Lecture 7: More Data Visualization with ggplot2

STAT 450, Fall 2021

`ggplot2` references:

https://ggplot2.tidyverse.org/

https://ggplot2.tidyverse.org/reference/index.html

Diamonds Data Set

The `diamonds` data set comes with `ggplot2` and contains information about over 50,000 diamonds, including the `price`, `carat`, `cut`, `color`, and `clarity`.

```
library(ggplot2)
diamonds
```

```
## # A tibble: 53,940 x 10
##    carat cut       color clarity depth table price     x     y     z
##    <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
##  1  0.23 Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
##  2  0.21 Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
##  3  0.23 Good      E     VS1      56.9    65   327  4.05  4.07  2.31
##  4  0.29 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
##  5  0.31 Good      J     SI2      63.3    58   335  4.34  4.35  2.75
##  6  0.24 Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
##  7  0.24 Very Good I     VVS1     62.3    57   336  3.95  3.98  2.47
##  8  0.26 Very Good H     SI1      61.9    55   337  4.07  4.11  2.53
##  9  0.22 Fair      E     VS2      65.1    61   337  3.87  3.78  2.49
## 10  0.23 Very Good H     VS1      59.4    61   338  4     4.05  2.39
## # ... with 53,930 more rows
```

To read about this data set in the help menu type

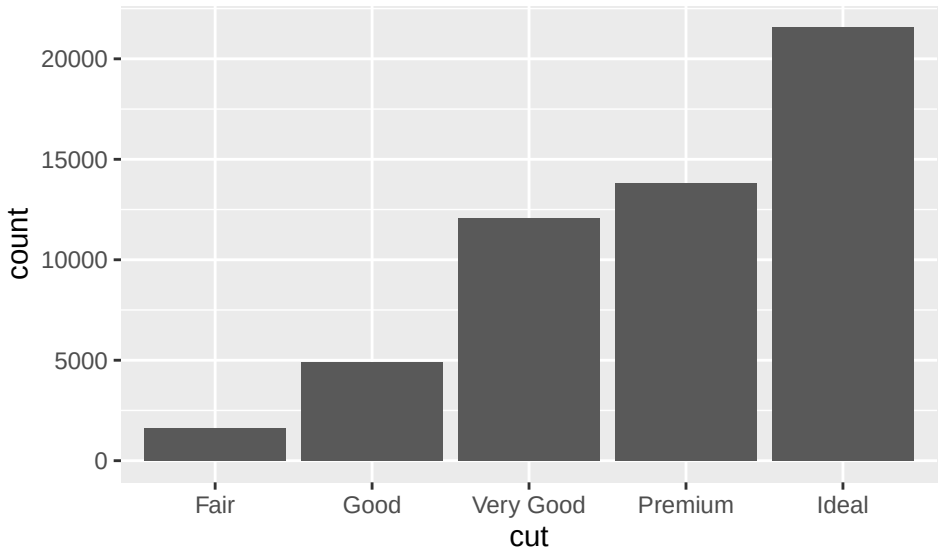
```
help(diamonds)
```

Exercise: What do you think is the meaning of the label `<ord>` for the `cut` column?

Bar Plots

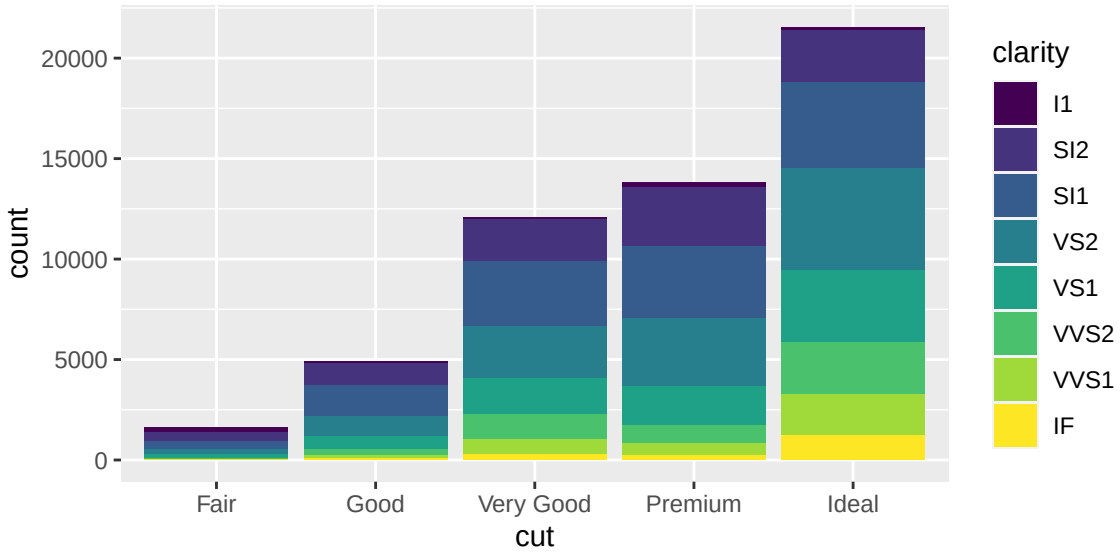
Here is a bar plot that shows the counts for each category of `cut`. We see that there are more diamonds with high quality cuts than with low quality cuts.

```
ggplot(data = diamonds) +
  geom_bar(aes(x = cut))
```



