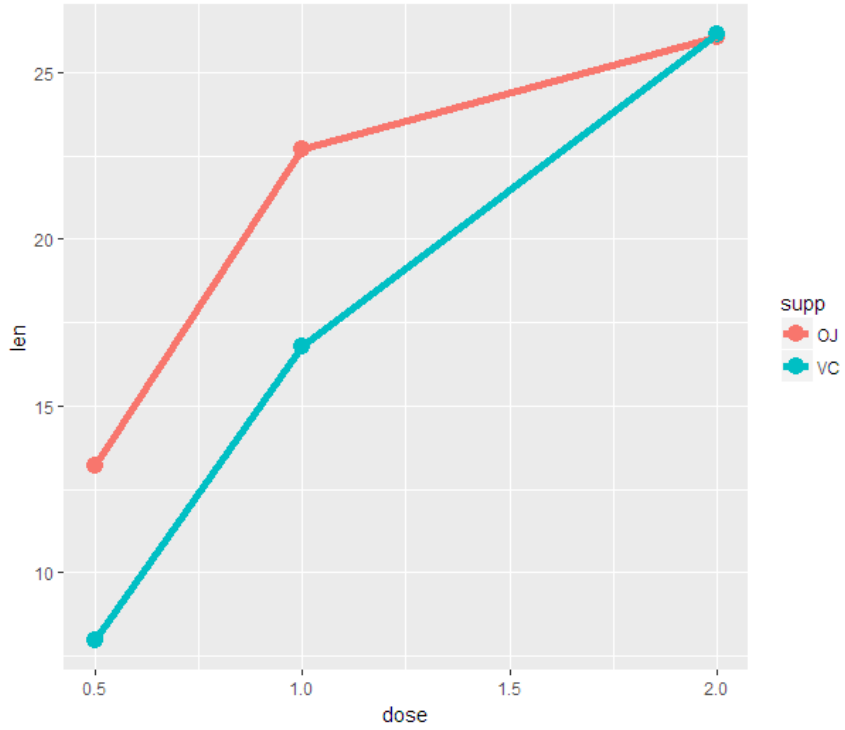
Using stat_summary

```
d <- ggplot(mtcars, aes(cyl, mpg)) +
            geom_point(size=3, alpha=.5)
d + aes(colour = factor(am)) +
    stat_summary(fun.y = mean, geom="line")
```



Using stat_summary

```
ggplot(data=ToothGrowth,
       aes(x=dose, y=len, color=supp)) +
       stat_summary(fun.y=mean, geom="point", size=4) +
       stat_summary(fun.y=mean, geom="line", size=2)
```

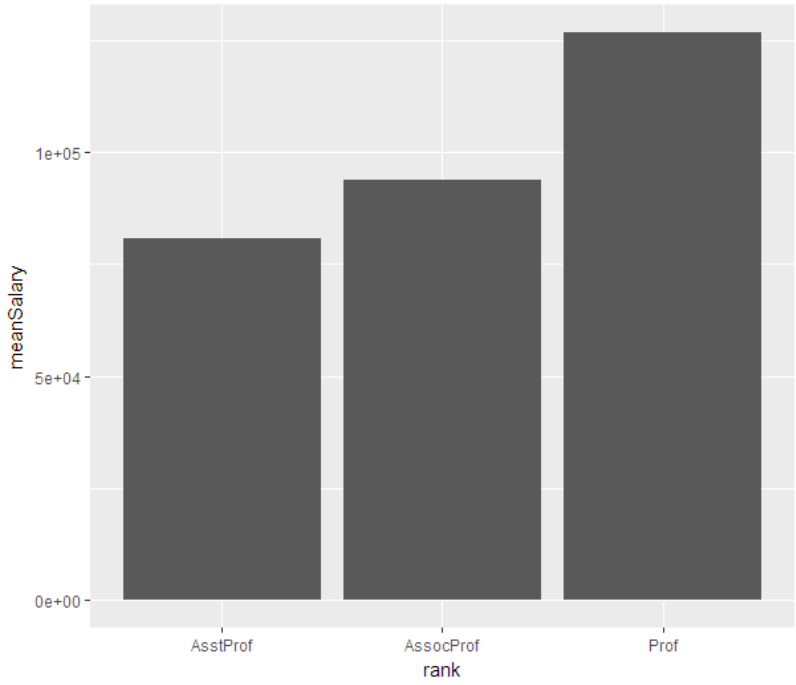


Using ggplot2 on summarized data

```
data(Salaries, package="car")
df <- group_by(Salaries, rank) %>%
      summarise(meanSalary = mean(salary))
```

```
      rank meanSalary
    <fctr>      <dbl>
1  AsstProf   80775.99
2 AssocProf   93876.44
3      Prof  126772.11
```

```
ggplot(data=df,
  aes(x=rank, y=meanSalary)) +
  geom_bar(stat="identity")
```



Facets

```
facets_grid( rowvar ~ colvar)
```

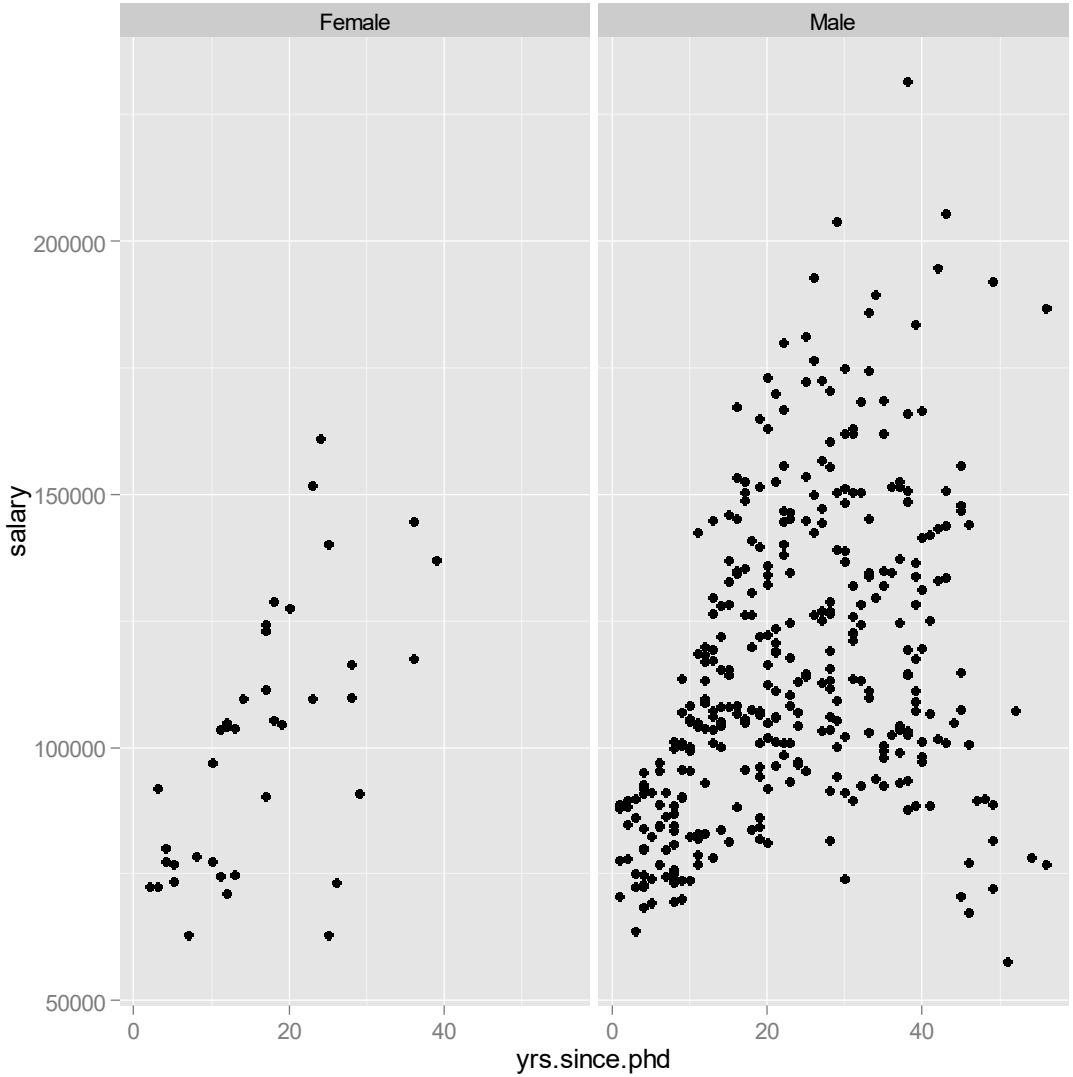
```
facets_grid( . ~ colvar)
```
just columns

```
facets_grid(rowvar ~ .)
```
just rows

```
facets_wrap(~ var, ncol=#)
```
one classification variable wrapped to fill page

