Visual Summary

Contents

Visualizing Data

Early in the analysis pipeline, visualizing data can aid in forming an understanding of the data, identifying trends, and spotting anomalies. Once the analysis has been completed, a good visualization can get attention, keep interest and effectively convey analysis insights to stakeholders. Our focus will be on visualizing data early in the analysis process. Accordingly, we will focus on a wide variety of charts but not worry much about presentation.

ggplot2

There are three general visualization packages in R: graphics, lattice, and ggplot2. `graphics` is the oldest and since it usually loads automatically is also known as Base Graphics. `lattice` makes better looking charts with less code, through the use of good default arguments. `ggplot2` attempts to integrate the benefits of `graphics` and `lattice` with a layered approach to generating plots. `ggplot2` implements the grammar of graphics, a coherent system for describing and building graphs (Grolemund and Wickham 2020).

The layered approach of `ggplot2`, inspired by the notion of a grammar of graphics Wilkinson et al (2005), is described by Hadley Wickham, the primary author of ggplot2, in this article.

More on ggplot2

- . Cheatsheet
- . ggplot2 functions
- . ggplot2: Elegant Graphics for Data Analysis by Hadley Wickham
- . Graphical Data Analysis with R by Antony Unwin
- . R Graphics Cookbook, a bit dated, but still useful

Grammar of Graphics

Each chart built with ggplot2 must include the following

- Data
- Aesthetic mapping (aes)
  - Describes how variables are mapped onto graphical attributes
  - Visual attribute of data including x-y axes, color, fill, shape, and alpha
- Geometric objects (geom)
  - Determines how values are rendered graphically, as bars (geom_bar), scatterplot (geom_point), line (geom_line), etc.

Thus, the template for graphic in ggplot2 is:

```
ggplot(data = <Enter Data Here>, mapping = aes(<Enter Aesthetic(s) here>))+
  <Enter geom function here>
```

Orientation

The toolkit of R in general and ggplot2 in particular are a means to an end. The sections that follow are specifically organized around ways to explore data. If you are interested in the vast visualization capabilities of ggplot2, see chapter on Data Visualization in R for Data Science.

Next, we will look at charts to examine distribution of a variable, examine variance, and spot outliers. After that, we will look at charts to compare groups. Next, we will examine charts for exploring relationships between variables. We will finish up by looking at ways to expand our view by examining multiple variables and multiple charts.

Setup

To illustrate visual summaries using ggplot2, we will make use of the mpg dataset that comes with ggplot2. To get started, install ggplot2, call the library and explore the dataset.

```
# install.packages('ggplot2')  # run this line if you have not installed ggplot2
library(ggplot2)
str(mpg)
```

```
## tibble [234 x 11] (S3: tbl_df/tbl/data.frame)
##  $ manufacturer: chr [1:234] "audi" "audi" "audi" "audi" ...
##  $ model       : chr [1:234] "a4" "a4" "a4" "a4" ...
##  $ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int [1:234] 1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl         : int [1:234] 4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : chr [1:234] "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
##  $ drv         : chr [1:234] "f" "f" "f" "f" ...
##  $ cty         : int [1:234] 18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy         : int [1:234] 29 29 31 30 26 26 27 26 25 28 ...
##  $ fl          : chr [1:234] "p" "p" "p" "p" ...
##  $ class       : chr [1:234] "compact" "compact" "compact" "compact" ...
```

Examine Distribution

In this section, we will look at charts to examine the distribution of a variable, examine variance, and spot outliers.

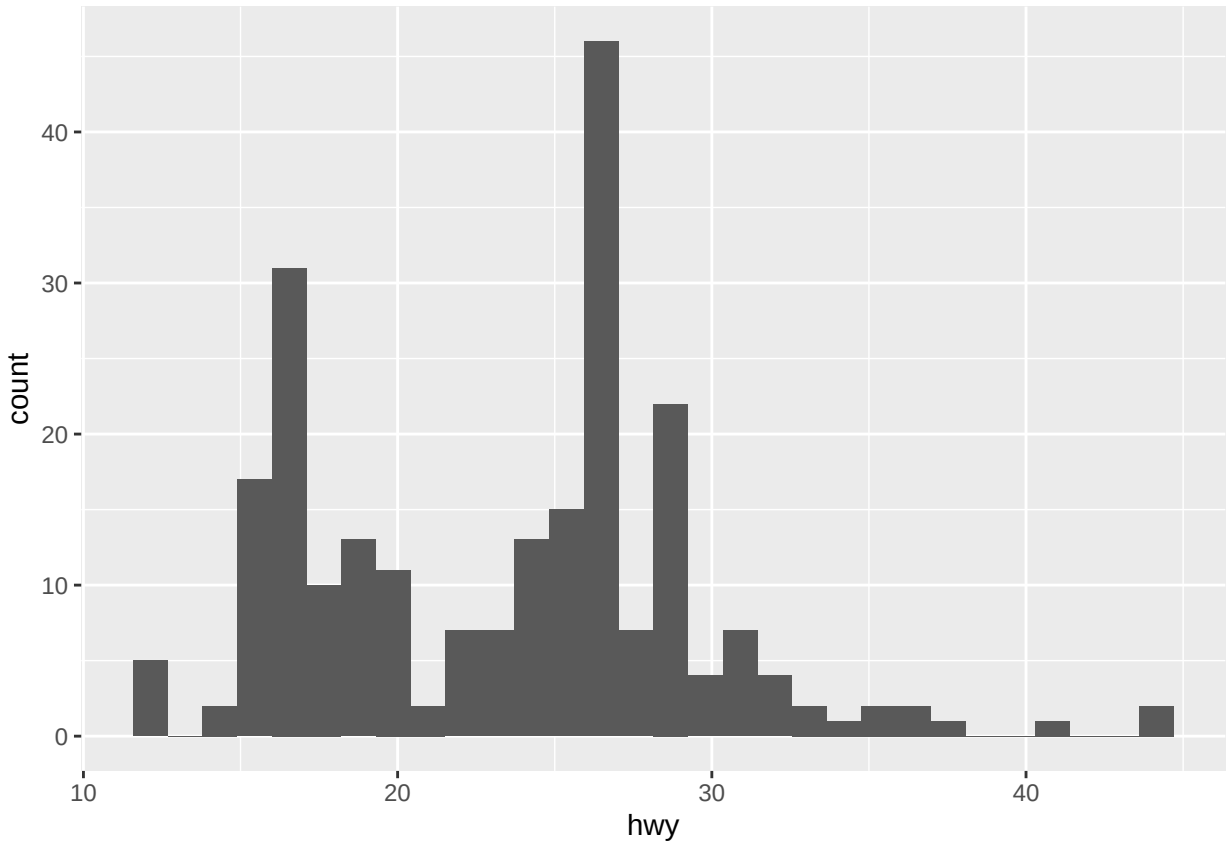
Histogram

Histogram is the most common chart for exploring the distribution of a numerical variable. It is generated by binning the variable into discrete categories and constructing a bar chart of counts of various categories. Shape of the distribution may reveal, (a) a distribution that is symmetric or skewed, (b) distribution that is flat or peaked, or (c) presence of outliers.

Let us examine the distribution of highway gas mileage (hwy). As with every ggplot2 chart, we are going to need to specify

- data: mpg
- aes: x = hwy
- geom: histogram

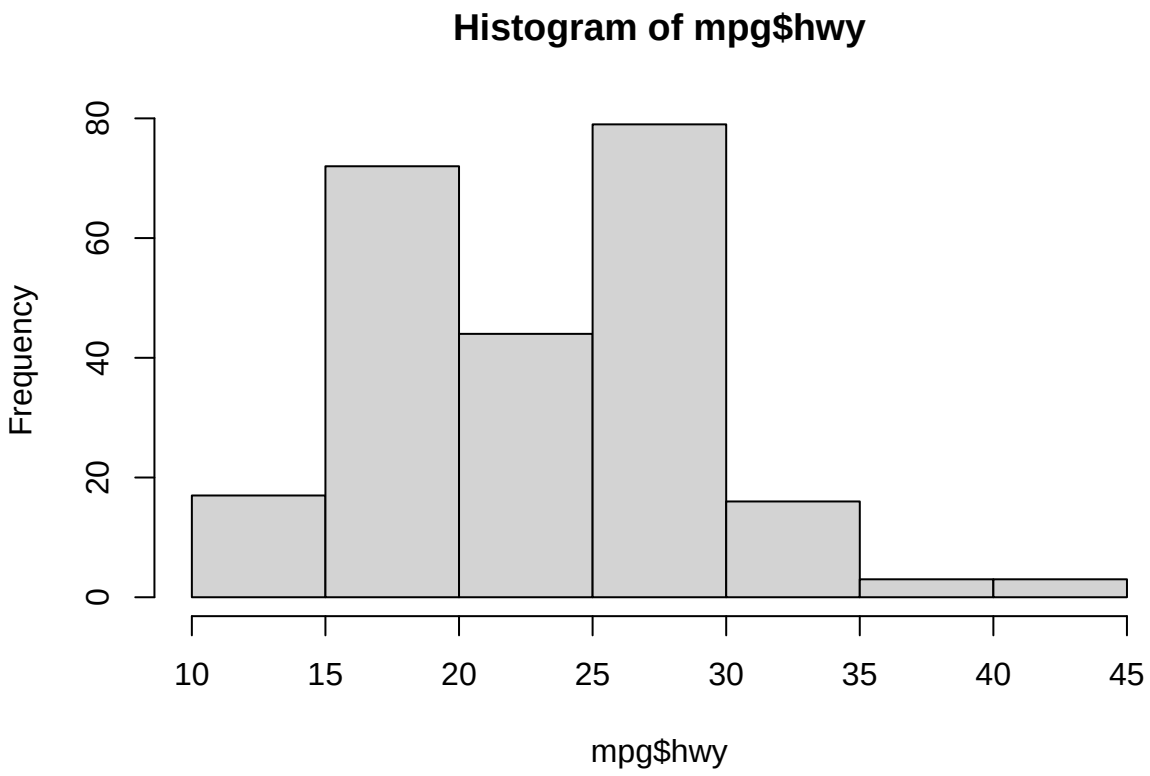
```
ggplot(data=mpg,mapping = aes(x=hwy))+
  geom_histogram()
```



Based on the above chart, what can you say about the shape of the distribution? Is it symmetrical or skewed? Does it seem to be flat or peaked? Does it have a single peak or multiple peaks? Can you spot any outliers?

While the focus here is to visually explore data using `ggplot2`, most of the charts illustrated here can also be constructed using `graphics` package, however the out-of-the-box results don't look as good. Getting to the level of a ggplot2 package is possible but requires specifying a number of arguments. To illustrate this point, here is the result from using `hist()` function from `graphics` package.

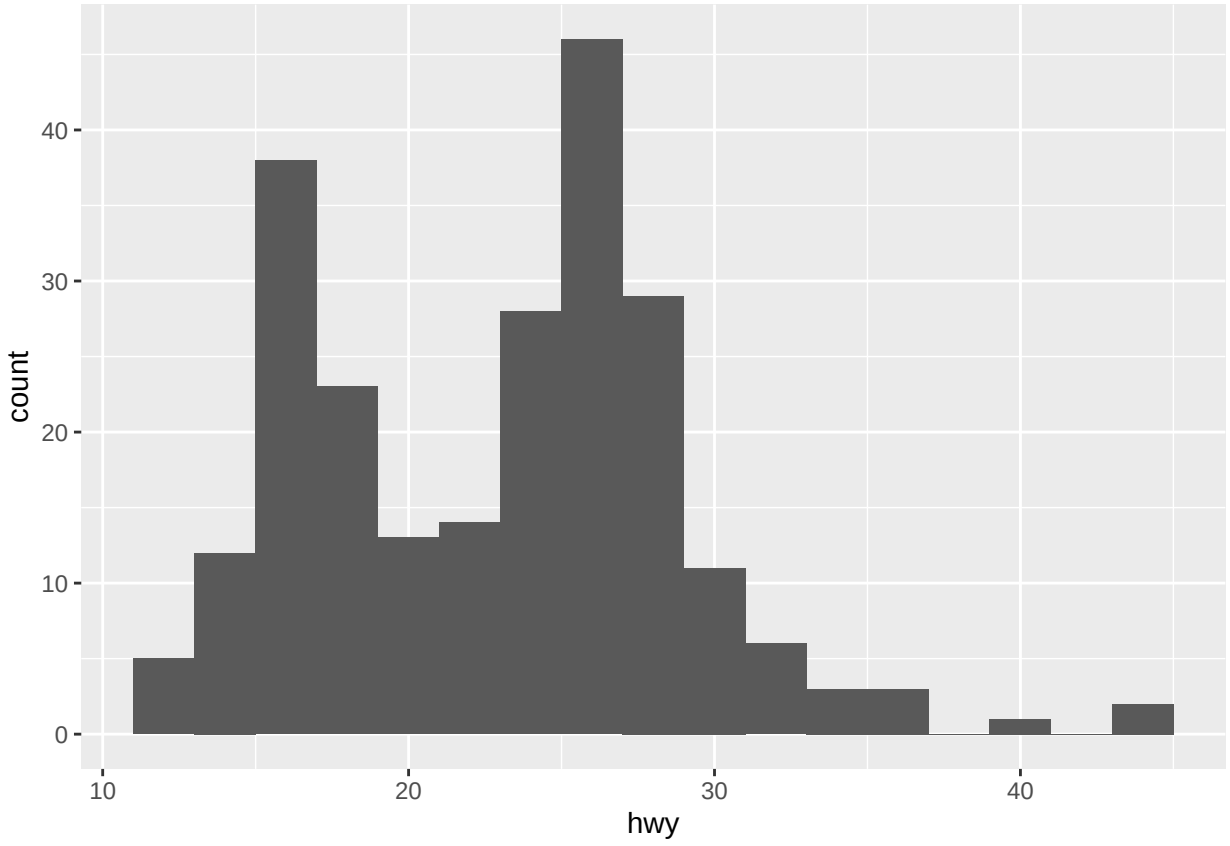
```
hist(mpg$hwy)
```



Binwith

You may have noticed that we didn't specify how to bin `hwy` to create the histogram. This is because geom_histogram() used a default of 30 for number of bins. We can override this by specifying either number of `bins` or a `binwidth`.

