Lecture 20 - Plotting continuous data with ggplot2

Most of today will be spent working through examples of a variety of plots for continuous data that can be generated using R and the ggplot2 package. We'll work through the provided code and talk about what is happening and why it works. Make sure to ask any questions you may have.

```
# only ever install packages one time
install.packages("ggplot2")
install.packages("cowplot")
```

```
# load a package every time you wish to use it
library(ggplot2)
library(cowplot)
```

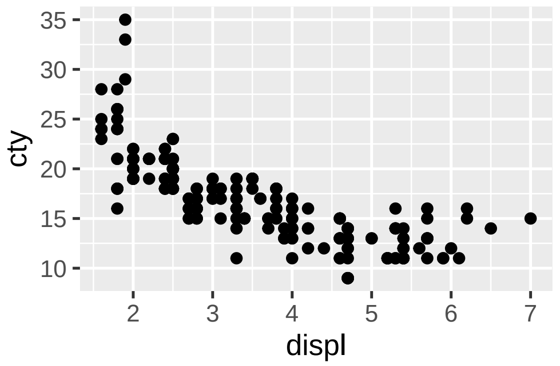
```
#read in data
mpg = read.table("mpg.txt", header=TRUE, sep="\t", stringsAsFactors=FALSE)

dim(mpg)
```

```
## [1] 234   9
```

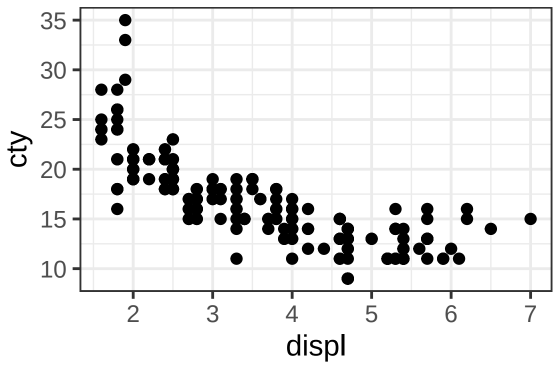
```
head(mpg)
```

```
##   manufacturer model displ year cyl      trans drv cty hwy
## 1         audi    a4   1.8 1999   4   auto(l5)   f  18  29
## 2         audi    a4   1.8 1999   4 manual(m5)   f  21  29
## 3         audi    a4   2.0 2008   4 manual(m6)   f  20  31
## 4         audi    a4   2.0 2008   4   auto(av)   f  21  30
## 5         audi    a4   2.8 1999   6   auto(l5)   f  16  26
## 6         audi    a4   2.8 1999   6 manual(m5)   f  18  26
```

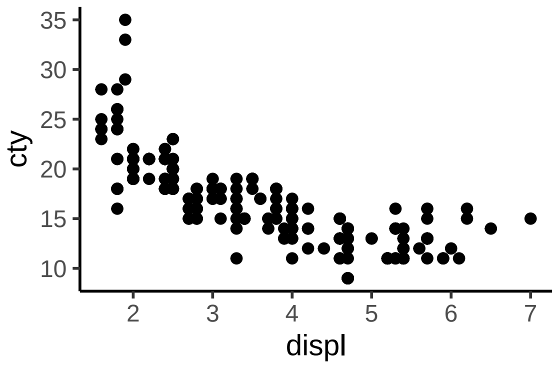
```
# plot of displacement (engine size) vs. city miles per gallon (cty)
ggplot(data = mpg,
       aes(x = displ, y = cty)) +
  geom_point()
```



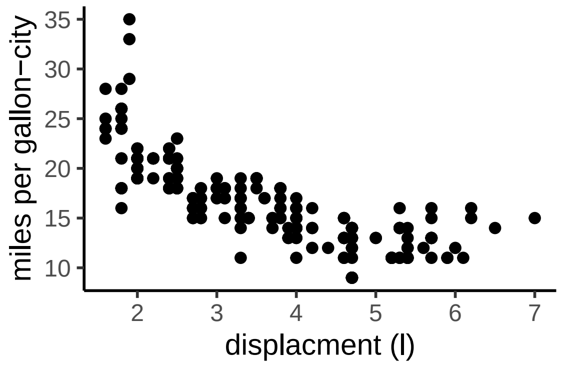
```
# remove grey background
ggplot(data = mpg,
       aes(x = displ, y = cty)) +
  geom_point() +
  theme_bw()
```