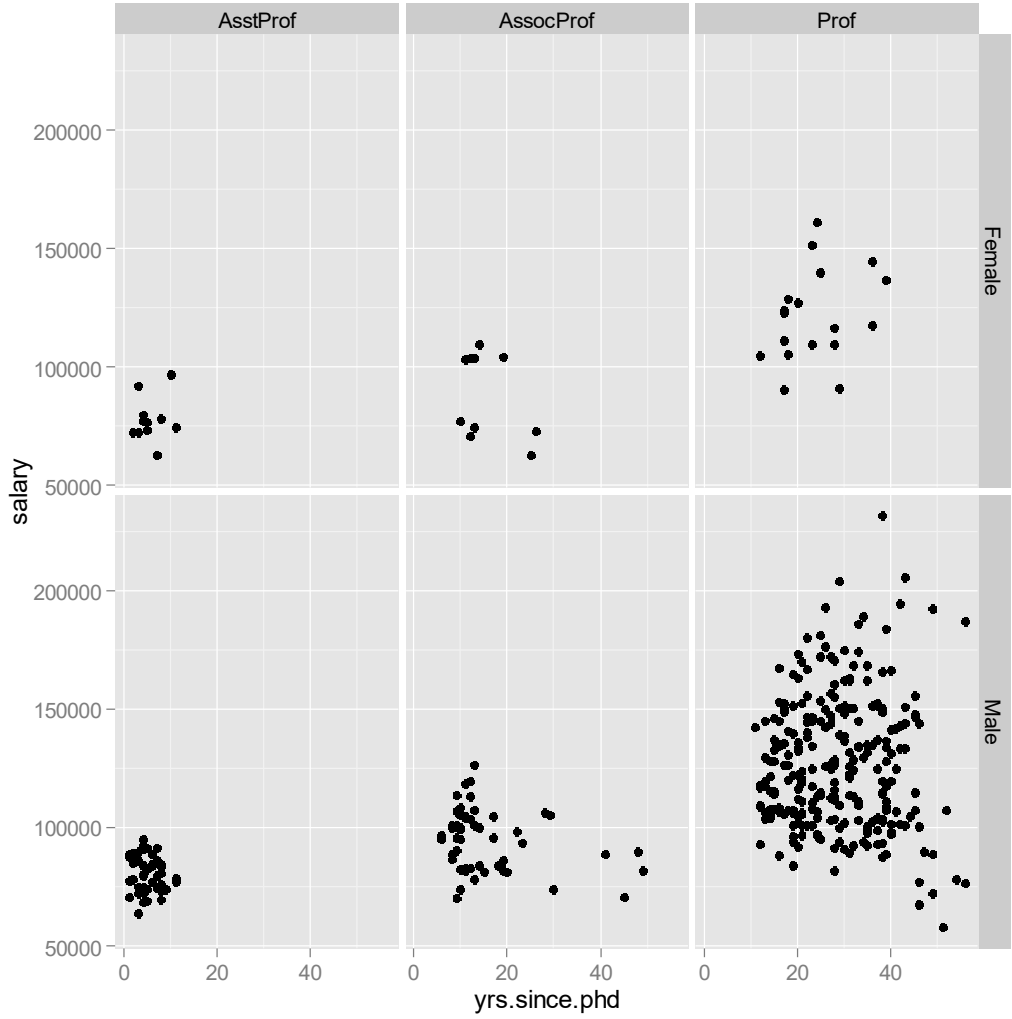
Facets

```
ggplot(data=Salaries,
  aes(x=yrs.since.phd,
      y=salary)) +
geom_point() +
facet_grid(. ~ sex)
```



Facets

```
ggplot(data=Salaries,
  aes(x=yrs.since.phd,
      y=salary)) +
geom_point() +
facet_grid(sex ~ rank)
```



Scales

```
scale_x_continuous()
scale_y_continuous()

scale_x_discrete()
scale_y_discrete()
```

Axes

```
scale_color_continuous()
scale_color_manual()
scale_color_brewer()
```

Colors

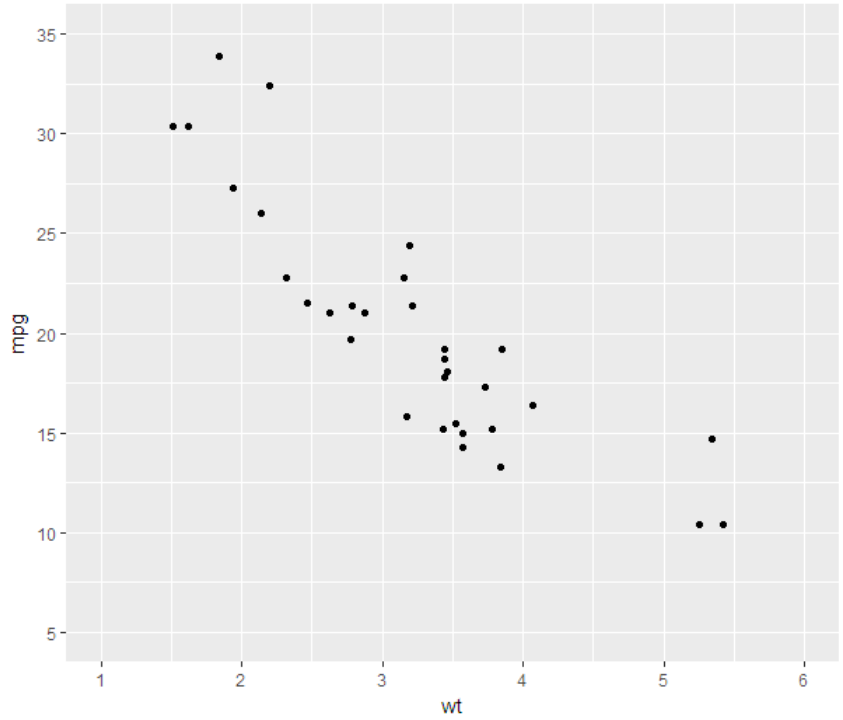
```
scale_fill_continuous()
scale_fill_manual()
```