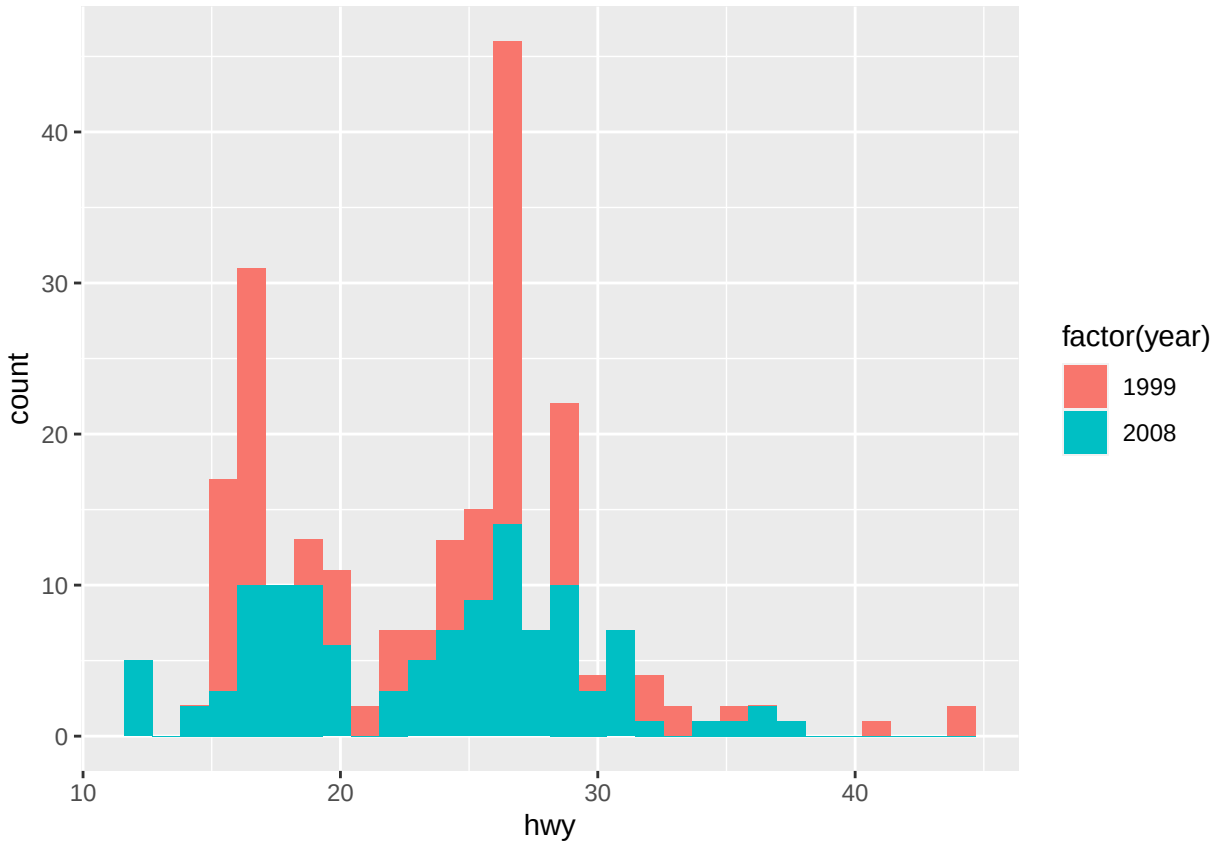
```
ggplot(data=mpg,mapping = aes(x=hwy))+
  geom_histogram(binwidth = 2)
```



Compare distributions

Drawing inferences about the above distribution must be guarded as the data differ in many ways including the model year. So, lets examine the distribution for each year separately. To do this, we will represent `year` using different fill colors. Specifically, we will introduce the following change: `mapping = aes(x=hwy, fill=factor(year))`

```
ggplot(data=mpg,mapping = aes(x=hwy,fill=factor(year)))+
  geom_histogram()
```

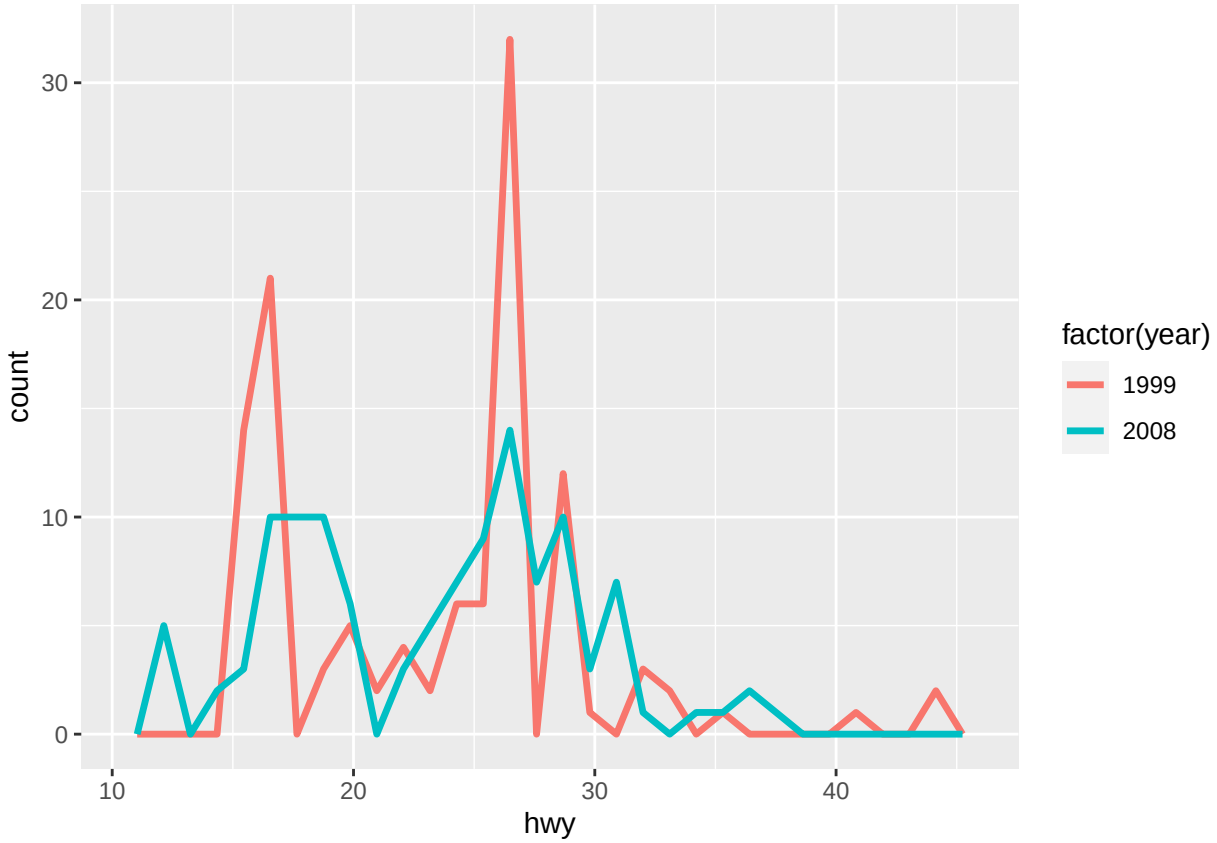


Frequency Polygon

We added a variable to the original histogram to view the distribution of `hwy` across two years, however, overlaid histograms made it hard to see data for 1999. To address this, we will use a line rather than bars. To create a frequency polygon, simply replace the geom for histogram with an aptly named geom called `geom_freqpoly`. Another point worth noting is that an aesthetic meant for one geom may not be appropriate for a different one. In this instance, we need to use a `color` aesthetic rather than `fill` to depict each year. Finally, this time around, we are also going to add a size argument that applies across the geom. To sum up, here are the changes:

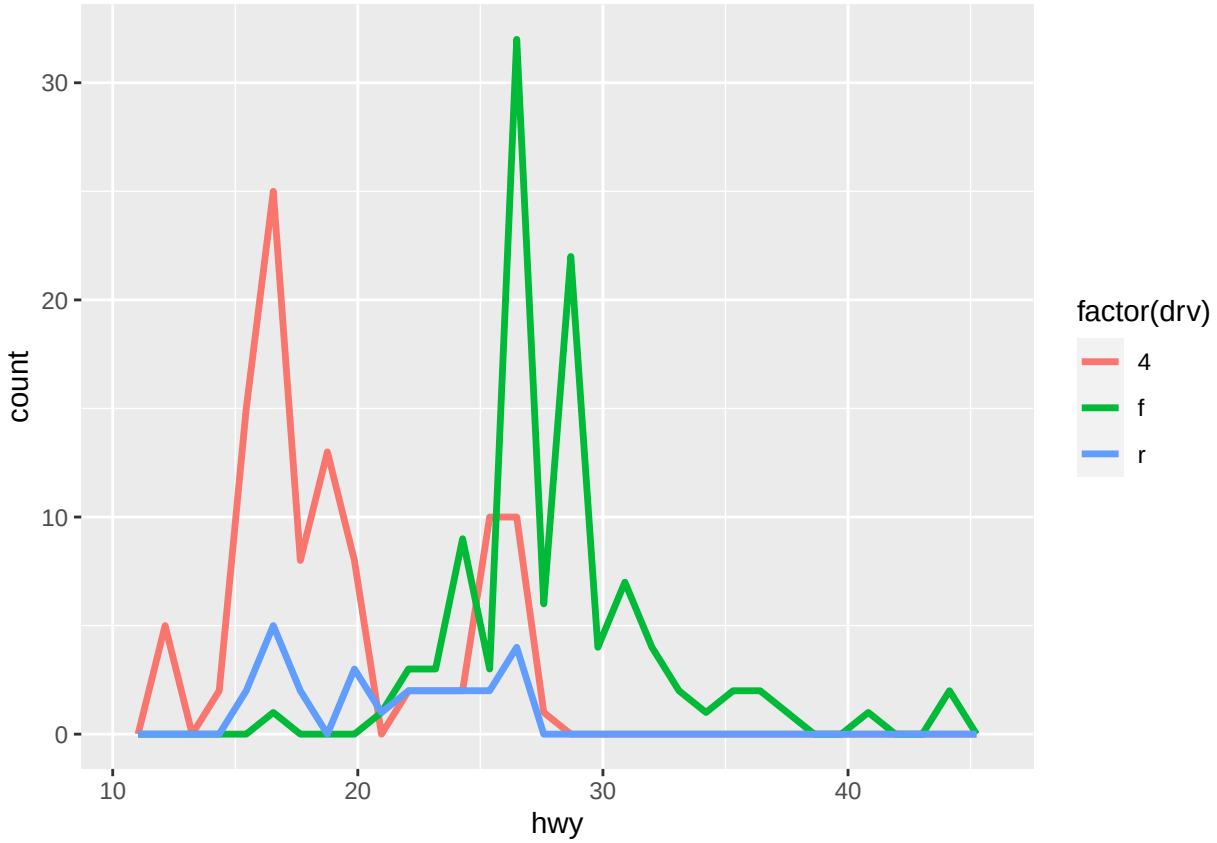
- geom_freqpoly(size=1.2)
- color = factor(year)

```
ggplot(data=mpg,mapping = aes(x=hwy,color=factor(year)))+
  geom_freqpoly(size=1.2)
```



Differences in frequency of observations (i.e., sample size) can make it difficult to compare distributions. While the number of cars for years 1999 and 2008 was the same, this is not the case for drive train (`drv`). Most of the cars are all wheel or front wheel drive.

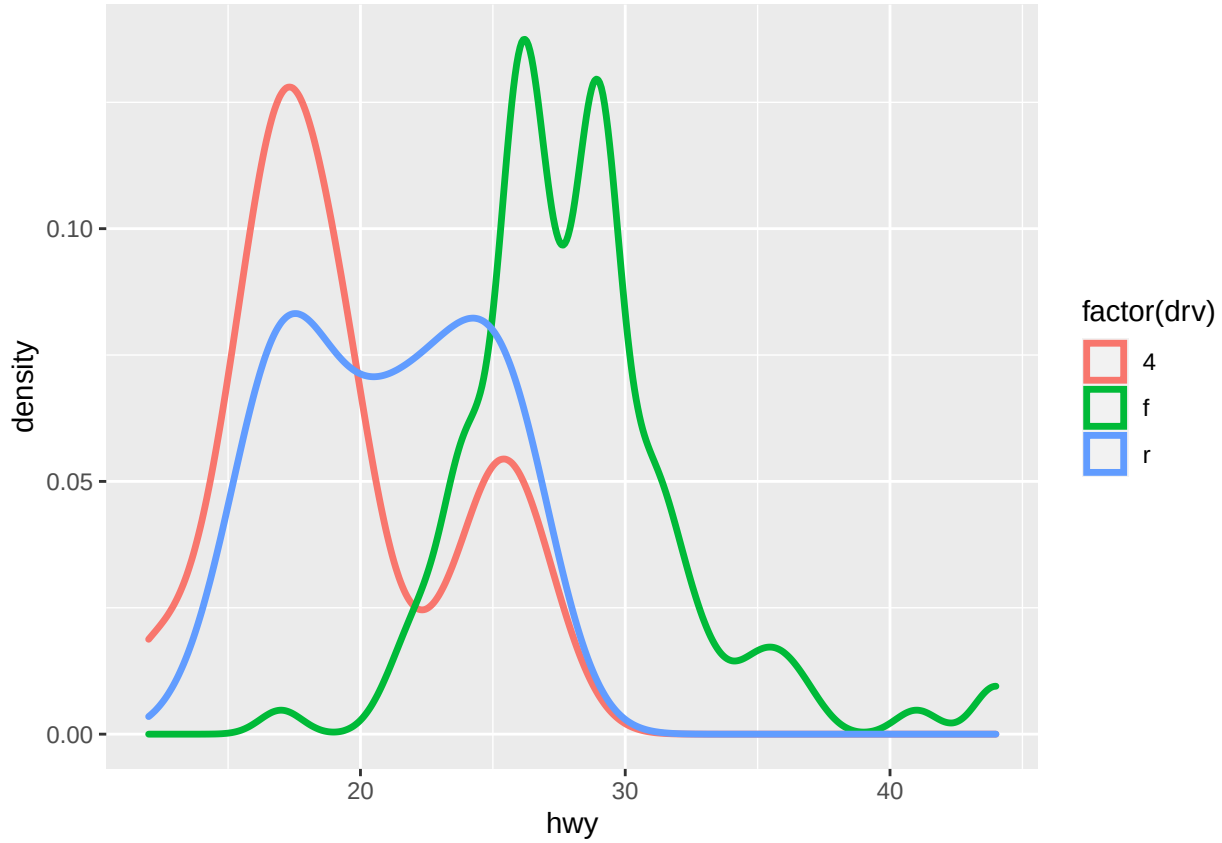
```
ggplot(data=mpg,aes(x=hwy,color=factor(drv)))+
  geom_freqpoly(size=1.2)
```



Density curve

The issue of unequal sample sizes can be addressed by comparing standardized frequency distributions called density curves.

```
ggplot(data=mpg,aes(x=hwy,color=factor(drv)))+
  geom_density(size=1.2)
```



What can you say about the distribution of highway gas mileage different drive trains?