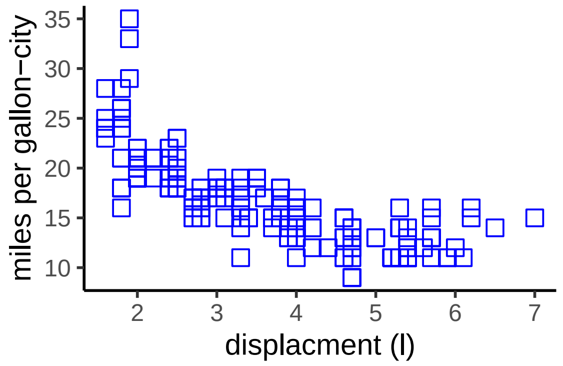
```
# remove grey background and gridlines
ggplot(data = mpg,
       aes(x = displ, y = cty)) +
  geom_point() +
  theme_classic()
```



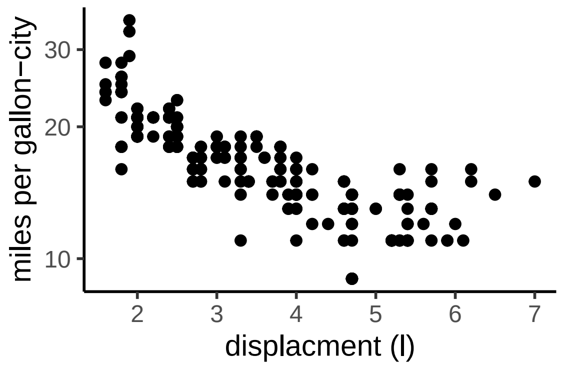
```
# change the x and y labels; cartesian coordinates are default
ggplot(data = mpg,
       aes(x = displ, y = cty)) +
  geom_point() +
  xlab("displacment (l)") +
  ylab("miles per gallon-city") +
  theme_classic()
```



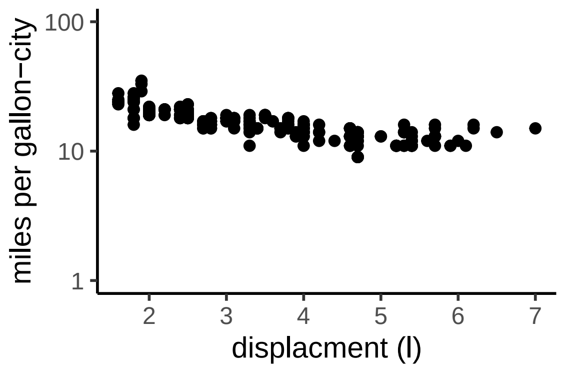
```
# arguments to geom_point() can alter the appearance of the points
ggplot(data = mpg,
       aes(x = displ, y = cty)) +
  geom_point(color = "blue", shape = 22, size = 3) +
  xlab("displacment (l)") +
  ylab("miles per gallon-city") +
  theme_classic()
```



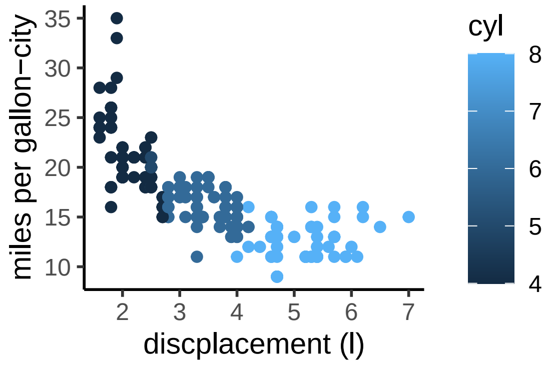
```
# log transform the y-axis
ggplot(data = mpg,
       aes(x = displ, y = cty)) +
  geom_point() +
  xlab("displacment (l)") +
  ylab("miles per gallon-city") +
  scale_y_log10() +
  theme_classic()
```



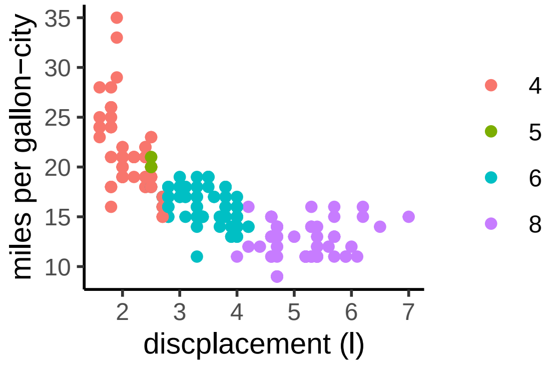
```
# arguments to scale can also customize the range and tick locations
ggplot(data = mpg,
       aes(x = displ, y = cty)) +
  geom_point() +
  theme_classic() +
  xlab("displacment (l)") +
  ylab("miles per gallon-city") +
  scale_y_log10(limits=c(1,100),
                breaks=c(1,10,100))
```