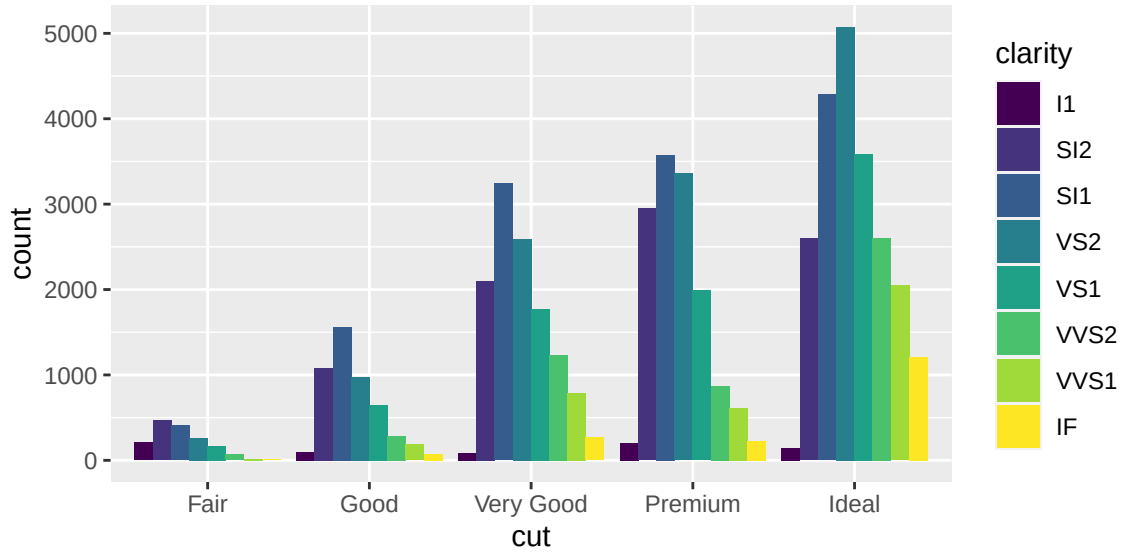
The following code creates a **stacked bar plot**, with stacks corresponding to the categorical variable `clarity`.

```
ggplot(data = diamonds) +
  geom_bar(aes(x = cut, fill = clarity))
```