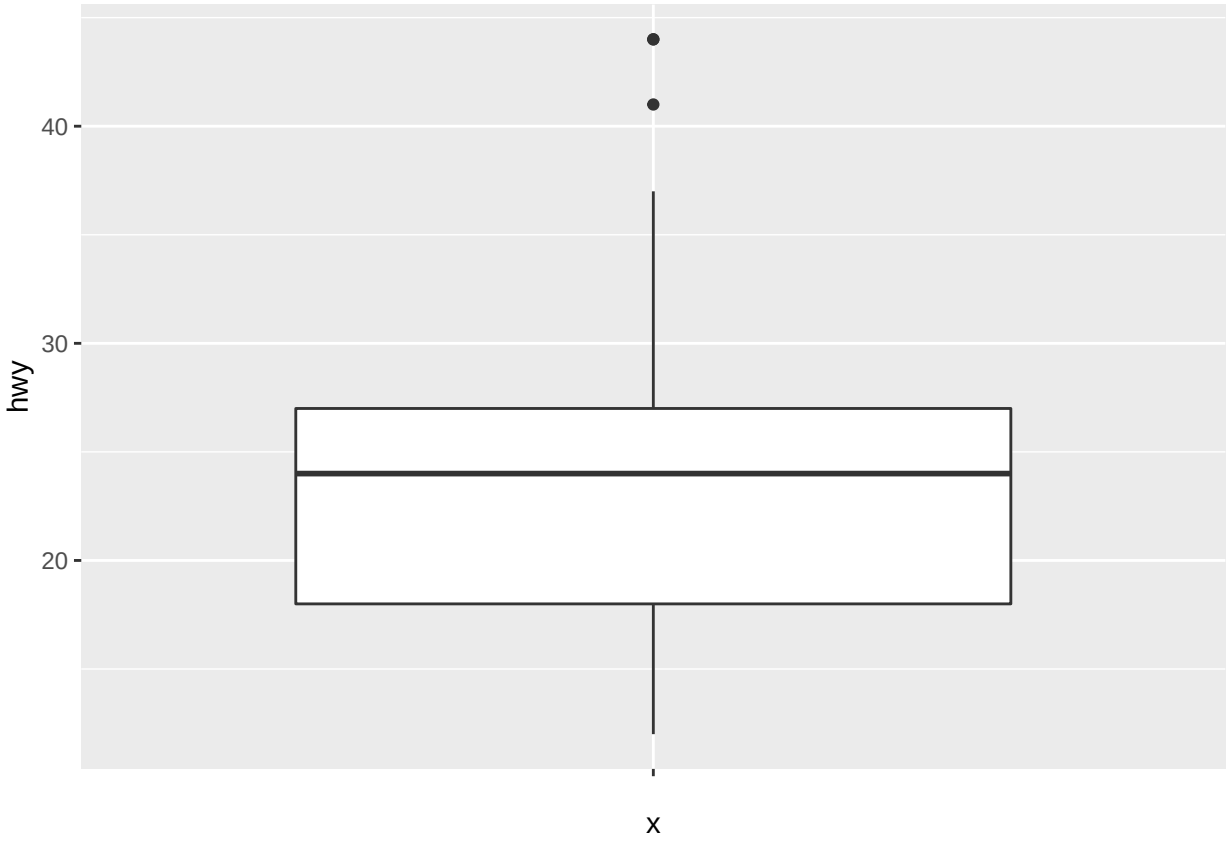
Boxplot

A box plot (also known as a box and whisker plot) is another handy chart for examining a distribution. However, it is not as popular as a histogram for examining distribution of a single variable. On the other hand, these plots are useful for comparing distributions. They are also useful for spotting outliers.

Since, this is a different plot, we are going to use a different geom, intuitively named, `geom_boxplot`. To get started, we will make a boxplot for just one group, which is really not what boxplots are designed for, therefore we are going to use an awkward aesthetic for x. Here is what is new about this plot

- geom_boxplot()
- x = ""

```
ggplot(data=mpg,aes(x="",y=hwy))+
  geom_boxplot()
```

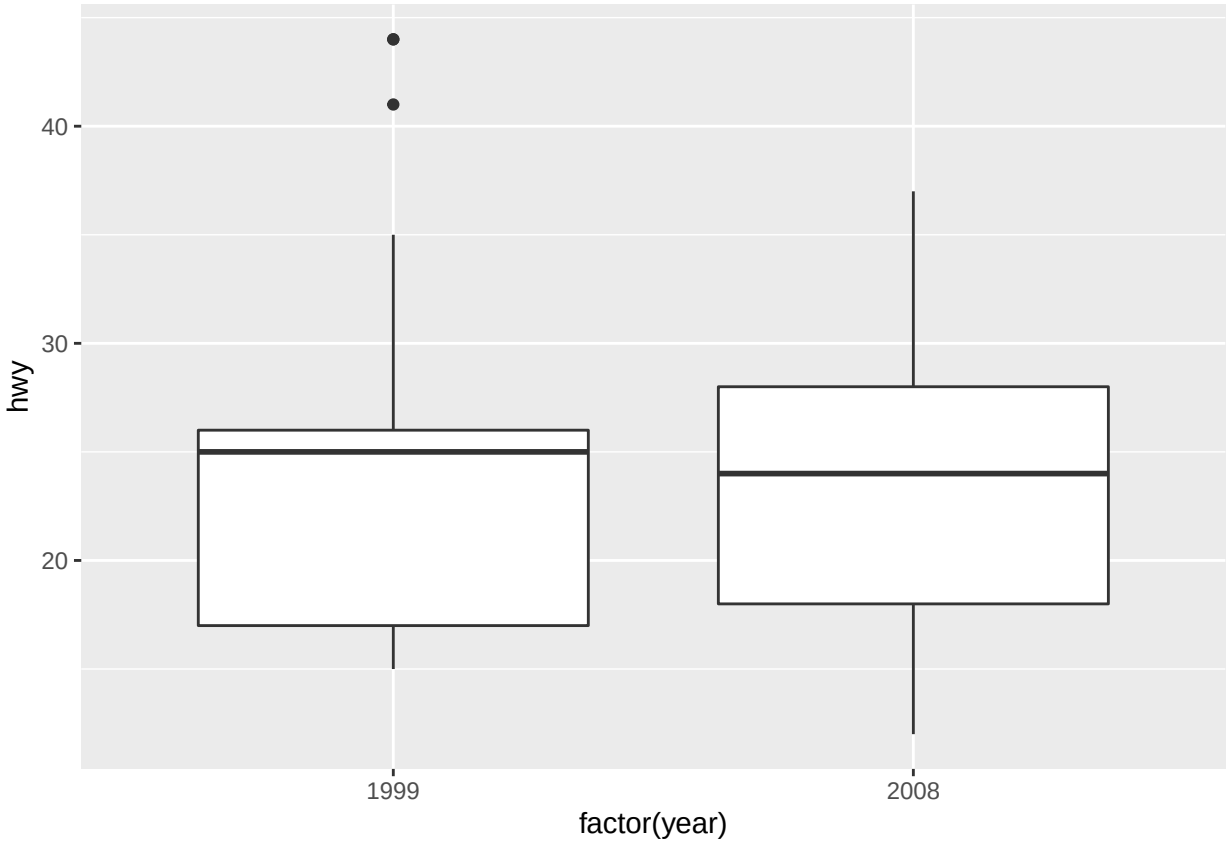


What can you say about the spread (or variance) of gas mileage? Can you spot any outliers?

Boxplot by year

Boxplots are most often used to compare distributions, so let us do just that by contrasting gas mileage across the two years we have data on.

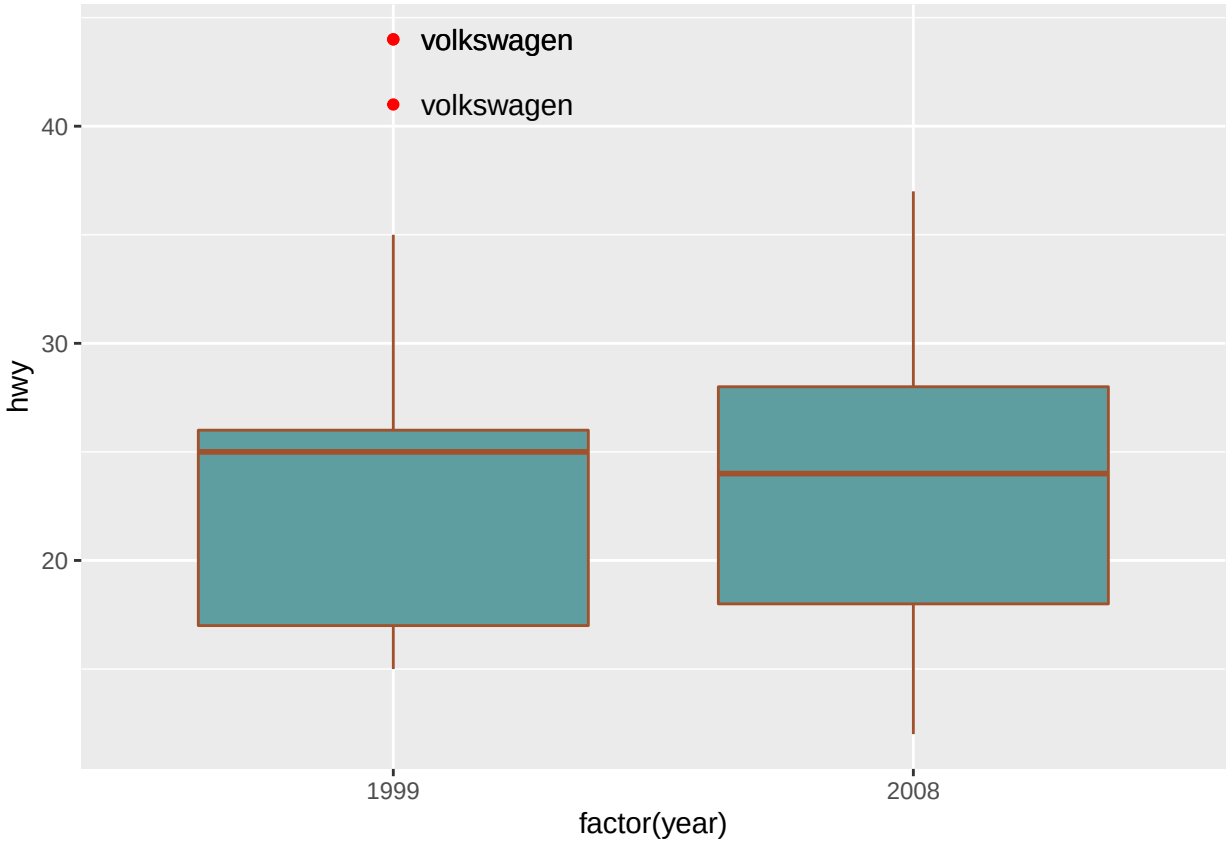
```
ggplot(data=mpg,aes(x=factor(year),y=hwy))+
  geom_boxplot()
```



To draw attention to the plots, and highlight the outliers, we are going to add three cosmetic edits. * Highlight outliers: outlier.color=`red` * Label outliers: geom_text(data=mpg[mpg$hwy > quantile(mpg$hwy,0.99),], aes(label=manufacturer),nudge_x = 0.2) * Add color to the boxes and lines: fill='cadetblue',color='sienna'

The first edit simply highlights outliers with the color red. The second edit is a bit more complicated. While data and mappings loaded into ggplot() apply to all geoms, any given geom can have its own data and mapping. In this case, we have selected all the data in the 99th percentile and passed it to geom_text. Furthermore, we have used `label` aesthetic to display manufacturer name and nudged it 0.2 units so it doesn't eclipse the dot. The third edit is simple in that it fills the boxes with a color and uses a different color for the lines.