Fill

Also shape, and size

Scales

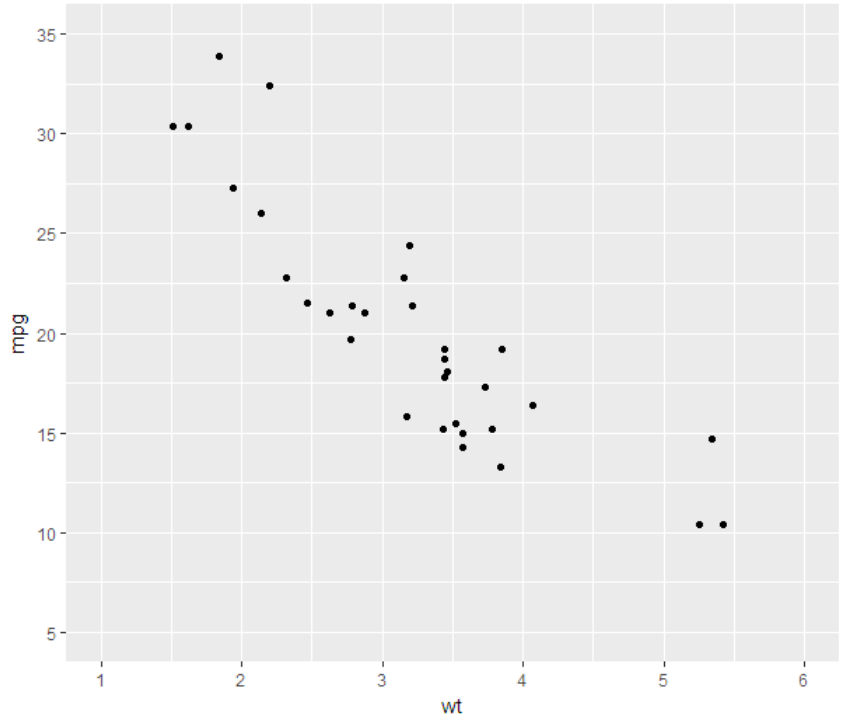
```
ggplot(mtcars, aes(x=wt, y=mpg)) + geom_point() +
  scale_x_continuous(breaks=seq(1,6,1), limits=c(1, 6)) +
  scale_y_continuous(breaks=seq(5, 35, 5),limits=c(5,35))
```



breaks,
limits,
labels

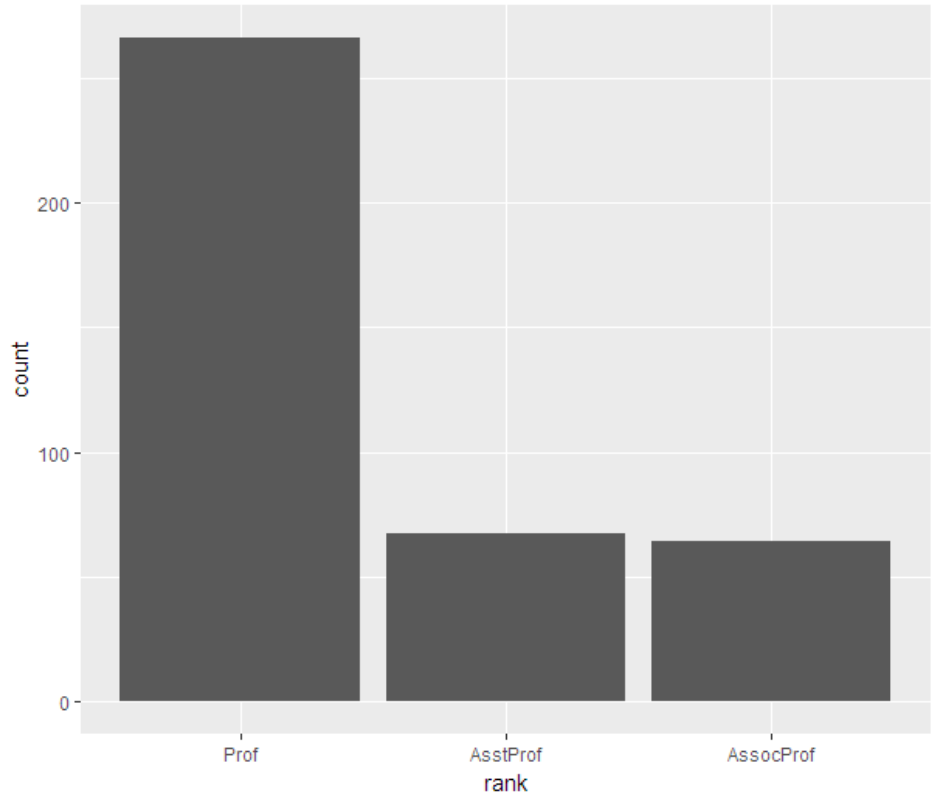
Scales

```
ggplot(mtcars, aes(x=wt, y=mpg)) + geom_point() +
  scale_x_continuous(breaks=seq(1,6,1), limits=c(1, 6)) +
  scale_y_continuous(breaks=seq(5, 35, 5),limits=c(5,35))
```



Scales

```
ggplot(Salaries, aes(x=rank)) + geom_bar() +
  scale_x_discrete(limits = c("Prof", "AsstProf", "AssocProf"))
```

