The pipe operator %>%

- A common pattern in R is to perform a series of operations on a dataset, where each operation is supposed to be performed on the result of the previous operation.
- We can think of this like a *pipeline* or *assembly* line, where each "station" along the way performs some task and then passes the result on to the next step.
- We can use the pipe operator to make the syntax for this a bit easier to digest.
- Example: We start with the counties dataset and compute a tally/count of the counties on each state. Then we want to sort the resulting values, then put them in a dotplot. Here are the two ways to do that, with and without the pipe operator:

```
# Without pipe:
dotplot(sort(tally(~state, data=counties)))
# With pipe:
tally(~state, data=counties) %>% sort() %>% dotplot()
```

- Syntactically, the following two general steps are the same:

```
f(x, ...)
x %>% f(...)
```

- Therefore the pipe symbol only works when object to the left of the pipe is what should go to the first spot within the function.
- This is NOT the case for functions like tally, which expect the dataset in the data= spot.