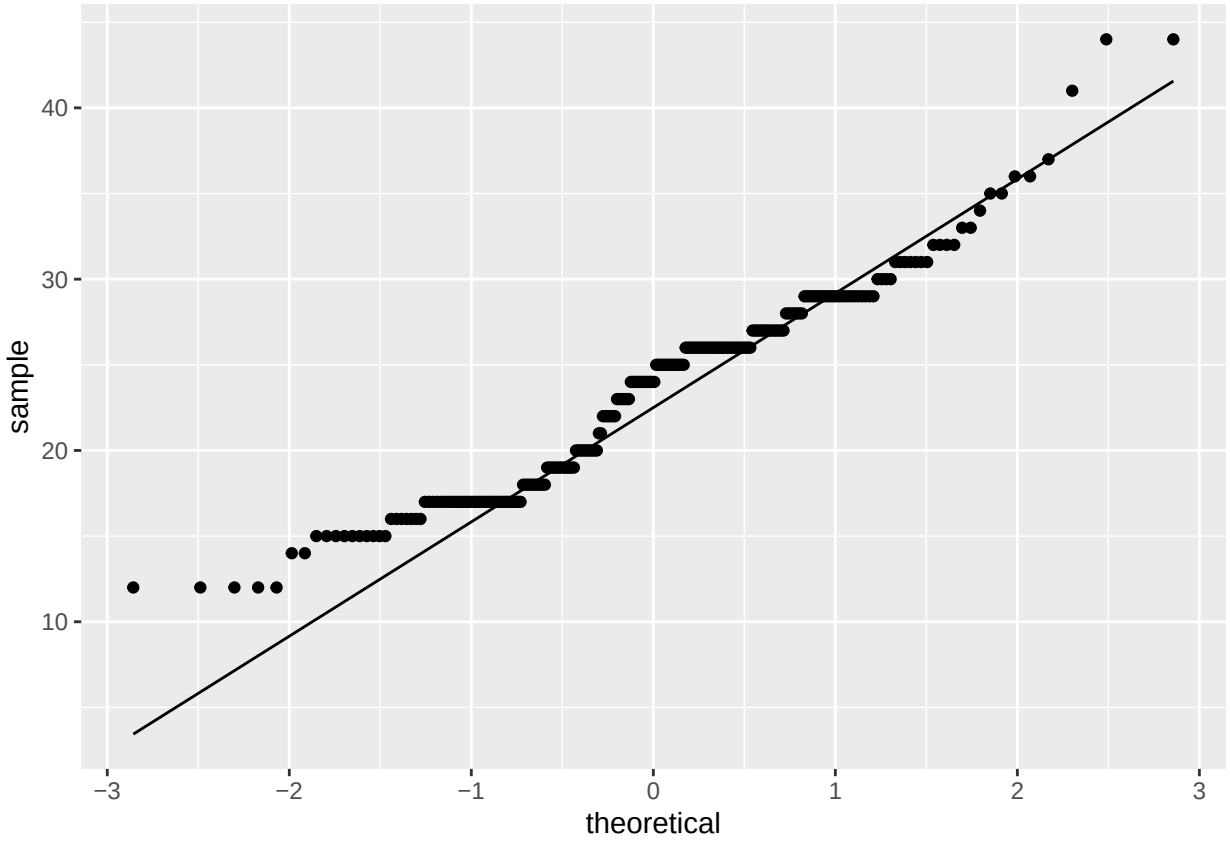
```
ggplot(data=mpg,aes(x=factor(year),y=hwy))+
  geom_boxplot(outlier.color='red', fill='cadetblue',color='sienna')+
  geom_text(data=mpg[mpg$hwy>quantile(mpg$hwy,0.99),], aes(label=manufacturer),nudge_x = 0.2)
```



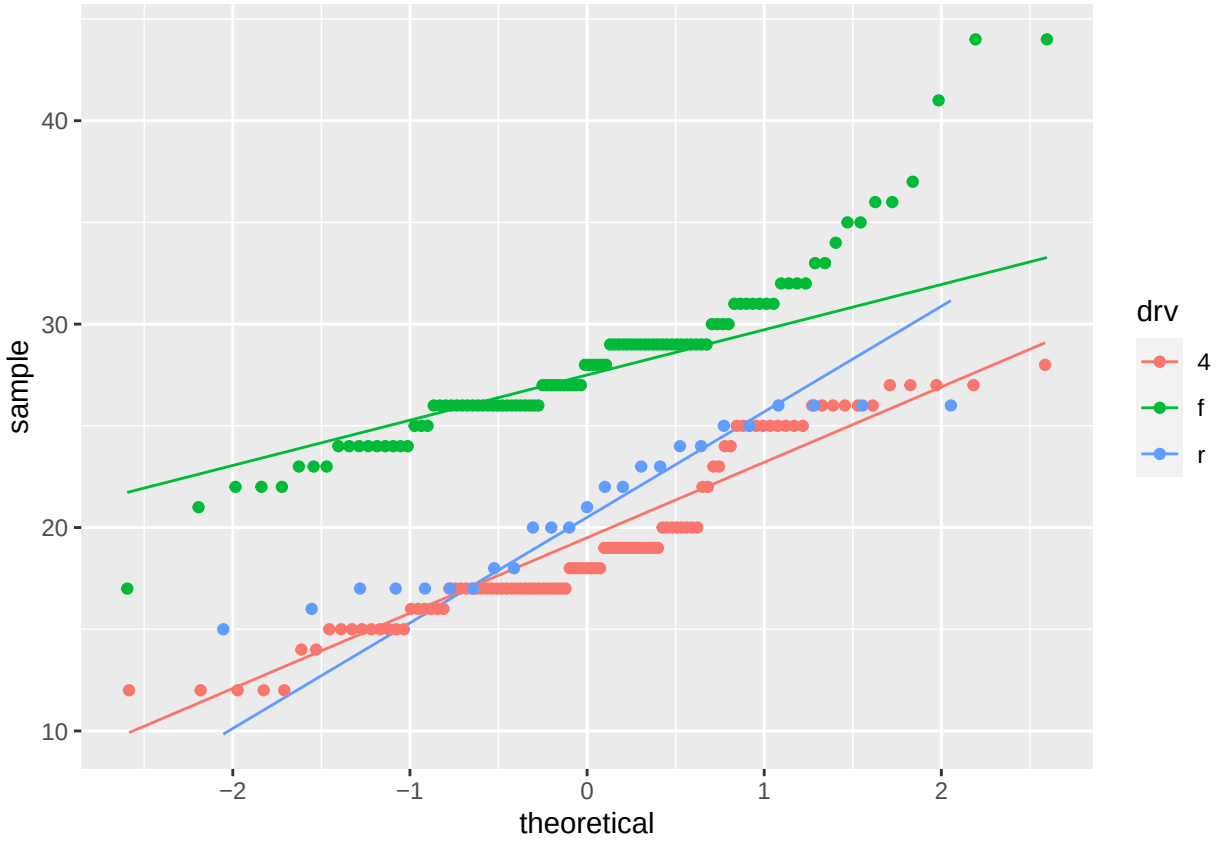
QQ Plot

Most commonly used to see if data comes from a normal distribution by plotting quantiles of data against quantiles expected from a normal distribution. If the data are normally distributed the points in the plot will cluster around the diagonal running from bottom left to top right. Lets examine distribution of `hwy` and then examine it for each drive train, `drv`.

```
ggplot(data=mpg, aes(sample=hwy))+
  geom_qq_line()+
  geom_qq()
```




```
ggplot(data=mpg, aes(sample=hwy, color=drv))+
  geom_qq_line()+
  geom_qq()
```



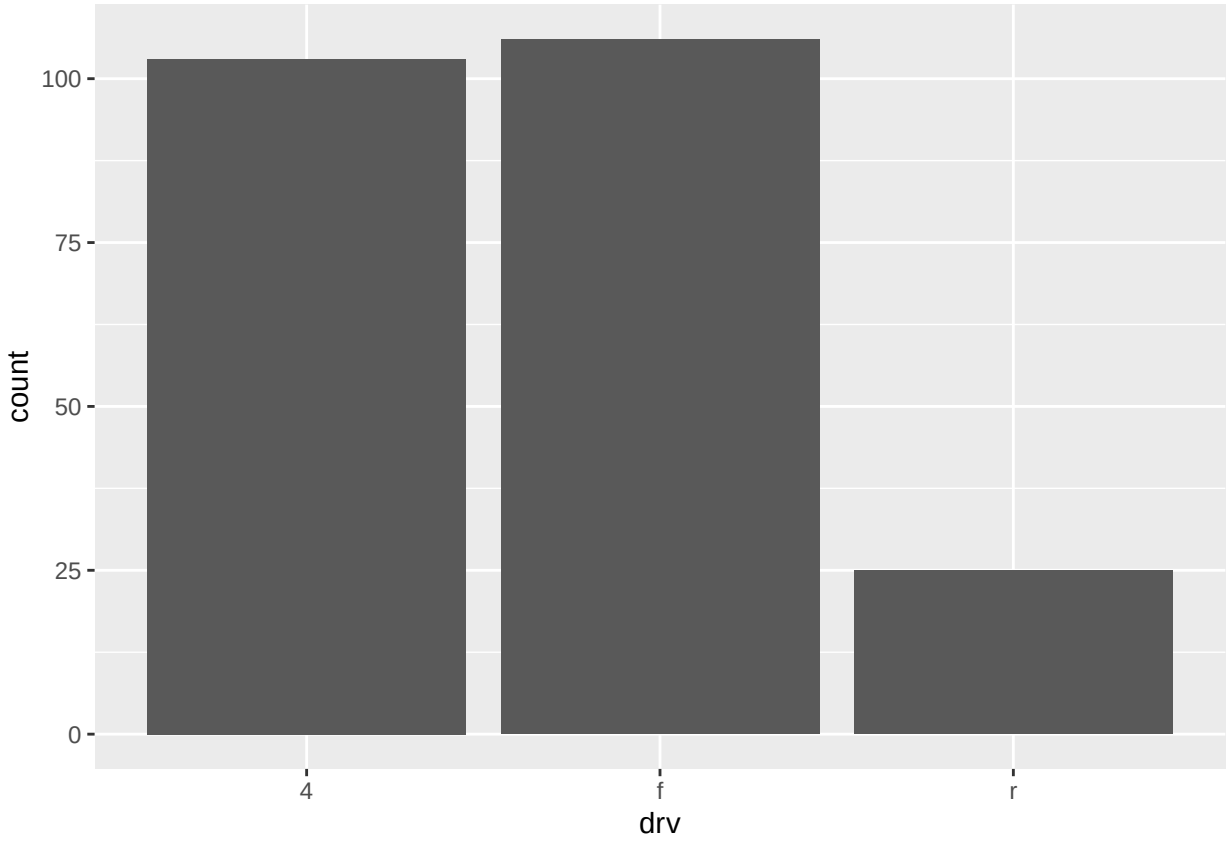
Compare Groups

Now, we turn our attention to charts to compare groups.

Bar Chart

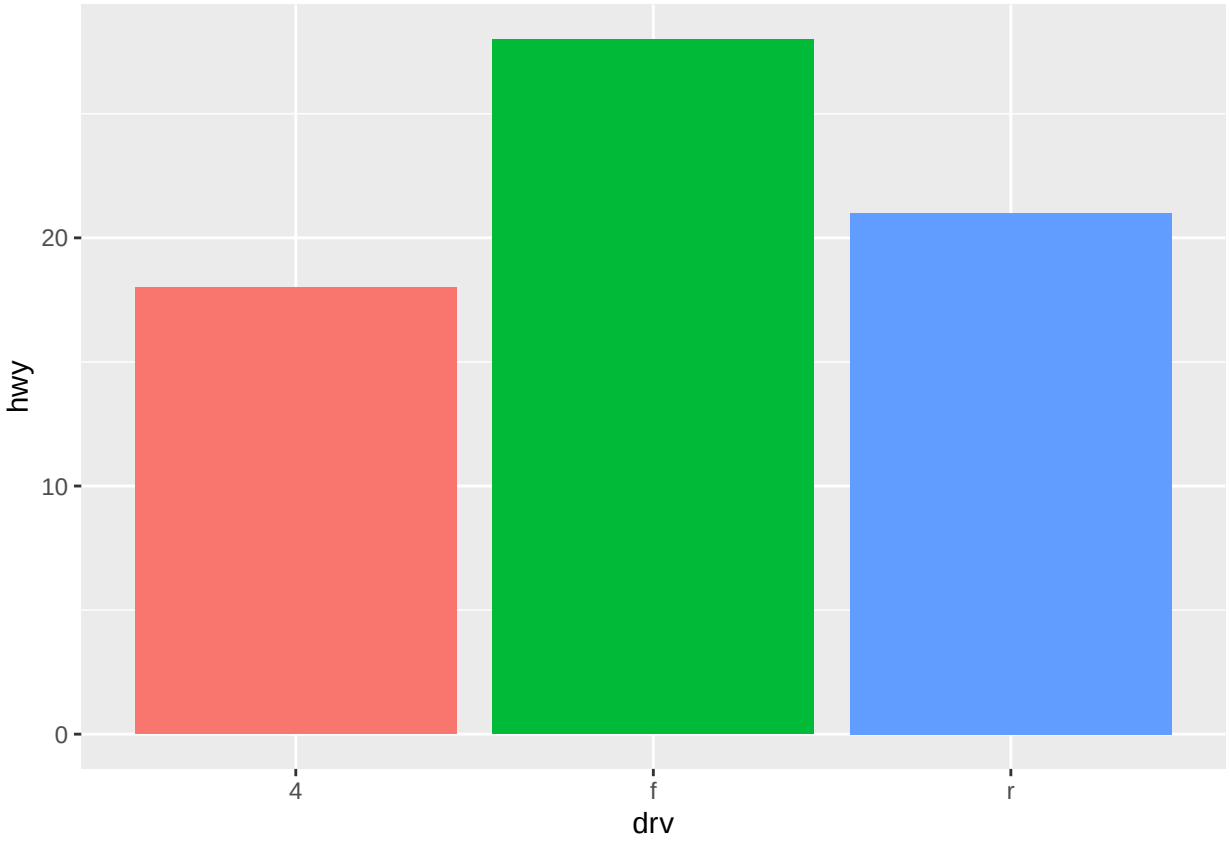
Typically used to compare summary of numerical variables across levels of a categorical variable. Bar charts are created using the geom, `geom_bar`. To get familiar with the workings of geom_bar, let us create a bar chart that compares the number of cars of each drive train, `drv`. Since the goal is to compare counts, we don't need to specify a numerical variable.

```
ggplot(data=mpg, aes(x=drv))+
  geom_bar()
```