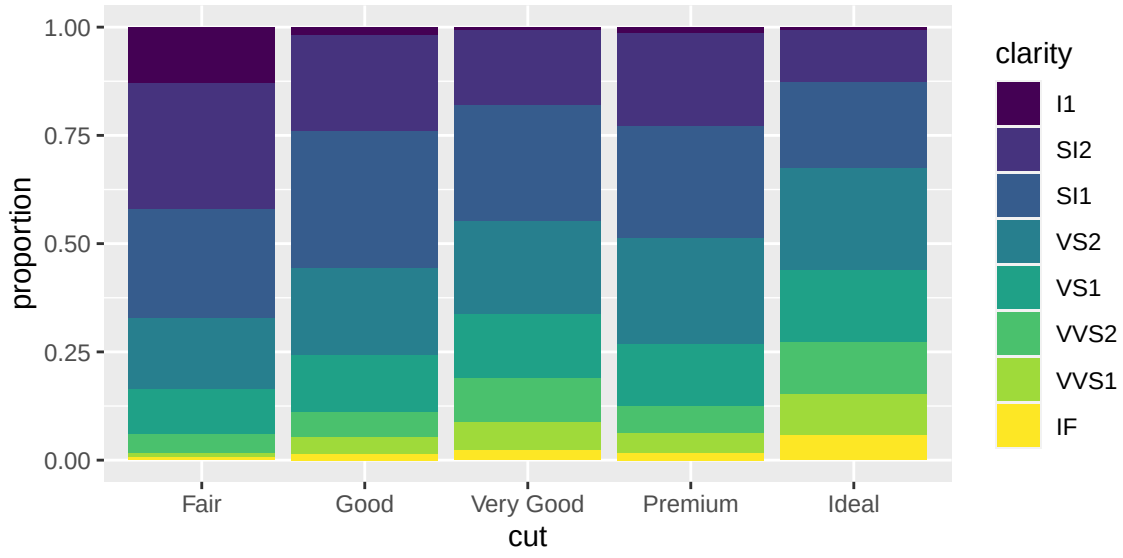
Setting `position = "dodge"` places the bars side-by-side instead of stacked on top of each other.

```
ggplot(data = diamonds) +
  geom_bar(aes(x = cut, fill = clarity), position = "dodge")
```



Setting `position = "fill"` lets us see the proportions of each diamond clarity, for each category of `cut`.

```
ggplot(data = diamonds) +
  geom_bar(aes(x = cut, fill = clarity), position = "fill") +
  ylab("proportion")
```



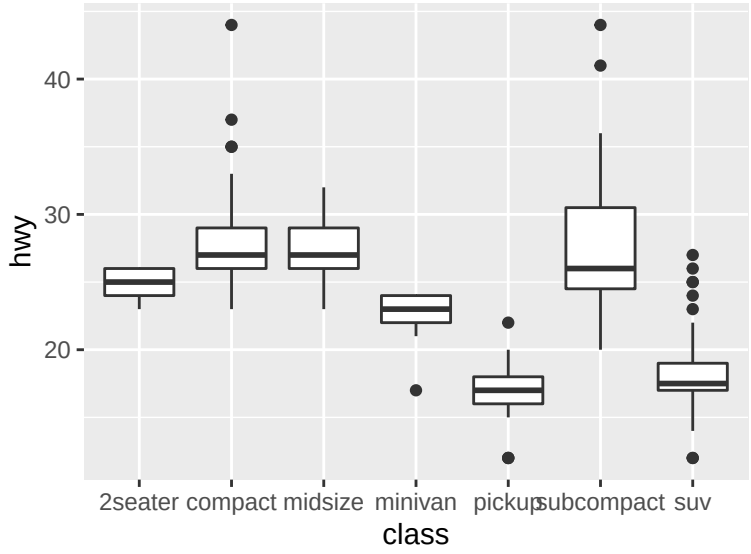
Coordinate Systems

The default coordinate system for `ggplot2` is the Cartesian coordinate system (plotting on the x and y axes). There are a number of other coordinate systems that can be useful, particularly when displaying maps.

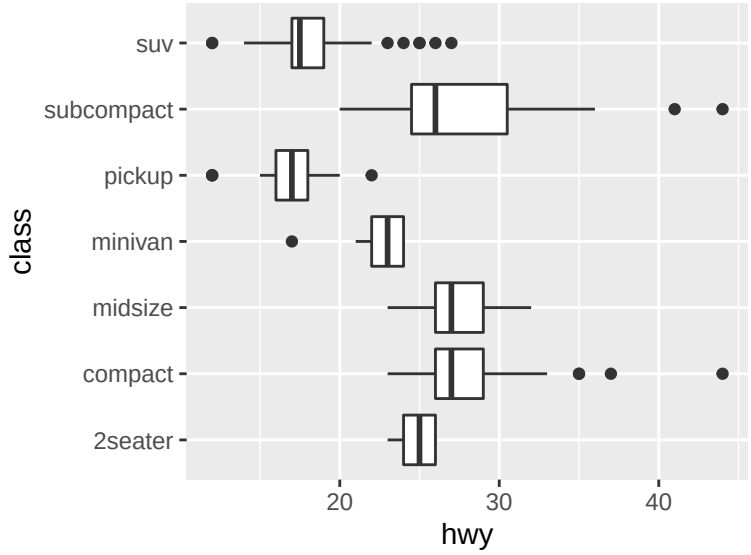
coord_flip()

Switches the x and y axes. This is useful when category names overlap on the x-axis.