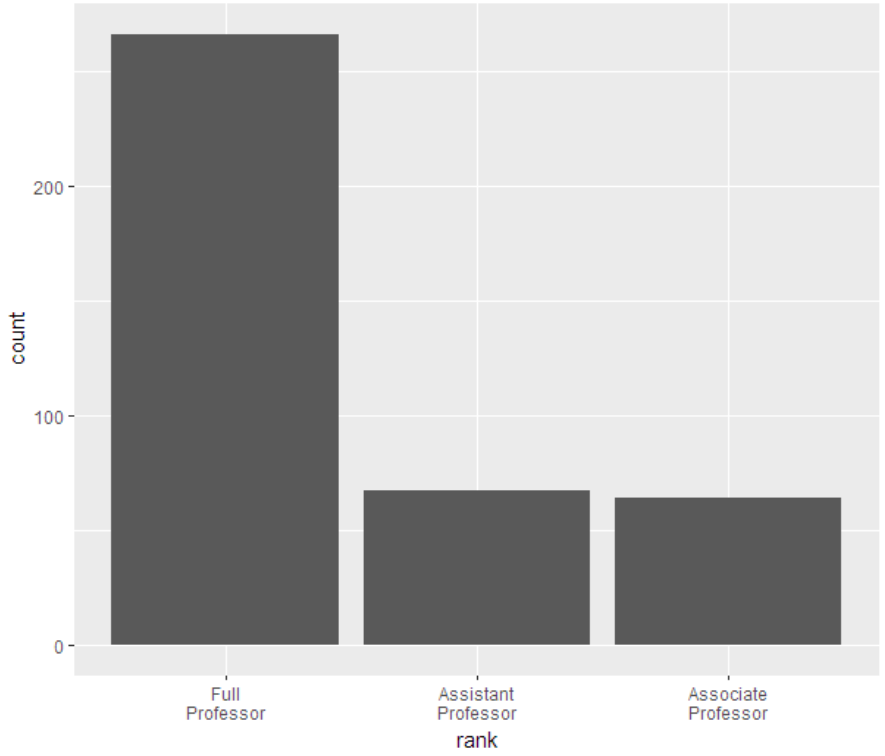


breaks, limits, labels

use limits to reorder levels

Scales

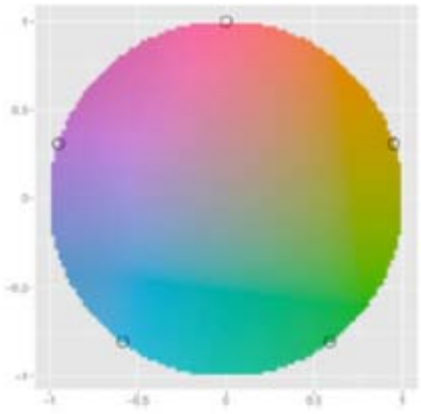
```
ggplot(Salaries, aes(x=rank)) + geom_bar() +
  scale_x_discrete(limits = c("Prof", "AsstProf", "AssocProf"),
      labels = c("Full\nProfessor", "Assistant\nProfessor" ,
                 "Associate\nProfessor"))
```



breaks,
limits,
labels

use limits
to reorder
levels

Scales – Color and Fill



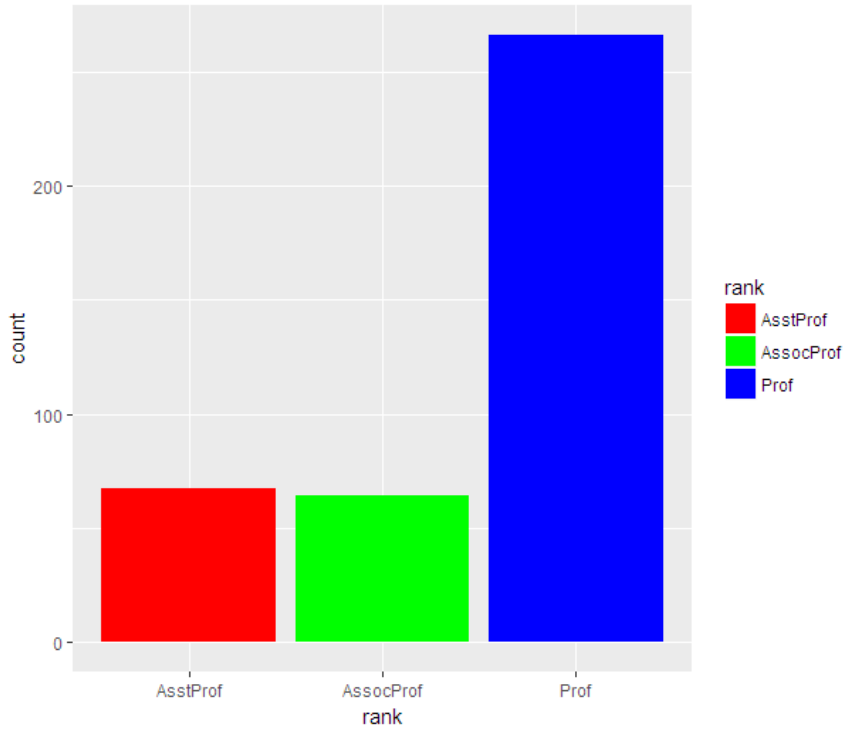
picking colors by name - "red"
or hex #ff0000

try colors() to list all built in colors

ggplot2 picks colors from around the circle
for example the 5 points above if there are five levels

Scales – Color/Fill

```
ggplot(Salaries, aes(x=rank, fill=rank)) + geom_bar() +
  scale_fill_manual(values=c("red", "green", "blue"))
```