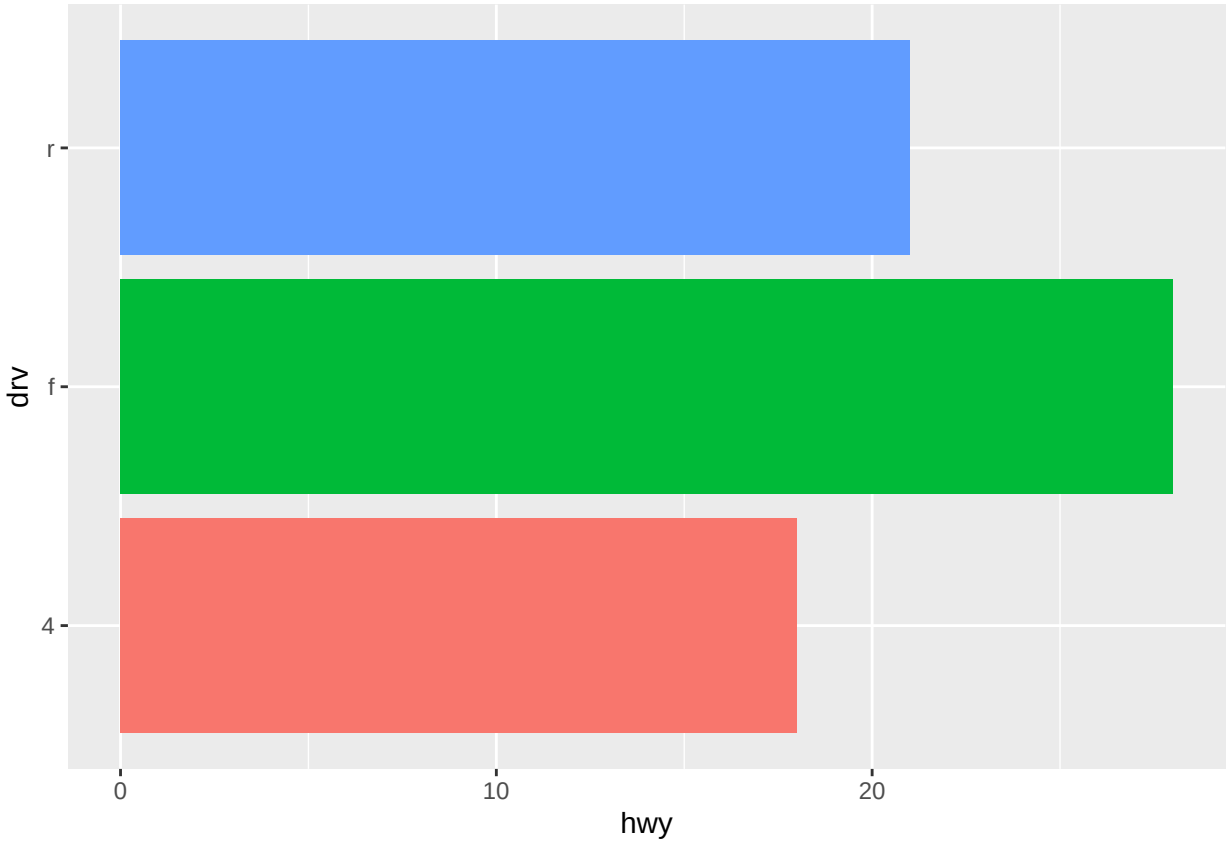
Of course, a more typical use of a bar chart is to compare a numerical summary (e.g., median, mean), across categories. Let us compare median highway gas mileage across drive trains. To do this, we need to specify the `y` variable in mapping. In addition, `geom_bar` has to be told how to summarize the y variable by passing specifying a `stat`. Finally, we wil use a fill aesthetic to distinguish the bars and turn off the legend. So, we are going to specify
* fill = drv
* stat = 'summary'
* fun = 'median'
* show.legend=F

```
ggplot(data=mpg, aes(x=drv, y=hwy, fill=drv))+
  geom_bar(stat='summary', fun='median', show.legend=F)
```



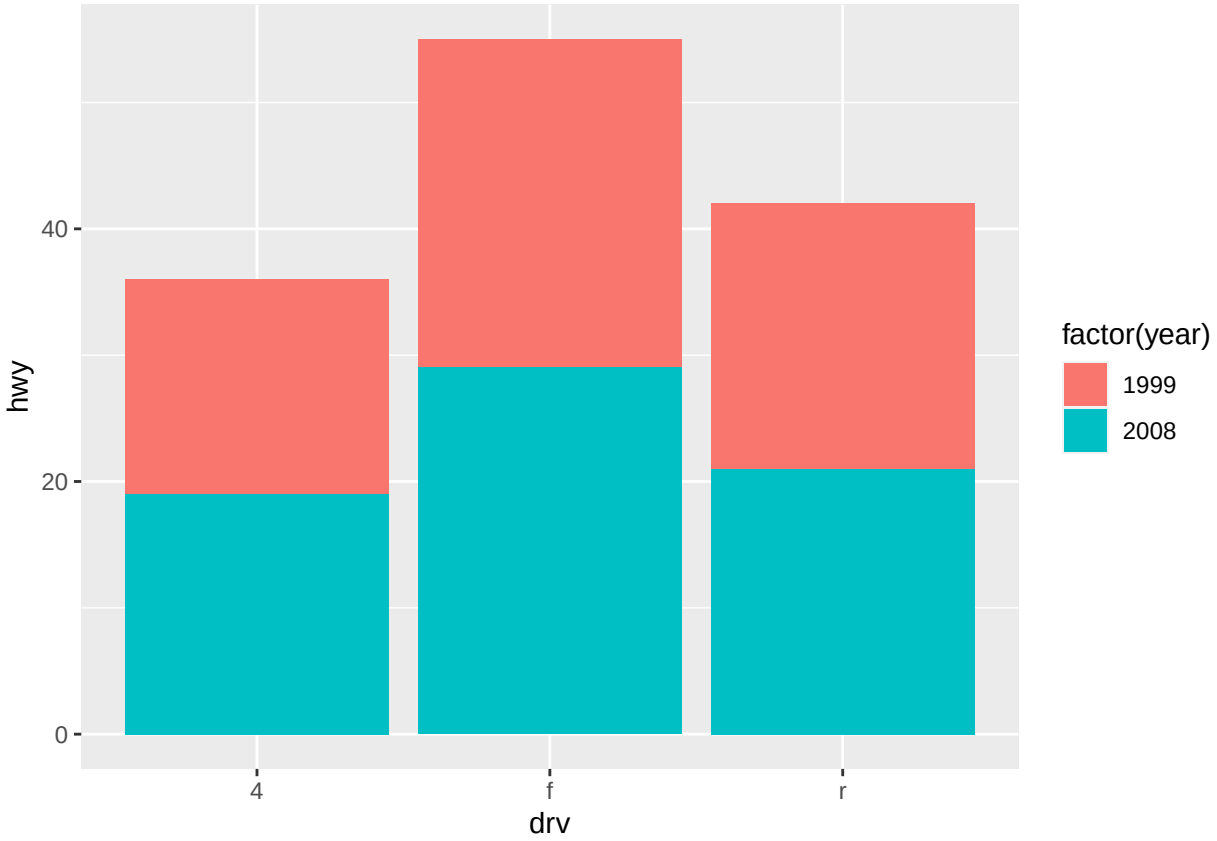
The chart constructed above is a vertical bar chart, also known as a column chart. This can be transformed into a horizontal bar chart by flipping the axes using coord_flip()

```
ggplot(data=mpg, aes(x=drv, y=hwy, fill=drv))+
  geom_bar(stat='summary', fun='median', show.legend=F)+
  coord_flip()
```



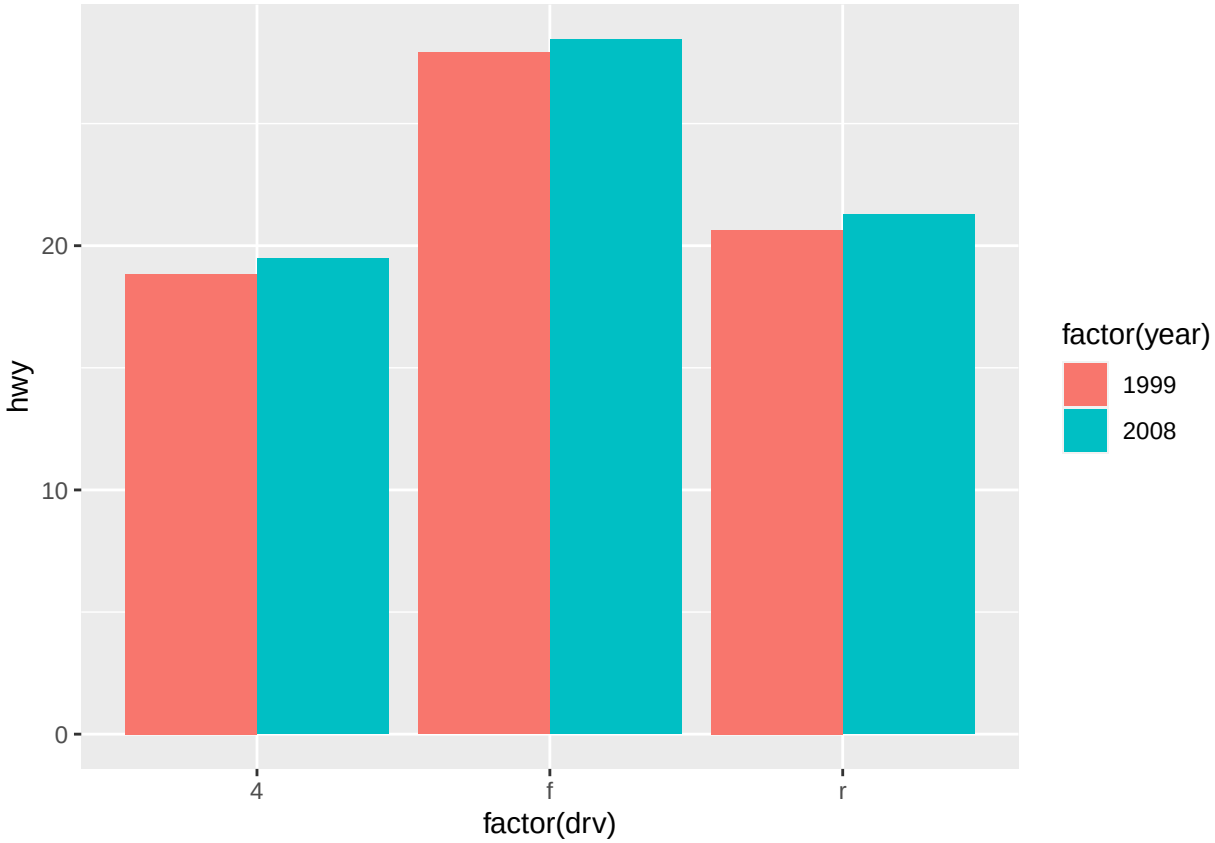
Often, one is interested in examining numerical summaries for more than one categorical variable. With the x- and y- aesthetics used for the categorical and numeric variable, we can look to other aesthetics to represent additional variables. Here, we will introduce `year` to the above chart using a `fill` aesthetic. Note, we are passing `factor(year)` to fill because we want to treat `year` as discrete, not continuous. The latter will represent `year` using shades of gray.

```
ggplot(data=mpg,aes(x=drv,fill=factor(year),y=hwy))+
  geom_bar(stat = 'summary',fun='median')
```



The above chart does the job of representing a third variable, but it is hard to interpret. This is because `year` is stacked on for each `drv`, because the default argument for position is stack. Let us explicitly specify `position` as `dodge`.

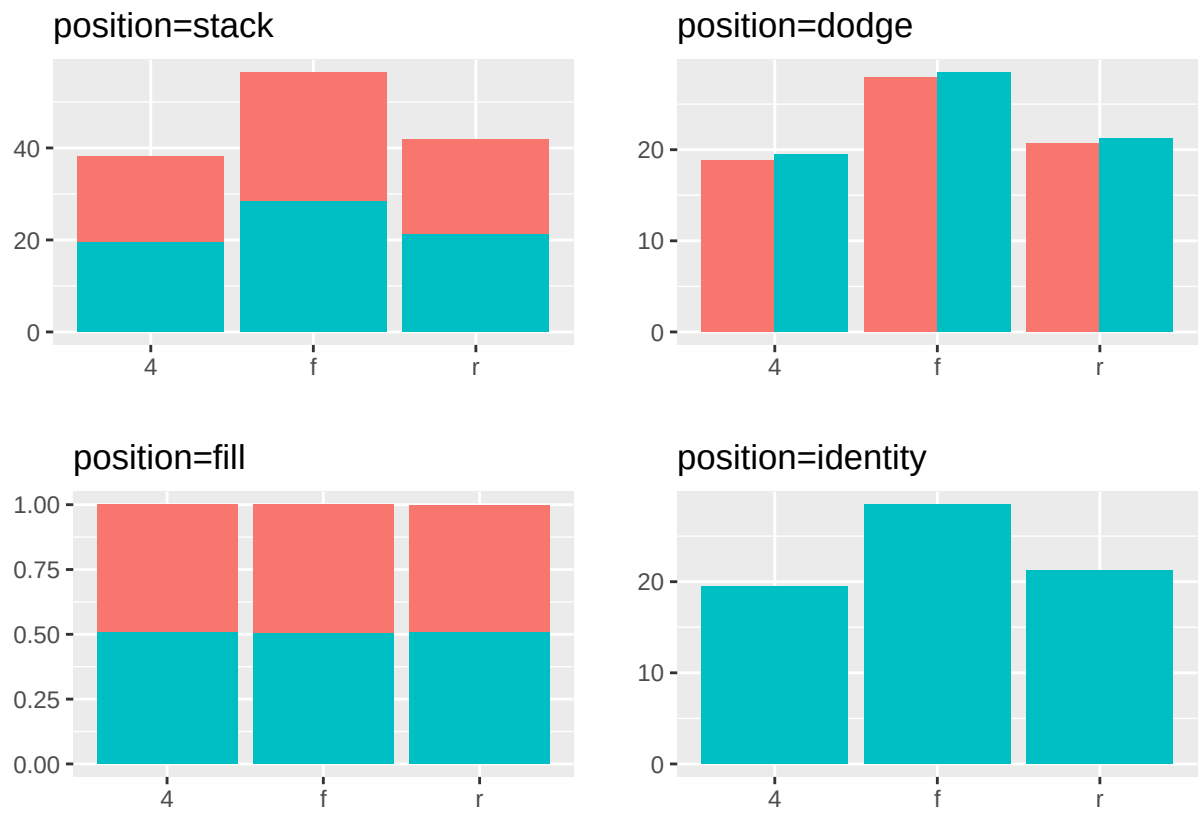
```
ggplot(data=mpg,aes(x=factor(drv),fill=factor(year),y=hwy))+
  geom_bar(stat = 'summary', fun='mean', position='dodge')
```



Now, you may wonder, if the goal was to color bars differently, why not use the color aesthetic instead of fill. Try it out and you will learn something new. Also, if you are curious about how fill uses color, use `year` instead of `factor(year)`.