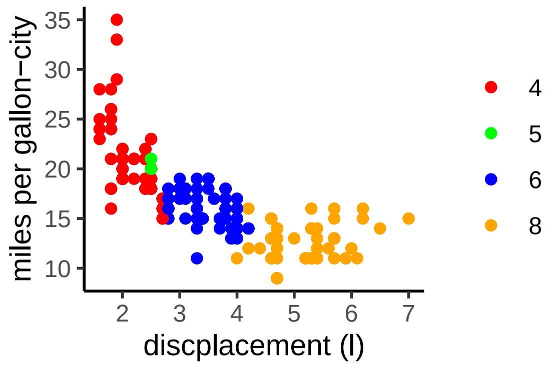
```
# we can also color code points based on continuous or categorical variables
# continuous
ggplot(data = mpg,
       aes(x = displ, y = cty, color = cyl)) +
  geom_point() +
  xlab("discplacement (l)") +
  ylab("miles per gallon-city") +
  theme_classic()
```



```
# categorical
ggplot(data = mpg,
       aes(x = displ, y = cty, color = as.factor(cyl))) +
  geom_point() +
  xlab("discplacement (l)") +
  ylab("miles per gallon-city") +
  theme_classic() +
  theme(legend.title=element_blank())
```

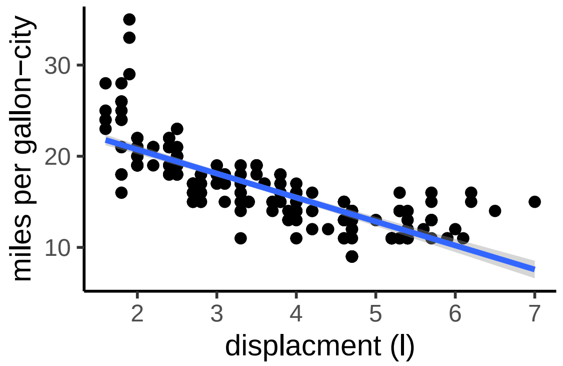


```r
# categorical - change display colors
ggplot(data = mpg,
       aes(x = displ, y = cty, color = as.factor(cyl))) +
  geom_point() +
  xlab("discplacement (l)") +
  ylab("miles per gallon-city") +
  scale_color_manual(values = c('red','green','blue','orange')) +
  theme_classic() +
  theme(legend.title=element_blank())
```



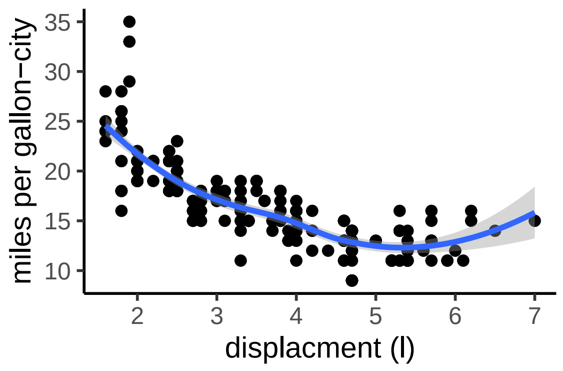
```r
# add a linear trendline with a new layer
ggplot(data = mpg,
       aes(x = displ, y = cty)) +
  geom_point() +
  xlab("displacment (l)") +
  ylab("miles per gallon-city") +
  stat_smooth(method="lm") +
  theme_classic()
```

```
## `geom_smooth()` using formula 'y ~ x'
```



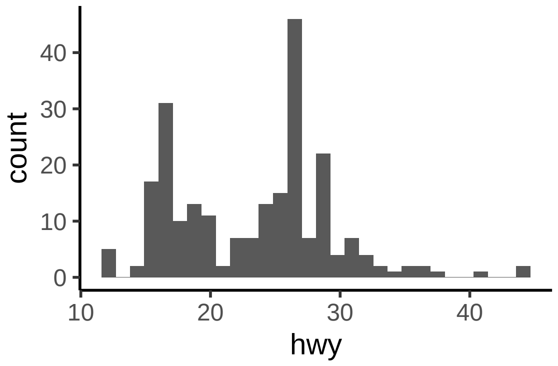
```r
# add a spline with a new layer
ggplot(data = mpg,
       aes(x = displ, y = cty)) +
  geom_point() +
  xlab("displacment (l)") +
  ylab("miles per gallon-city") +
  stat_smooth(method="loess") +
  theme_classic()
```

```
## `geom_smooth()` using formula 'y ~ x'
```