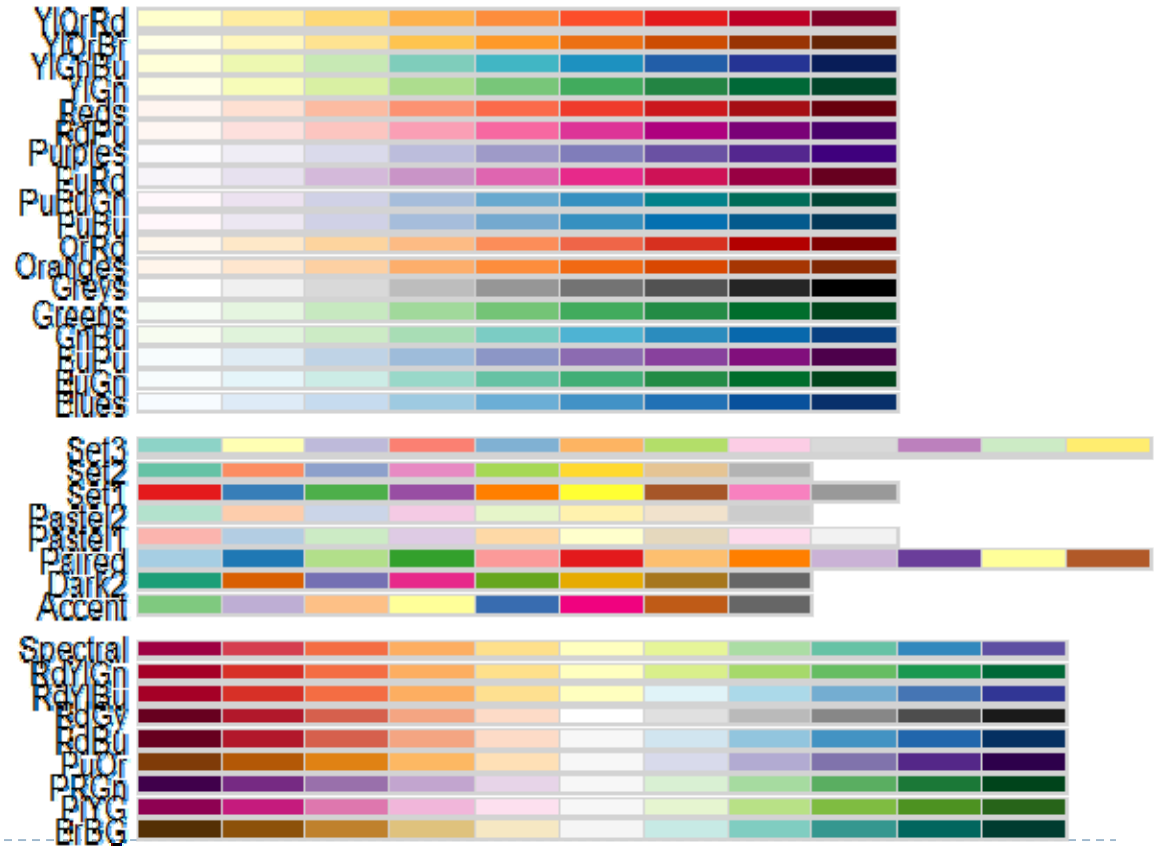
Scales – Color / Fill

Specify a color palate using

`scale_fill_brewer()`

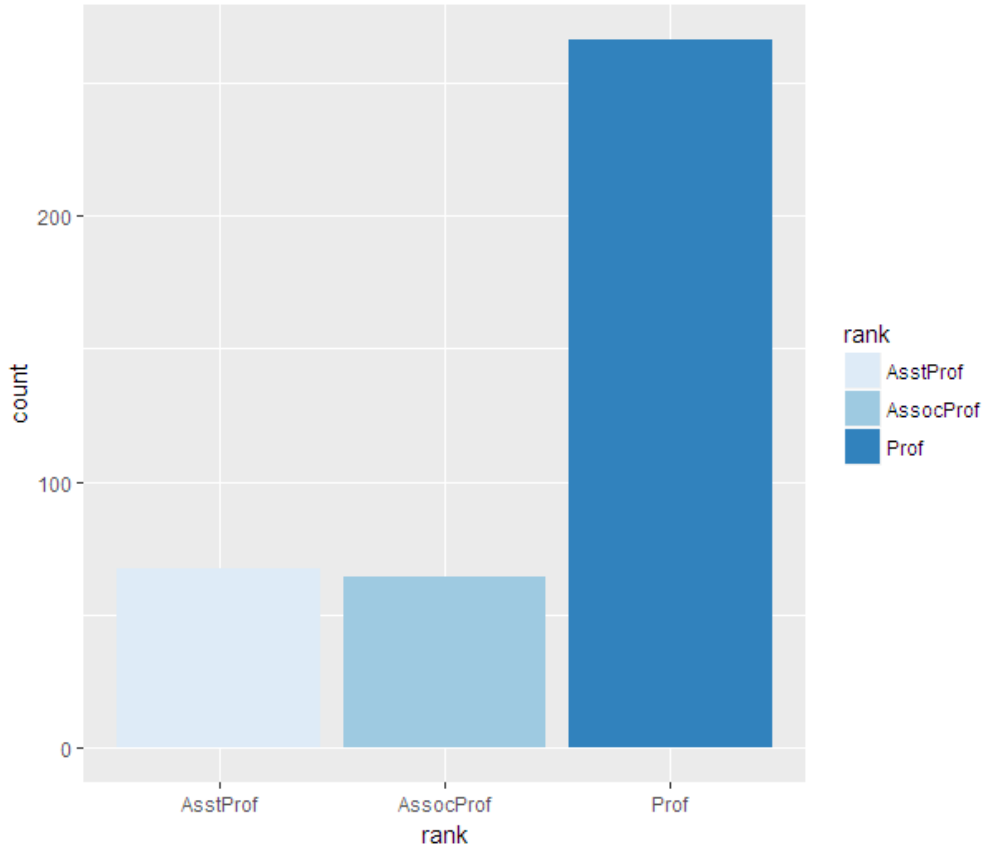
`scale_color_brewer()`

using `palette=` option



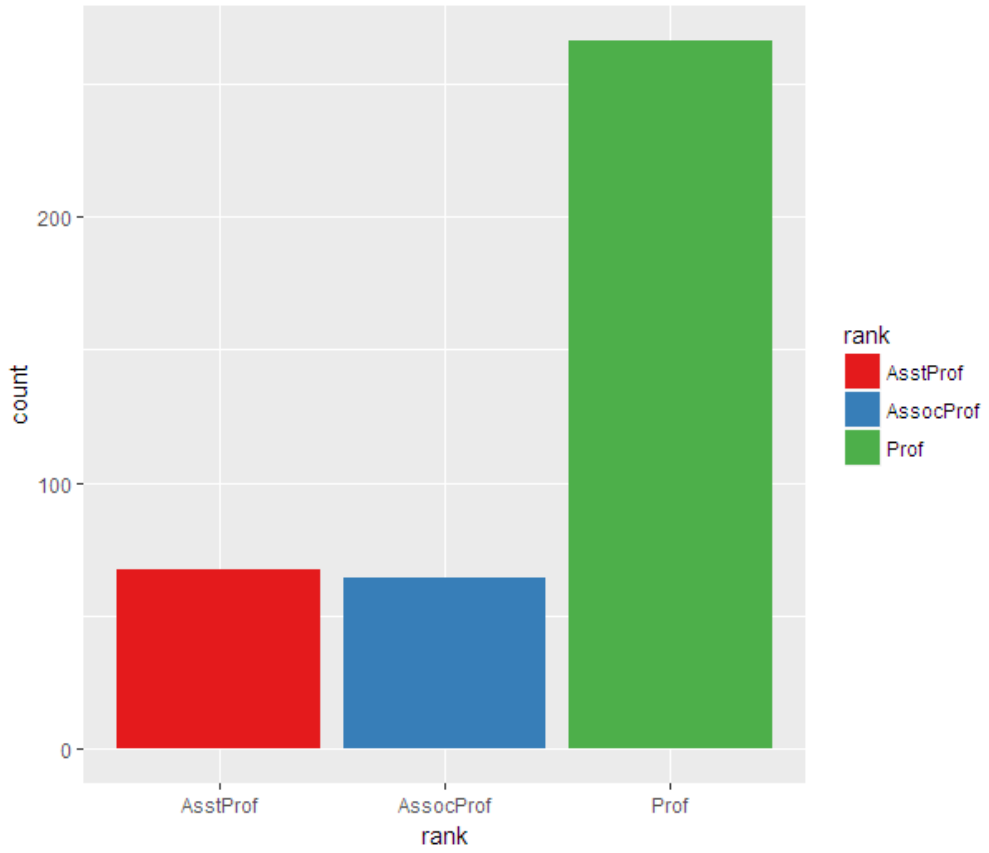
Scales – Color/Fill

```
ggplot(Salaries, aes(x=rank, fill=rank)) + geom_bar() +
  scale_fill_brewer()
```