Here is a visual of what the different values of `position` will do. (`jitter` is not illustrated as it is not meaninful in a bar chart).

```
positions = c('stack','dodge','fill','identity')
g = lapply(X = positions,FUN = function(x){
                            ggplot(data=mpg,aes(x=drv,y=hwy,fill=factor(year)))+
                            geom_bar(stat = 'summary',fun=mean,position=x)+
                            ggtitle(label=paste('position=',x,sep=''))+
                            guides(fill=F)+xlab('')+ylab('')} )
library(gridExtra)
grid.arrange(grobs = g)
```



Fun Exercise: Construct a bar chart to compare hwy for different drv and year. Which position is best?

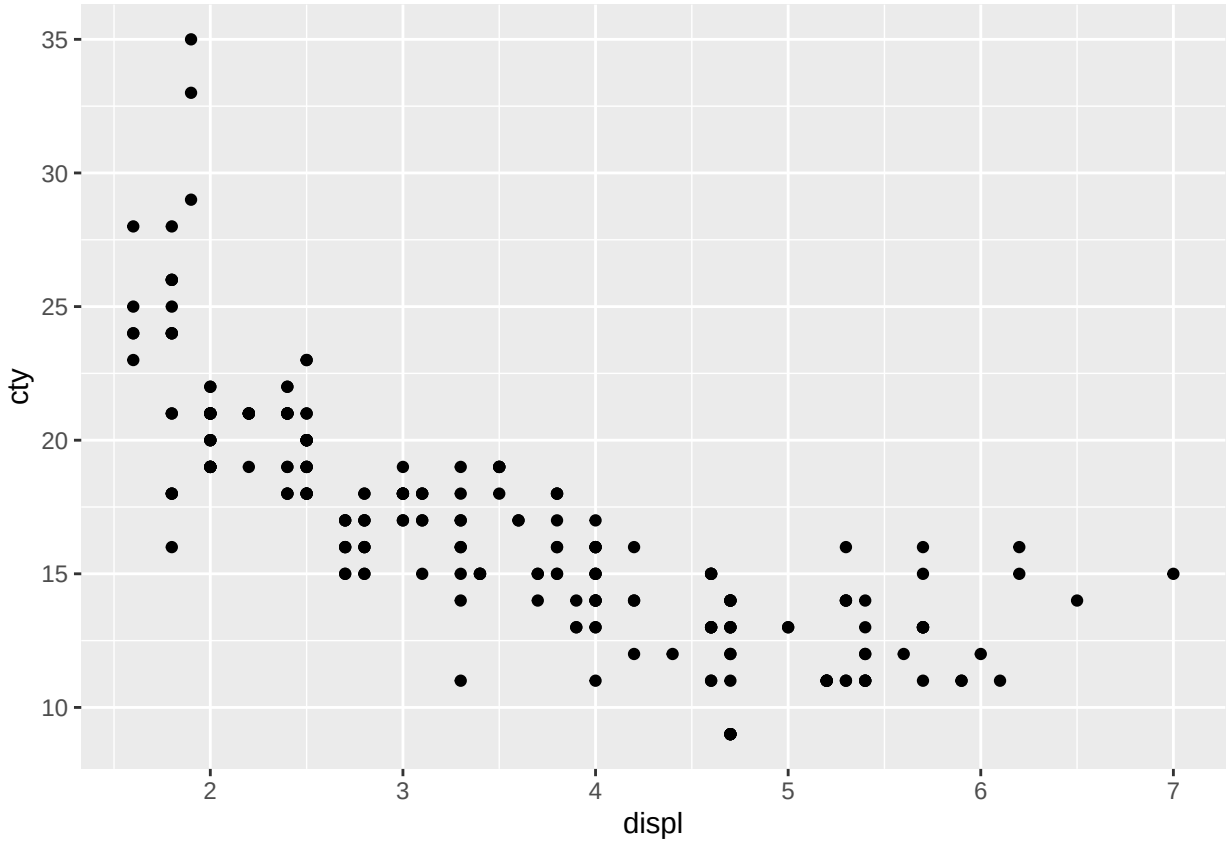
Explore Relationships

Now, we will examine charts for exploring relationships between numeric variables.

Scatterplot

A set of dots on a graph where the location of the dots is based on its value for the two numeric variables. The pattern of the dots in a scatterplot may reveal the existence and nature of relationship between variables. This is implemented using `geom_point()`. We are going to explore the relationship between engine displacement, `displ` and city gas mileage, `cty`.

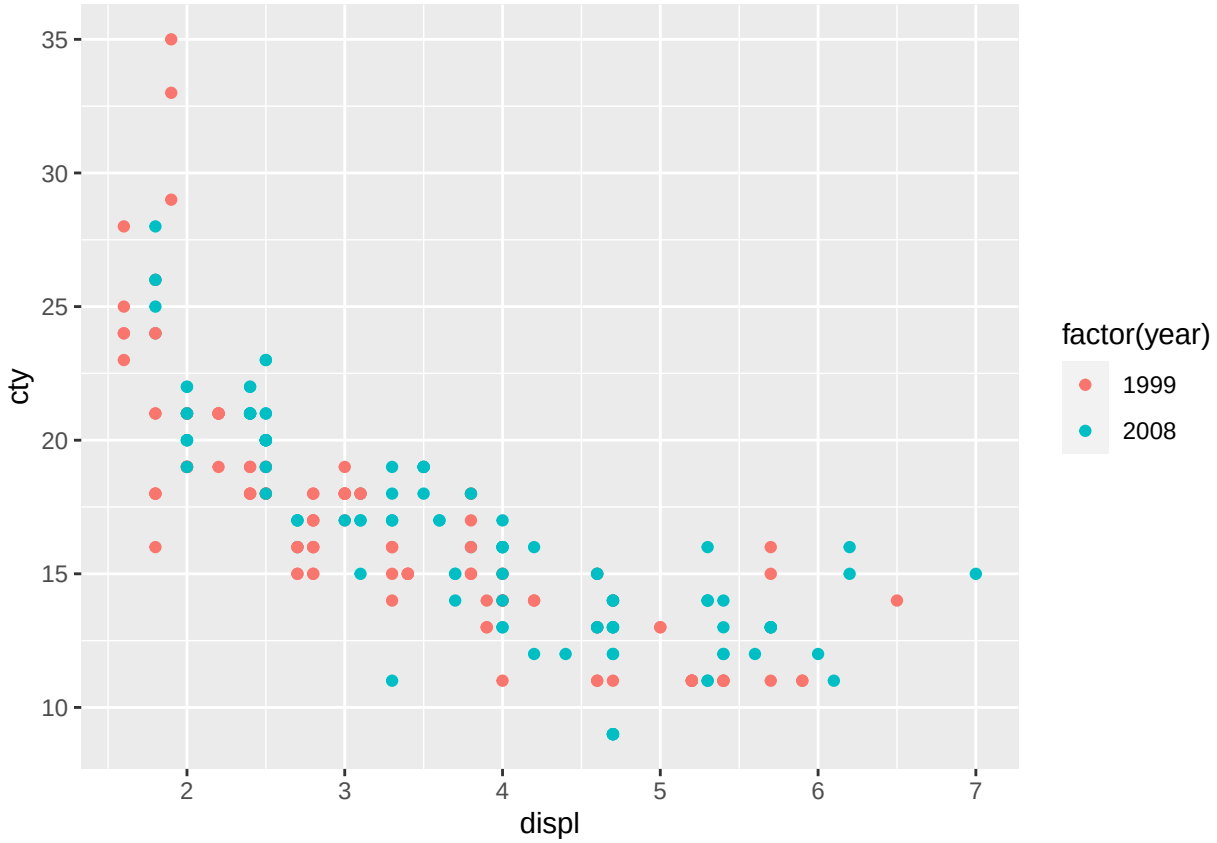
```
ggplot(data=mpg,aes(x=displ,y=cty))+
  geom_point()
```



What does this graph say about the relationship between displ and cty?

Let us now examine the relationship across cars made in different years by introducing `factor(year)` using a `color` aesthetic.

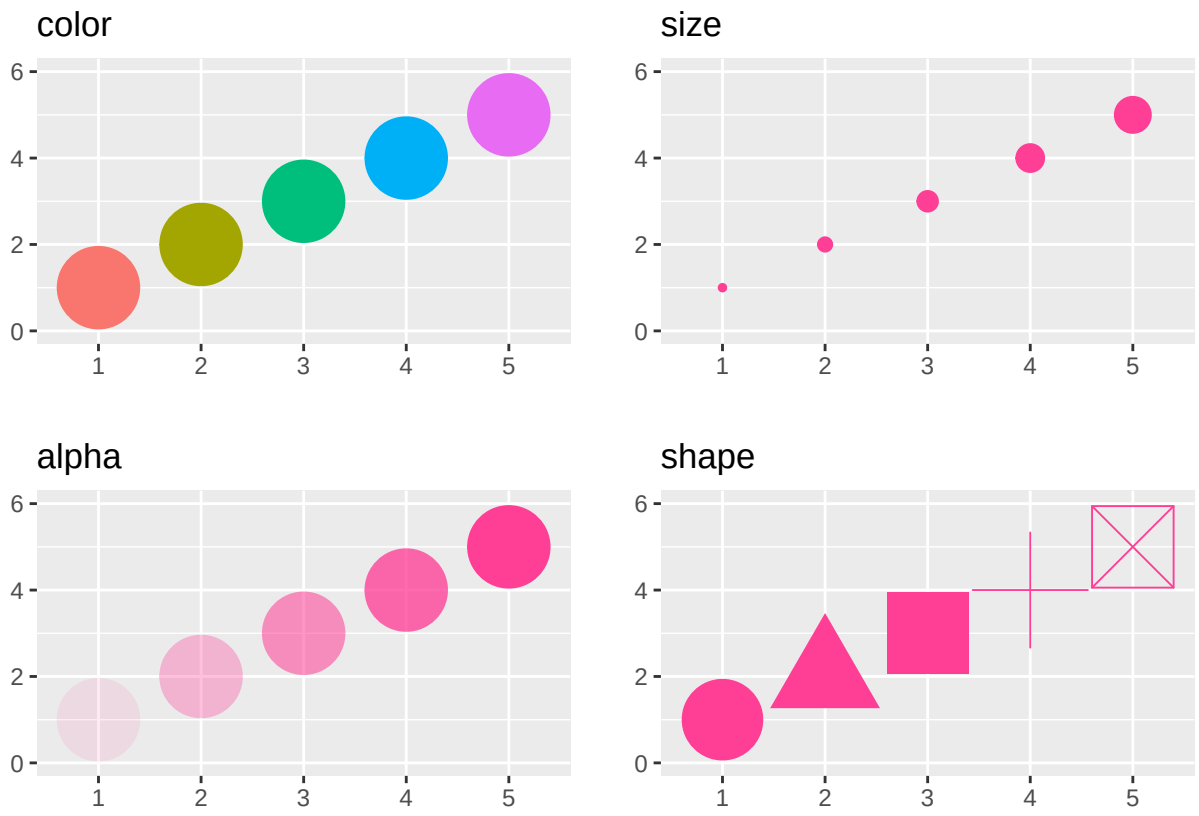
```
ggplot(data=mpg,aes(x=displ,y=cty,color=factor(year)))+
  geom_point()
```



In the above example, we represented year using a color aesthetic. We could have represented year using a different aesthetic such as size, shape or alpha. An aesthetic is a visual property of the graph. The visual properties we can change include color, size, shape, alpha (or transparency) of the points. Here is a visual of the aesthetics and what they can do. (The fill aesthetic is not useful for scatter plots, so it is not illustrated here).

