Tutorial 12 - Reviewing ggplot

Reviewing continuous and categorical plots

The first part of tutorial this week will be a self-paced review of generating a few plots with the ggplot package in R. After students work through the code, instructors will code the examples and answer any questions students might have.

ggplot overview

Making plots with ggplot in R involves the following steps and ggplot functions:

- linking data, in a dataframe, to a ggplot object; `ggplot()`
- creating one or more layers that display the data; `geom` and `stat` functions
- specifying the display of axes, legends, etc.; `scale` and `label` functions
- specifying finer points of the plot appearance (font size, background color, etc.); `theme` functions

Each of these components do not need to be specified each time you make a plot with ggplot because defaults exist for most of these components. However, each time you make a plot you must create a ggplot object, including data to be used with `ggplot()` and specify a `geom` or `stat`.

```
# load a package every time you wish to use it
# this gives you access to the functions provided by that package, which
#     are not available in the base functionality of R
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.2
```

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

dim(mpg)
```

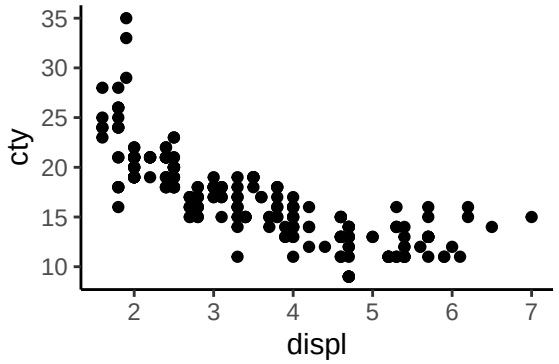
```
## [1] 234   9
```

```
head(mpg)
```

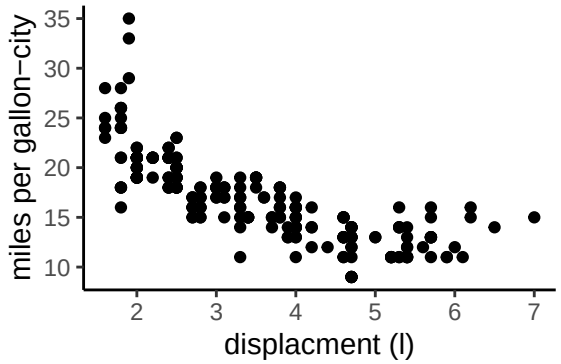
```
##   manufacturer model displ year cyl      trans drv cty hwy
## 1         audi    a4   1.8 1999   4   auto(l5)   f  18  29
## 2         audi    a4   1.8 1999   4 manual(m5)   f  21  29
## 3         audi    a4   2.0 2008   4 manual(m6)   f  20  31
## 4         audi    a4   2.0 2008   4   auto(av)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

```
#### scatter plot of two continuous variables
##   displacement (engine size) vs. city miles per gallon (cty)
#   -the ggplot call creates a ggplot object, including
#        defining the dataframe containing our data
#    within the ggplot call, aes() specifies x and y components of our figure
#   -geom_point() specifies the particular appearance of the data in the plot
#   -the absence of any scale or label functions means we are using the defaults
#   -theme_classic() specifies a classic appearance for background and fonts

ggplot(data = mpg,aes(x = displ, y = cty)) +
  geom_point() +
  theme_classic()
```

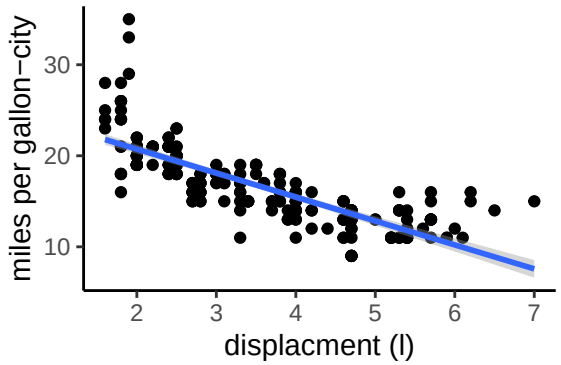



```
# we can add function calls that change labels to use something other than the default
ggplot(data = mpg,aes(x = displ, y = cty)) +
  geom_point() +
  xlab("displacment (l)") +
  ylab("miles per gallon-city") +
  theme_classic()
```