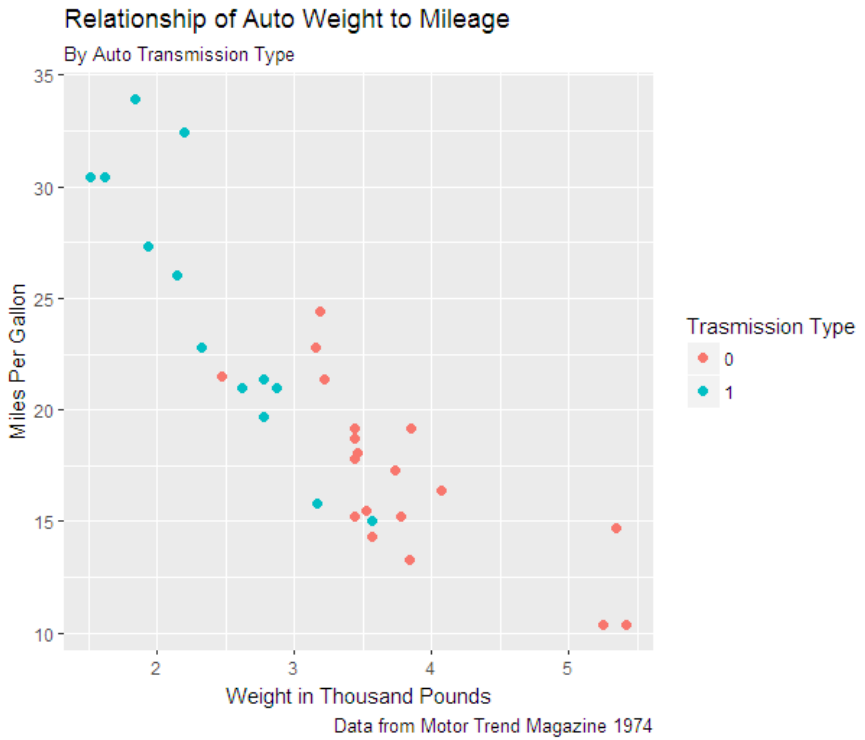
Scales – Color/Fill

```
ggplot(Salaries, aes(x=rank, fill=rank)) + geom_bar() +
  scale_fill_brewer(palette = "Set1")
```



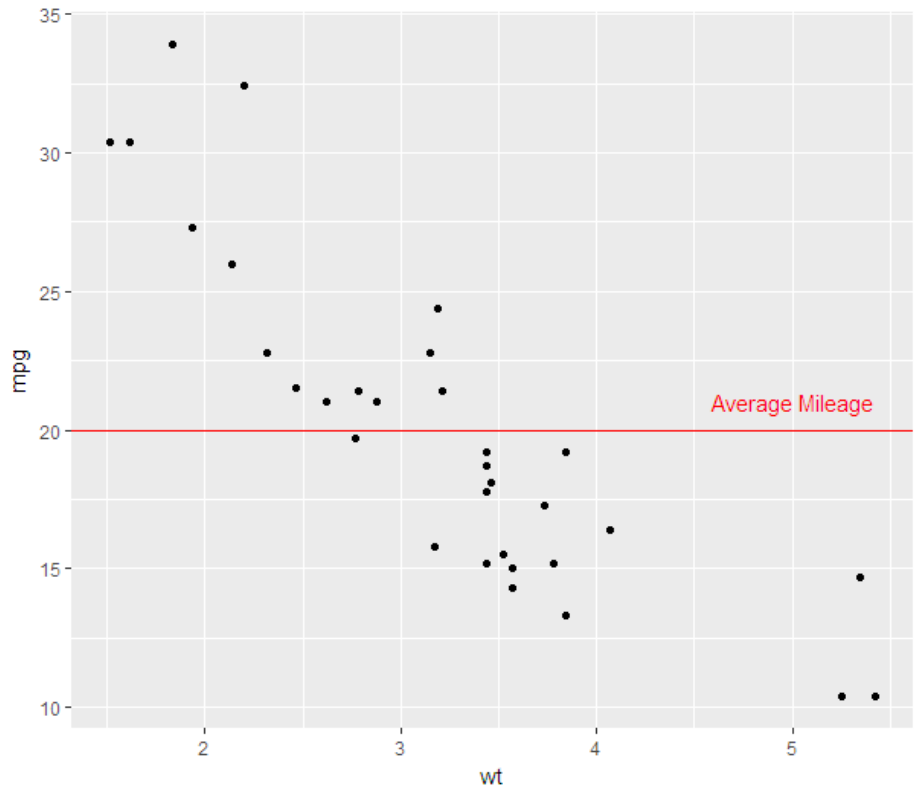
Annotations - Labels

```
p <- ggplot(data=mtcars, aes(x=wt, y=mpg, color=factor(am))) +
     geom_point(size=2) +
     labs(title="Relationship of Auto Weight to Mileage",
       subtitle="By Auto Transmission Type",
       caption = "Data from Motor Trend Magazine 1974",
       x = "Weight in Thousand Pounds",
       y="Miles Per Gallon",
       color = "Transmission Type")
p
```



Annotations – reference lines and labels

```
ggplot(data=mtcars, aes(x=wt, y=mpg)) + geom_point() +
  geom_hline(yintercept=20, color="red") +
  annotate("text", x=5, y=21, label="Average Mileage", color="red")
```