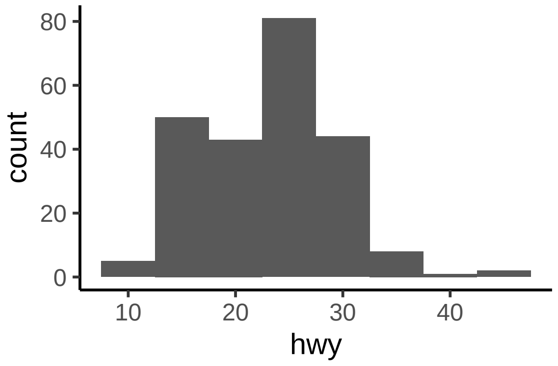


```r
# histogram of mpg hwy
ggplot(data = mpg, aes(x = hwy)) +
  geom_histogram() +
  theme_classic()
```

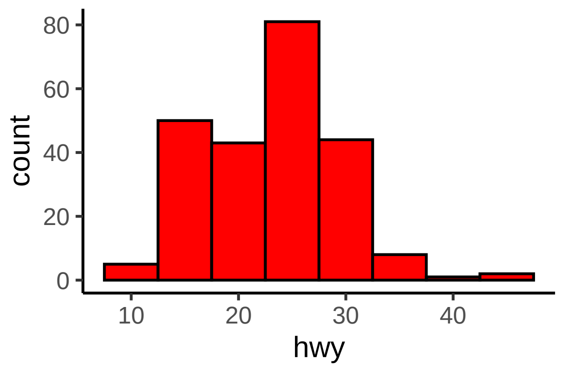
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



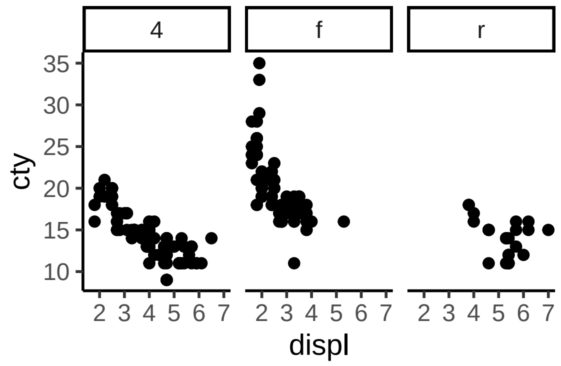
```r
# change number of bins
ggplot(data = mpg, aes(x = hwy)) +
  geom_histogram(binwidth = 5) +
  theme_classic()
```



```r
# change number of bins
# specify color in geom_histogram
ggplot(data = mpg, aes(x = hwy)) +
  geom_histogram(binwidth = 5, fill = "red", color = "black") +
  theme_classic()
```



```
# faceting allows the same plot across different categories to be generated
ggplot(data = mpg, aes(x = displ, y = cty)) +
  geom_point() +
  facet_wrap(~drv) +
  theme_classic()
```



```
# cowplot allows for multiple panels with different plot types
# makes sure you install the cowplot package with install.packages("cowplot")
# and load the package with library(cowplot)

#store the plots as variables
plot1 <- ggplot(data = mpg,
       aes(x = displ, y = cty, color = as.factor(cyl))) +
  geom_point() +
  xlab("discplacement (l)") +
  ylab("miles per gallon-city") +
  scale_color_manual(values = c('red','green','blue','orange')) +
  theme_classic() +
  theme(legend.title=element_blank())

plot2 <- ggplot(data = mpg, aes(x = hwy)) +
  geom_histogram(binwidth = 5, fill = "red", color = "black") +
  theme_classic()

#put the subplots together in a variable called "fig1"
fig1 <- plot_grid(plot1, plot2,
          labels = c("a", "b"),
          rel_widths = c(1, 0.85),
          ncol = 2,
          nrow = 1)
```

```r
#save your figure to an external file
ggsave(filename = "Fig1.pdf",
      plot = fig1,
      width = 8,
      height = 5,
      dpi = 300)
```

Challenge

Practice using the syntax demonstrated above by writing a script to generate the following plots using the mpg data.

1. A scatter plot of miles per gallon city versus miles per gallon highway. Color code the points by 'drv' (four-wheel drive vs. front-wheel drive vs. rear-wheel drive). Add a linear trendline to the plot.

2. A "density plot" of engine displacement.

3. A facetted series of scatter plots across different manufacturers. Each plot should show engine size (displacement, displ) vs. highway miles. Make each manufacturer a different color.