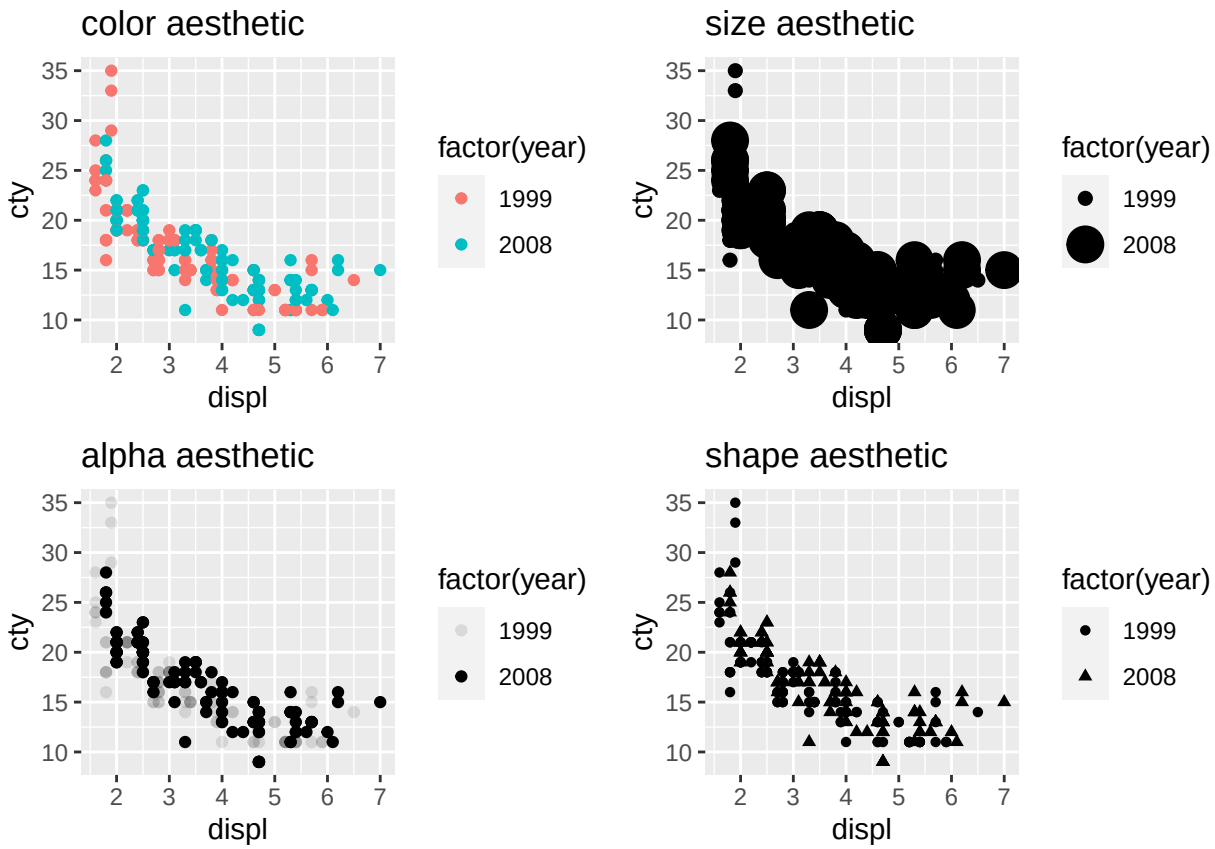
```
temp = data.frame(x=c('1','2','3','4','5'),y=1:5)
g1 = ggplot(data=temp,aes(x=x,y=y,color=x))+
  geom_point(size=14)+
  guides(color=F)+
  ggtitle('color')+
  ylim(0,6)+xlab('')+ylab('')
g2 = ggplot(data=temp,aes(x=x,y=y,size=y^3))+
  geom_point(color='violetred1')+
  guides(size=F)+
  ggtitle('size')+
  ylim(0,6)+xlab('')+ylab('')
g3 = ggplot(data=temp,aes(x=x,y=y,alpha=y))+
  geom_point(size=14,color='violetred1')+
  guides(alpha=F)+
  ggtitle('alpha')+
  ylim(0,6)+xlab('')+ylab('')
g4 = ggplot(data=temp,aes(x=x,y=y,shape=x))+
  geom_point(size=14,color='violetred1')+
  guides(shape=F)+
  ggtitle('shape')+
  ylim(0,6)+xlab('')+ylab('')
```

```
library(gridExtra)
chart = grid.arrange(g1,g2,g3,g4)
```



The choice of aesthetic is important in revealing patterns in the data as is obvious from the four charts below.

```
g1 = ggplot(data=mpg,aes(x=displ,y=cty,color=factor(year)))+
  geom_point()+ggtitle('color aesthetic')
g2 = ggplot(data=mpg,aes(x=displ,y=cty,size=factor(year)))+
  geom_point()+ggtitle('size aesthetic')
g3 = ggplot(data=mpg,aes(x=displ,y=cty,alpha=factor(year)))+
  geom_point()+ggtitle('alpha aesthetic')
g4 = ggplot(data=mpg,aes(x=displ,y=cty,shape=factor(year)))+
  geom_point()+ggtitle('shape aesthetic')
library(gridExtra)
chart = grid.arrange(g1,g2,g3,g4)
```



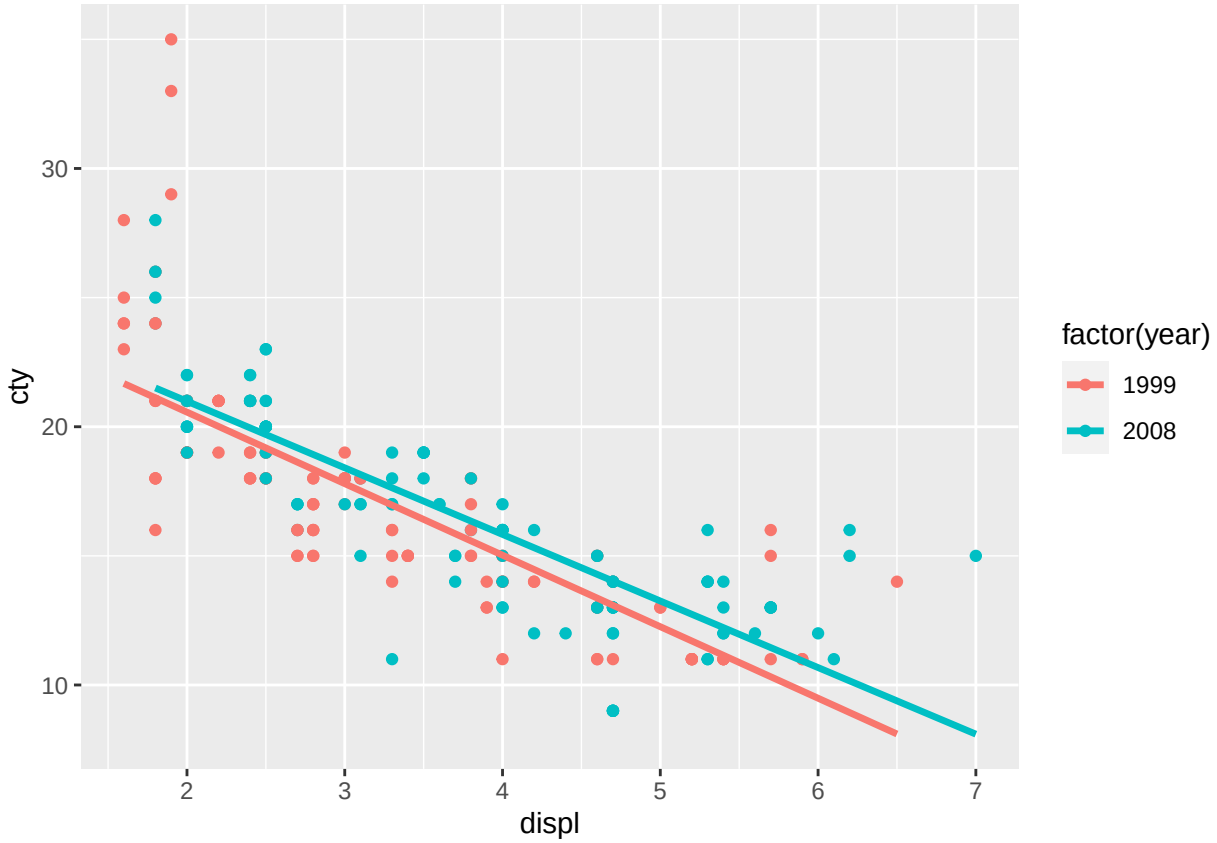
From the charts above, the importance of the type of aesthetic used may have become apparent. A few other things to bear in mind about the use of aesthetics: fill does not work with scatter plots, shape only works with six or fewer categories, and alpha may be better suited for numerical variables. Finally, while there is no limit on how many aesthetics may be used in a single plot, using too many risks muddying interpretation.

Exercise: Try plotting cyl onto the chart by using one of the five aesthetics. Next, experiment with year. You will notice that the aesthetics behave differently depending on the class of the variable.

Line Graph

With large number of points and overplotting, spotting a trend in a scatterplot can be difficult. A line graph can clearly depict the trend in the data. Moreover, with the layering architecture of ggplot2, one doesn't have to choose. Let us layer a regression geom on to the scatterplot illustrated earlier. Since we are using a color aesthetic, the resulting plot will have a regression line for each level of the color aesthetic, `year`. (This is because ggplot2 will automatically group the data when we map an aesthetic to a discrete variable).

```
ggplot(data=mpg,aes(x=displ,y=cty,color=factor(year)))+
  geom_point()+
  geom_smooth(method='lm',se=F,size=1.2)
```



Multiple Charts

In this section, we will look at ways to expand our view by examining multiple variables and multiple charts.

Facets

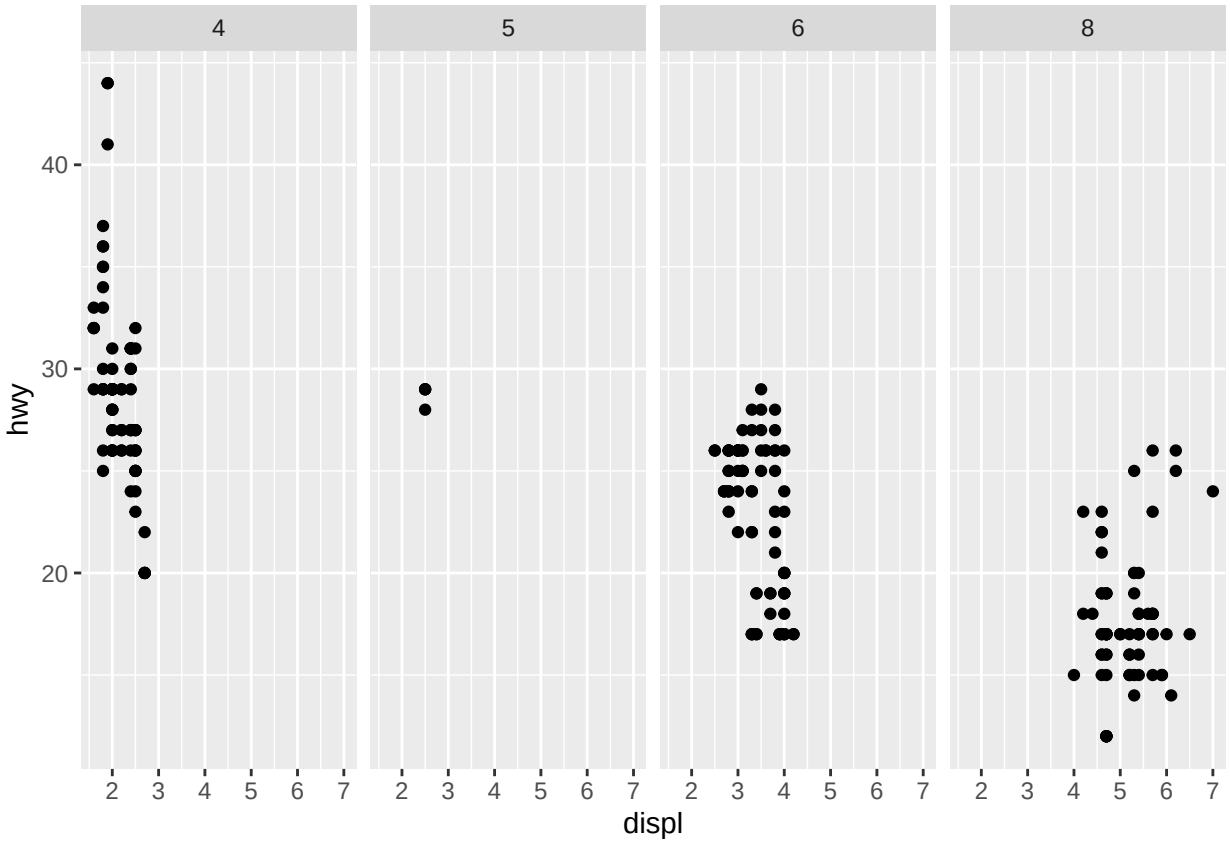
Aesthetics are a handy way to represent variables but sometimes including too many aesthetics may muddy the interpretation of the chart. An alternative to cramming in many variables into a single chart is to spread them over multiple related charts.