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




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



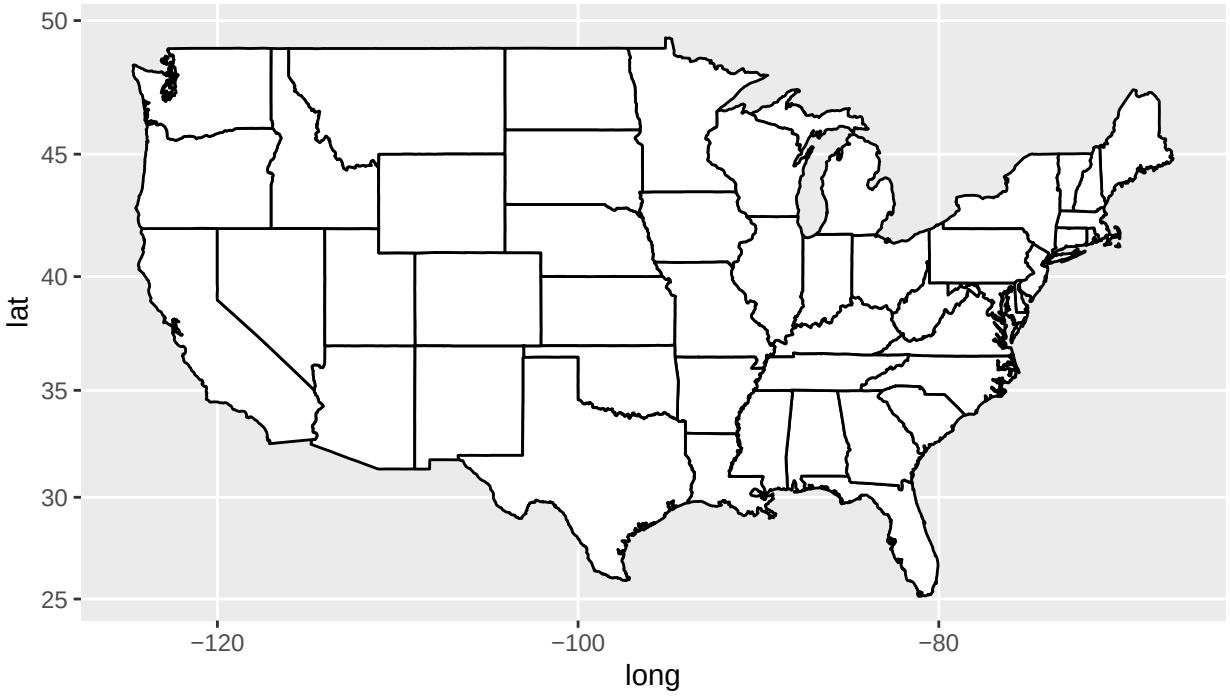
coord_map()

Approximates the correct aspect ratio for maps.

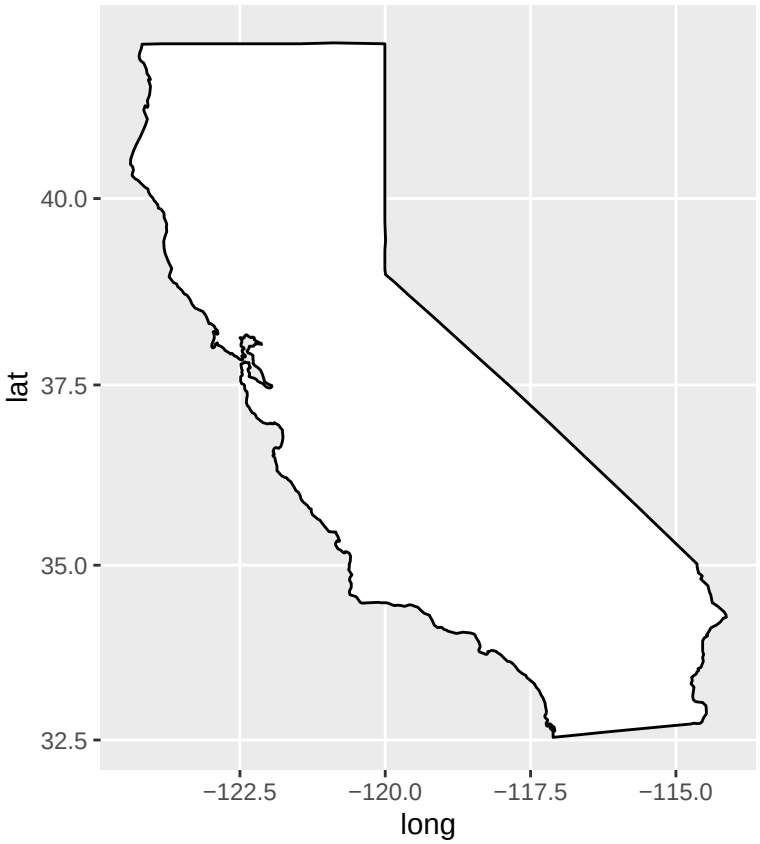
```
library(maps)
library(mapproj)
```

```
states <- map_data("state")
```

```
ggplot(states, aes(long, lat, group = group)) +
  geom_polygon(fill = "white", color = "black") +
  coord_map()
```



```r
ca <- map_data("state", "california")
ggplot(ca, aes(long, lat)) +
  geom_polygon(fill = "white", color = "black") +
  coord_map()
```