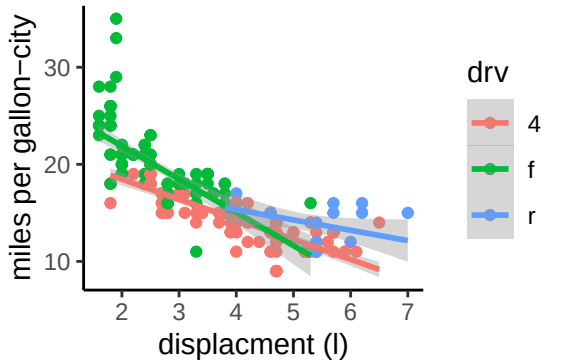

```
# we can add more than one layer to a plot, including things like trend lines
ggplot(data = mpg,aes(x = displ, y = cty)) +
  geom_point() +
  stat_smooth(method="lm") +
  xlab("displacment (l)") +
  ylab("miles per gallon-city") +
  theme_classic()
```

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



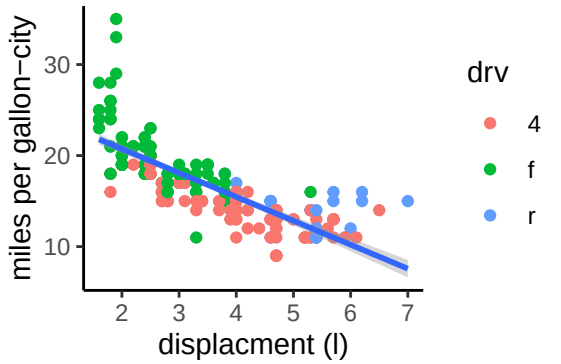

```
# we can alter coloring of the points too.
# here are two ways of doing that
ggplot(data = mpg,aes(x = displ, y = cty, color=drv)) +
  geom_point() +
  stat_smooth(method="lm") +
  xlab("displacment (l)") +
  ylab("miles per gallon-city") +
  theme_classic()
```

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




```
ggplot(data = mpg,aes(x = displ, y = cty)) +
  geom_point(aes(color=drv)) +
  stat_smooth(method="lm") +
  xlab("displacment (l)") +
  ylab("miles per gallon-city") +
  theme_classic()
```

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

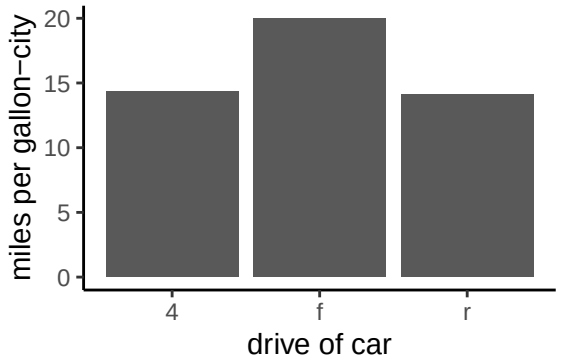


Question: Were there any differences in the figures generated by the two methods for changing point colors? What in the code generated those differences?

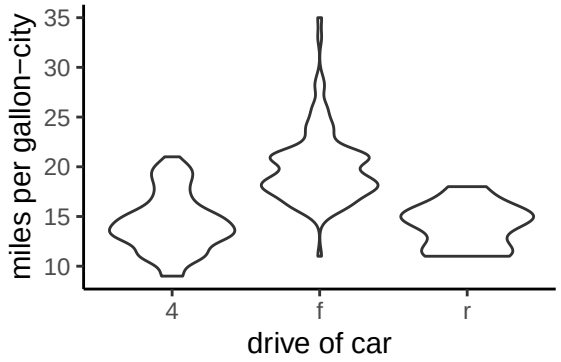
```
#### plots for a continuous and categorical variable
##   city miles per gallon (cty) as a function of drive

# a bar chart often shows counts or averages of values across categories
# stat_summary() can be used to simultaneously calculate summary stats
#   and plot these summary stats

# plot of average city mpg for each drive
ggplot(mpg, aes(x = drv, y = cty)) +
  stat_summary(fun = mean, geom = "bar") +
  xlab("drive of car") +
  ylab("miles per gallon-city") +
  theme_classic()
```




```
# a violin plot shows a distribution of the data, rather than only the mean, for a continuous variable
ggplot(mpg, aes(x=drv, y=cty)) +
  geom_violin() +
  xlab("drive of car") +
  ylab("miles per gallon-city") +
  theme_classic()
```



Question: What are the pros and cons of these two categorical plot types?

Question: Think back to the figures you submitted for your quiz on Monday, what are characteristics of a good plot?

Challenges

1. Write a script that uses a for loop to generate violin plots, using the species name as categories, of each measurement in the iris dataset.

2. Using the mpg dataset, generate a bar plot of mean engine sizes (measured as displacement, displ) for engines with different numbers of cylinders (cyl). Include a standard error estimate around each mean.

3. Using the iris dataset, generate a density plot with a density line for sepal width of each Iris species.