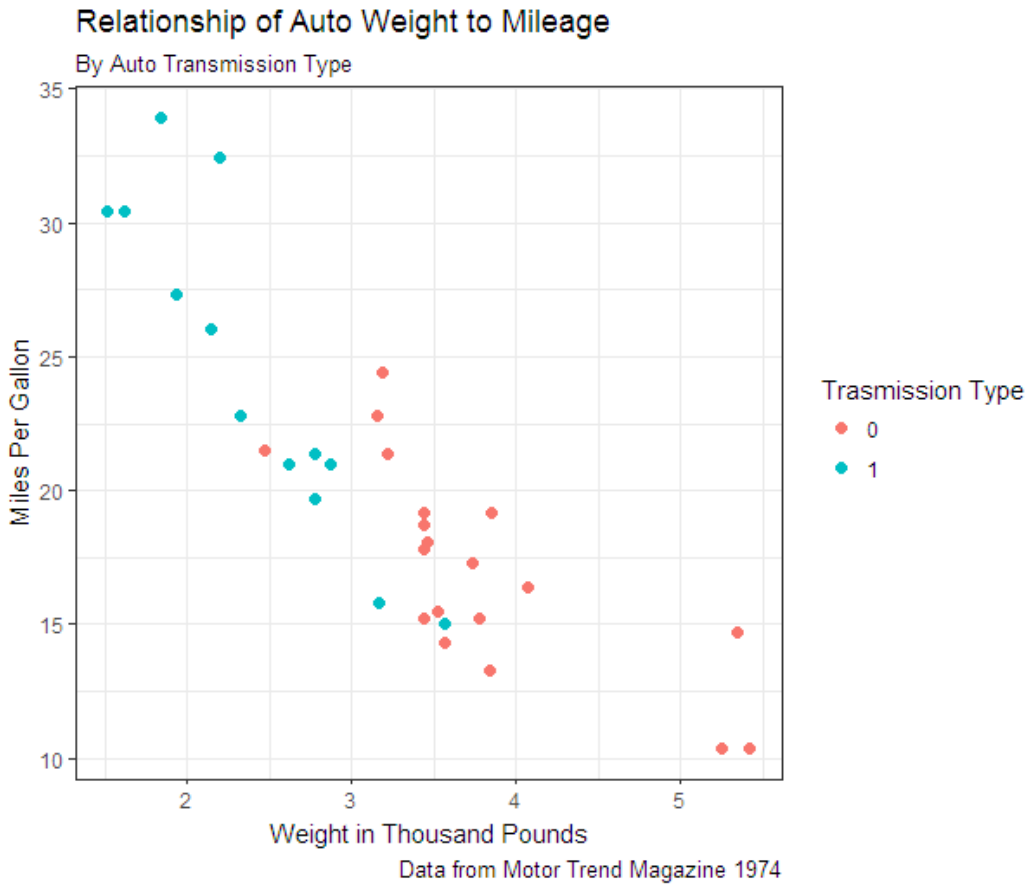


Themes

- See handout: Modifying ggplot2 themes

Themes - prepackaged

```
p + theme_bw()
```



Themes - prepackaged

```
library(ggthemes)
```

theme_base: a theme resembling the default base graphics in R. See also theme_par.

theme_calc: a theme based on LibreOffice Calc.

theme_economist: a theme based on the plots in the The Economist magazine.

theme_excel: a theme replicating the classic ugly gray charts in Excel

theme_few: theme from Stephen Few's "Practical Rules for Using Color in Charts".

theme_fivethirtyeight: a theme based on the plots at fivethirtyeight.com.

theme_gdocs: a theme based on Google Docs.

theme_hc: a theme based on Highcharts JS.

theme_par: a theme that uses the current values of the base graphics parameters in par.

theme_pander: a theme to use with the pander package.

theme_solarized: a theme using the solarized color palette.

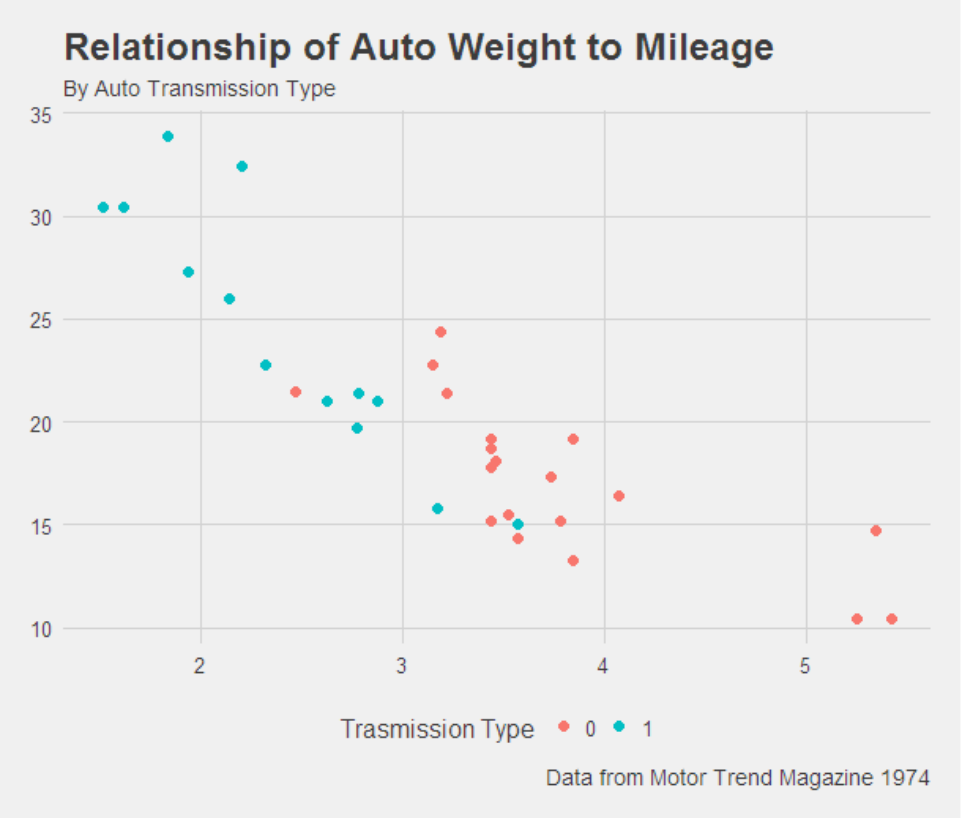
theme_stata: themes based on Stata graph schemes.

theme_tufte: a minimal ink theme based on Tufte's The Visual Display of Quantitative Information.

theme_wsj: a theme based on the plots in the The Wall Street Journal.

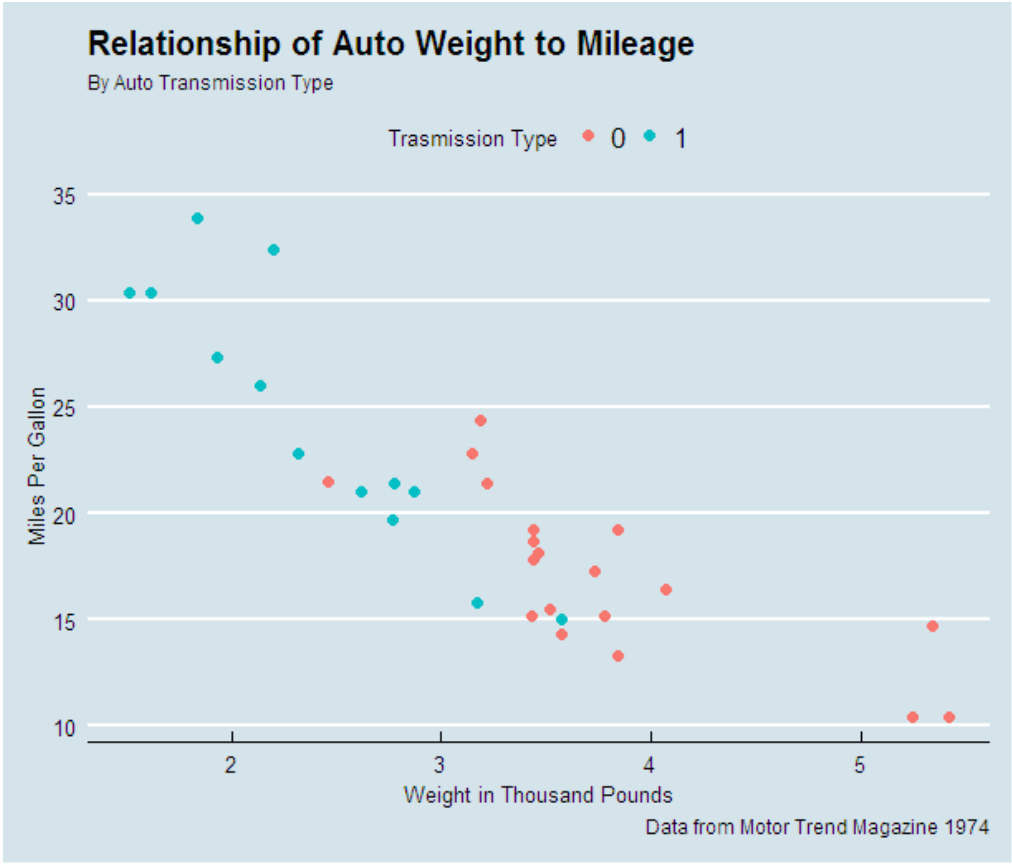
Themes - prepackaged

```
library(ggthemes)
p + theme_fivethirtyeight()
```