Exercises:

- Make a map of California without including `coord_map()`. How does the map visualization change? Is it better or worse?
- Make a map of another state of your choosing.

Plot of epicenters for earthquakes, magnitude 2.5 and higher, occurring in Southern California during the months of June and July 2019.[1] The Ridgecrest earthquakes occurred during this time period.

```
socal_quakes2019 <- read.csv("https://ericwfox.github.io/data/socal_quakes2019.csv")
```

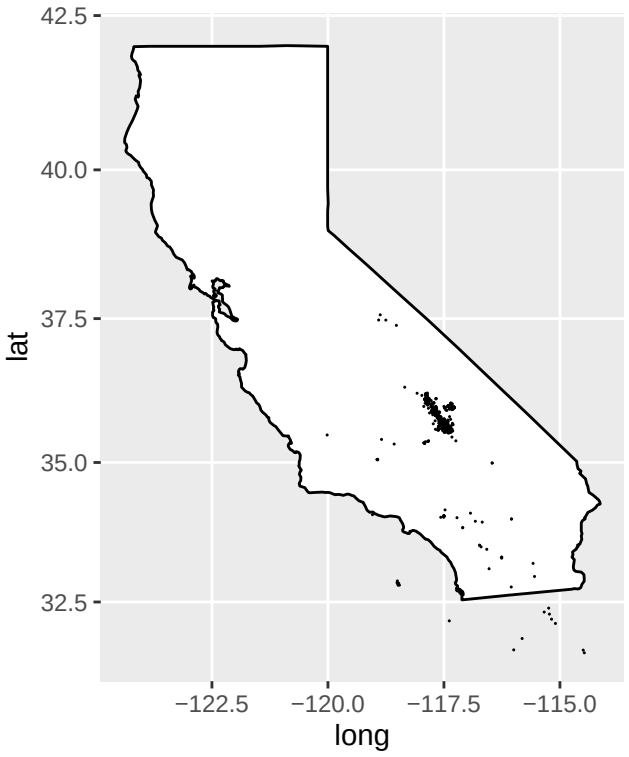
```
head(socal_quakes2019)
```

```
##         DATE    UTC_TIME  MAG      LAT       LON DEPTH
## 1 2019/06/02 01:41:39.42 2.63 34.05350 -117.4990   3.2
## 2 2019/06/02 02:18:53.08 2.61 34.05200 -117.4998   2.8
## 3 2019/06/02 02:19:48.58 3.10 34.04833 -117.5042   4.2
## 4 2019/06/02 20:01:55.34 2.55 33.84150 -117.0943  10.3
## 5 2019/06/02 20:04:55.50 3.06 33.84017 -117.0968  10.8
## 6 2019/06/02 23:36:36.03 3.21 34.05250 -117.5012   3.4
```

```
dim(socal_quakes2019)
```

```
## [1] 2067    6
```

```
ca <- map_data("state", "california")
ggplot(ca, aes(long, lat)) +
  geom_polygon(fill = "white", color = "black") + coord_map() +
  geom_point(data=socal_quakes2019, aes(LON, LAT), shape = 16, size=0.2)
```



[1]Data source: http://service.scedc.caltech.edu/eq-catalogs/date_mag_loc.php