Facets create a grid of smaller plots that display different subsets of the data

Row of Charts

Using .~cyl in facet_grid() will generate a row of charts

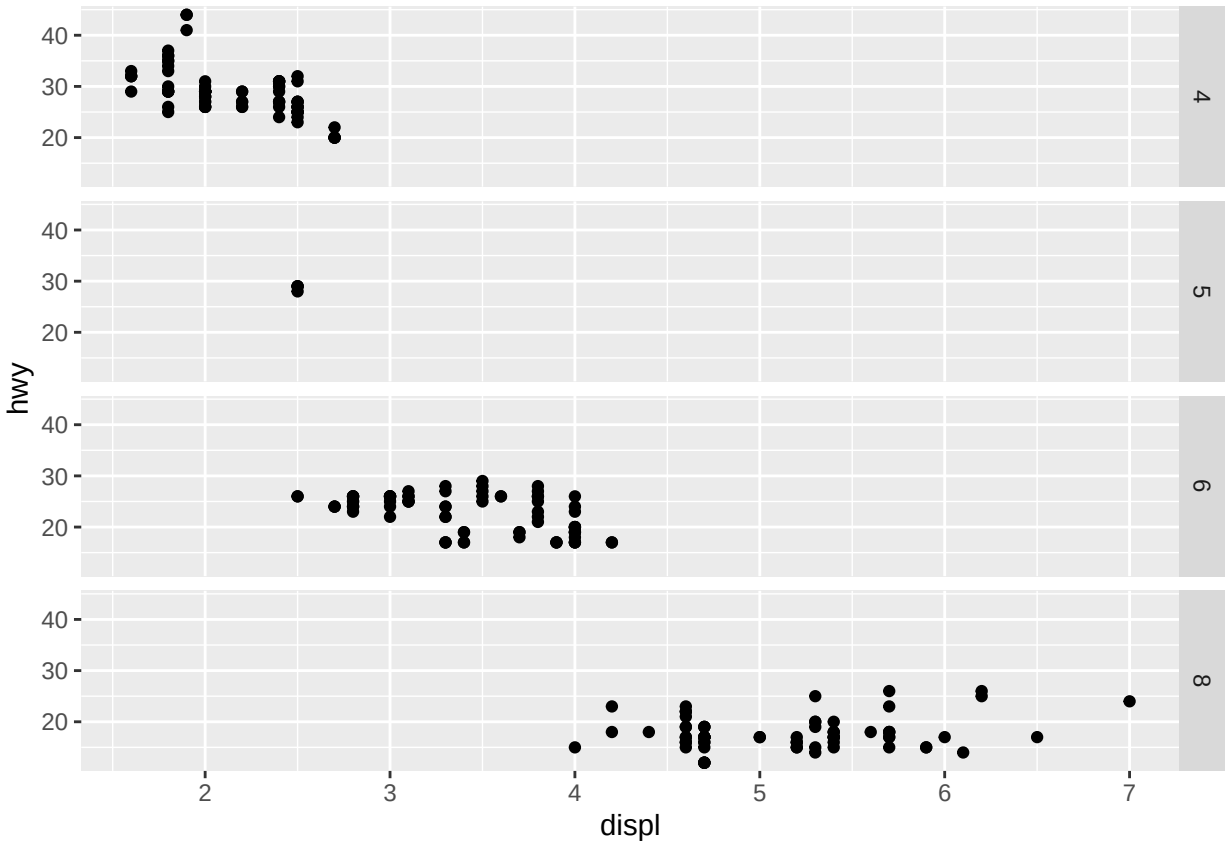
```
ggplot(data=mpg,aes(x=displ,y=hwy))+
  geom_point()+
  facet_grid(.~cyl)
```



Column of Charts

Simply changing the facet_grid from .~cyl to cyl~. will generate a column of charts

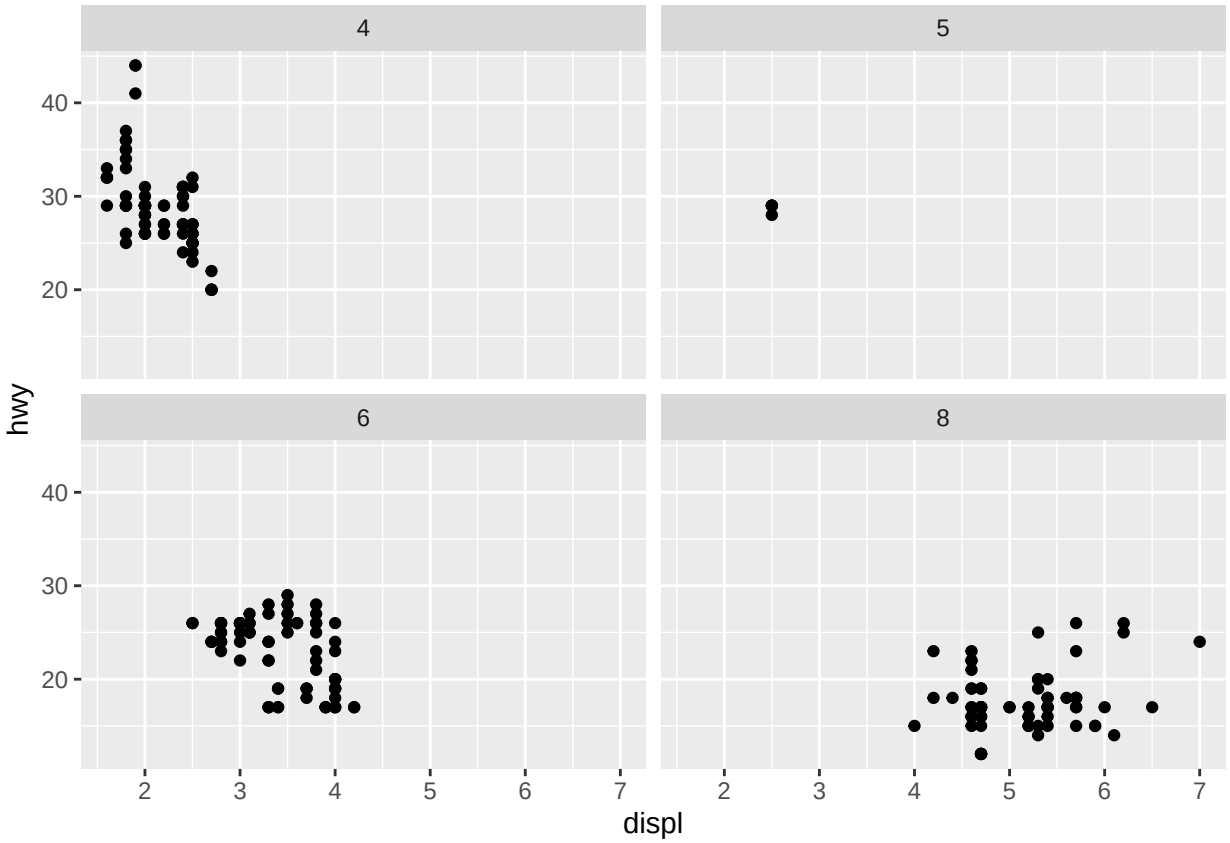
```
ggplot(data=mpg,aes(x=displ,y=hwy))+
  geom_point()+
  facet_grid(cyl~.)
```



Wrapping Charts

facet_wrap() will organize charts left to right and then start again in the next line from left to right. It is like a one-dimensional ribbon wrapped in two-dimensions.

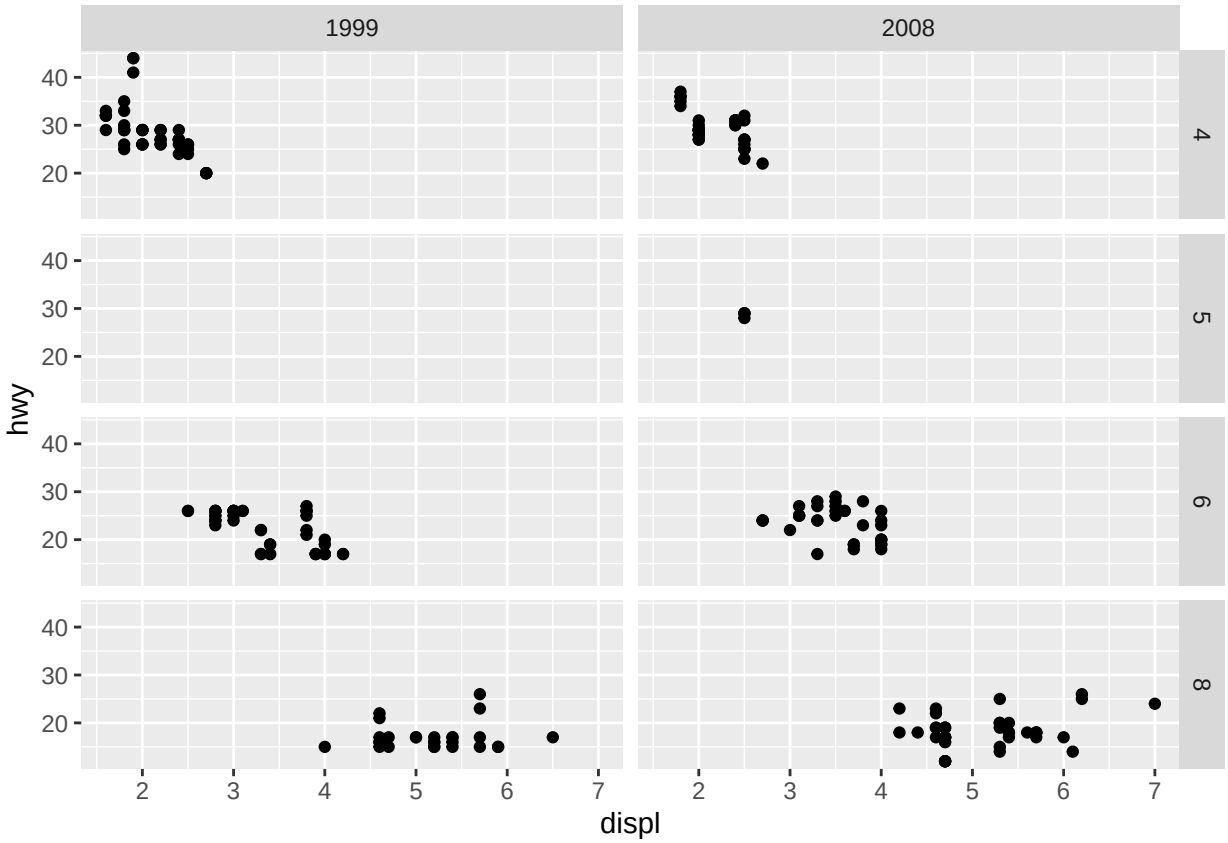
```
ggplot(data=mpg,aes(x=displ,y=hwy))+
  geom_point()+
  facet_wrap(~cyl)
```



Grid

Lastly, we can facet two different variables by including them as x and y axes in facet_grid(). In the example below, we add a fourth variable, year to facet_grid().

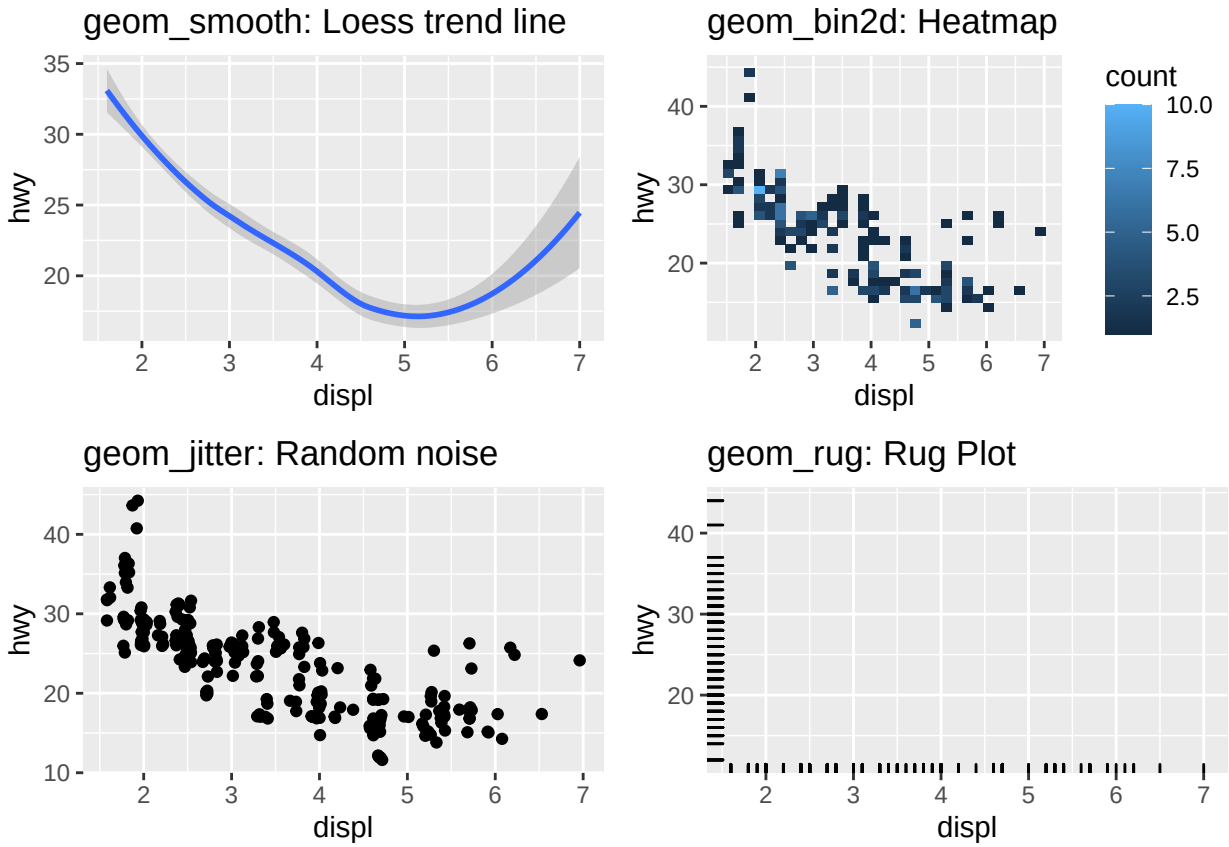
```
ggplot(data=mpg,aes(x=displ,y=hwy))+
  geom_point()+
  facet_grid(cyl~year)
```



Exercise: Try replicating the facet charts above by using drv and class instead of cyl and year.

More Charts

A geom is the geometrical object that a plot uses to represent data. Thus, to create a plot not created above, find the geom corresponding to it and add it as a layer to ggplot. For e.g., consider see the effect of passing dif-



ferent geom's.

Resources

Visualizing data is an excellent way to familiarize oneself with the data and spot patterns in it. We examined the use of `ggplot2` for constructing quick and dirty visuals but it can also be used for generating attractive presentation-ready charts. Here are some helfpul references:

- . Google Groups, and Stack Overflow
- . Cheatsheet
- . ggplot2 functions
- Books: ggplot2: Elegant Graphics for Data Analysis by Hadley Wickham, Graphical Data Analysis with R by Antony Unwin, and R Graphics Cookbook.

In closing, some great words by John Tukey:

"A picture is not merely a thousand words, it is much more likely to be scrutinized than words are to be read"

"Greatest value of a picture is when it forces us to notice what we never expected to see."