Themes - prepackaged

```
library(ggthemes)
p + theme_economist()
```



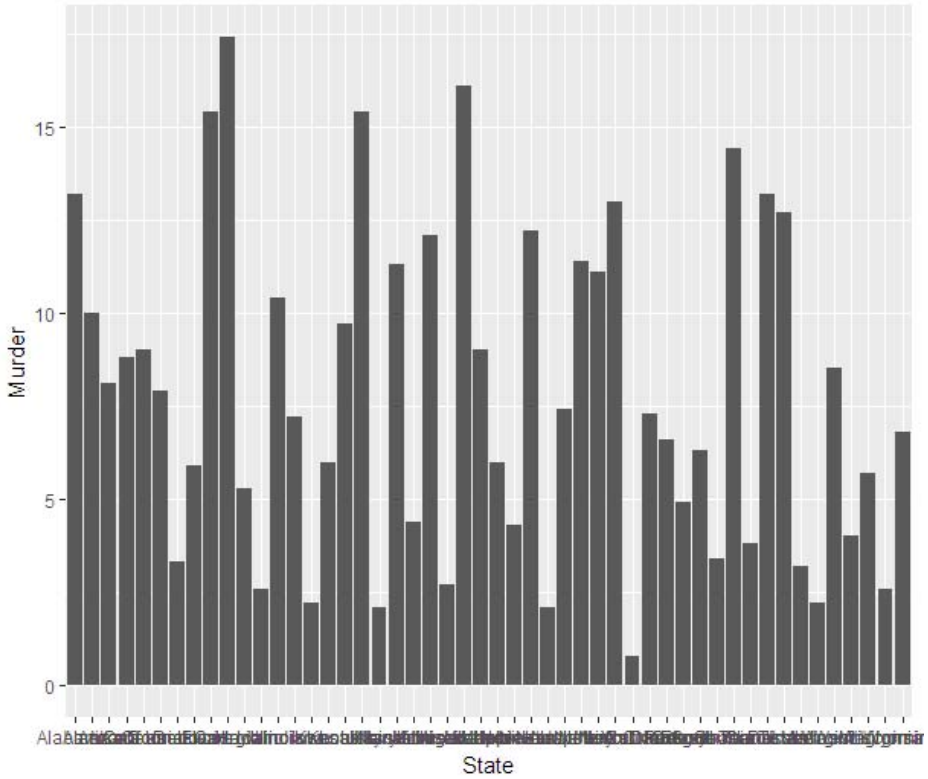
Saving your work

- `ggsave(filename="filename.ext", plot=)`
  - ext can be

    `eps, ps, tex, pdf, jpeg, tiff, png, bmp, svg, wmf`
  - plot defaults to last one created
  - wmf on windows platforms only
  - svg can be edited using Inkscape

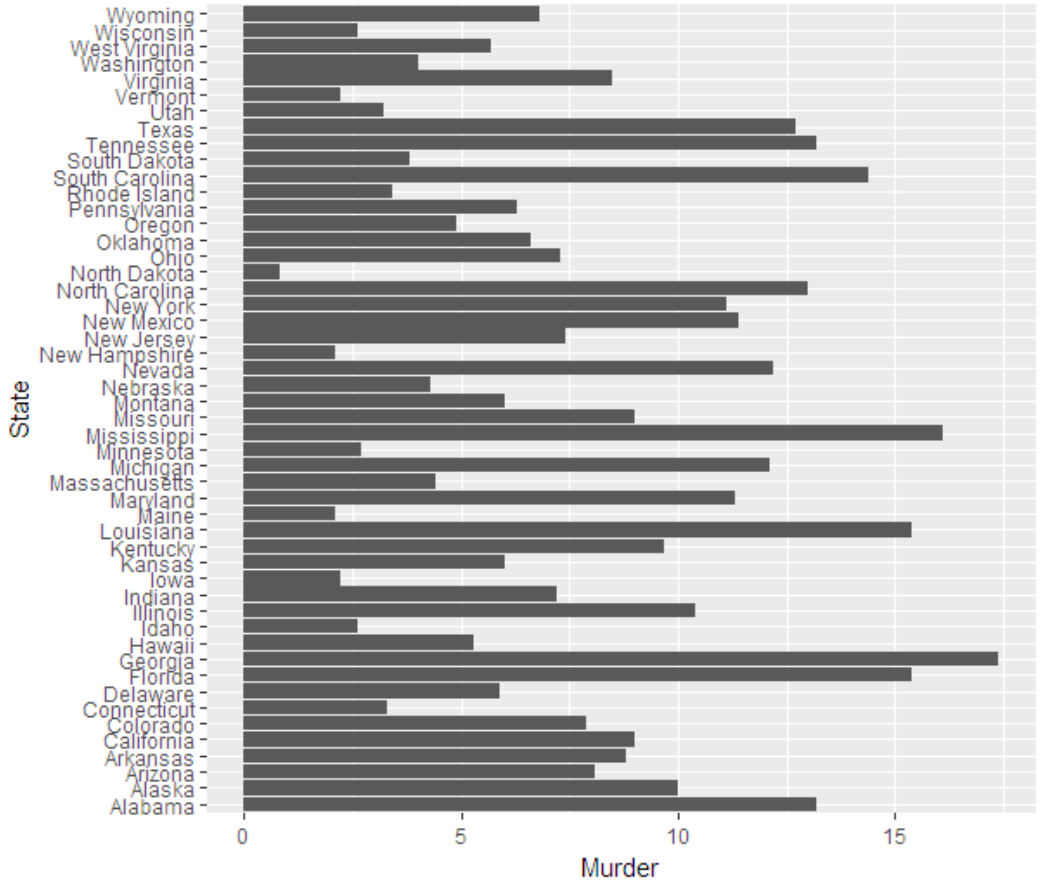
A final useful function

```
USArrests$State <- row.names(USArrests)
ggplot(data=USArrests, aes(x=State, y=Murder)) +
       geom_bar(stat="identity")
```



A final useful function

```
USArrests$State <- row.names(USArrests)
ggplot(data=USArrests, aes(x=State, y=Murder)) +
        geom_bar(stat="identity") + coord_flip()
```



Learning more

- Hadley Wickham – http://docs.ggplot2.org/
- Winston Chang- http://wiki.stdout.org/rcookbook/Graphs/