Intro to ggplot2

Contents

```
acitelli <- read.csv("acitelli.csv")
```

There are quite a few ways to make figures in R, we'll the popular package `ggplot2`. You can find a cheat sheet for `ggplot2` here. Be sure to install it first if you never have, or if you need to update it.

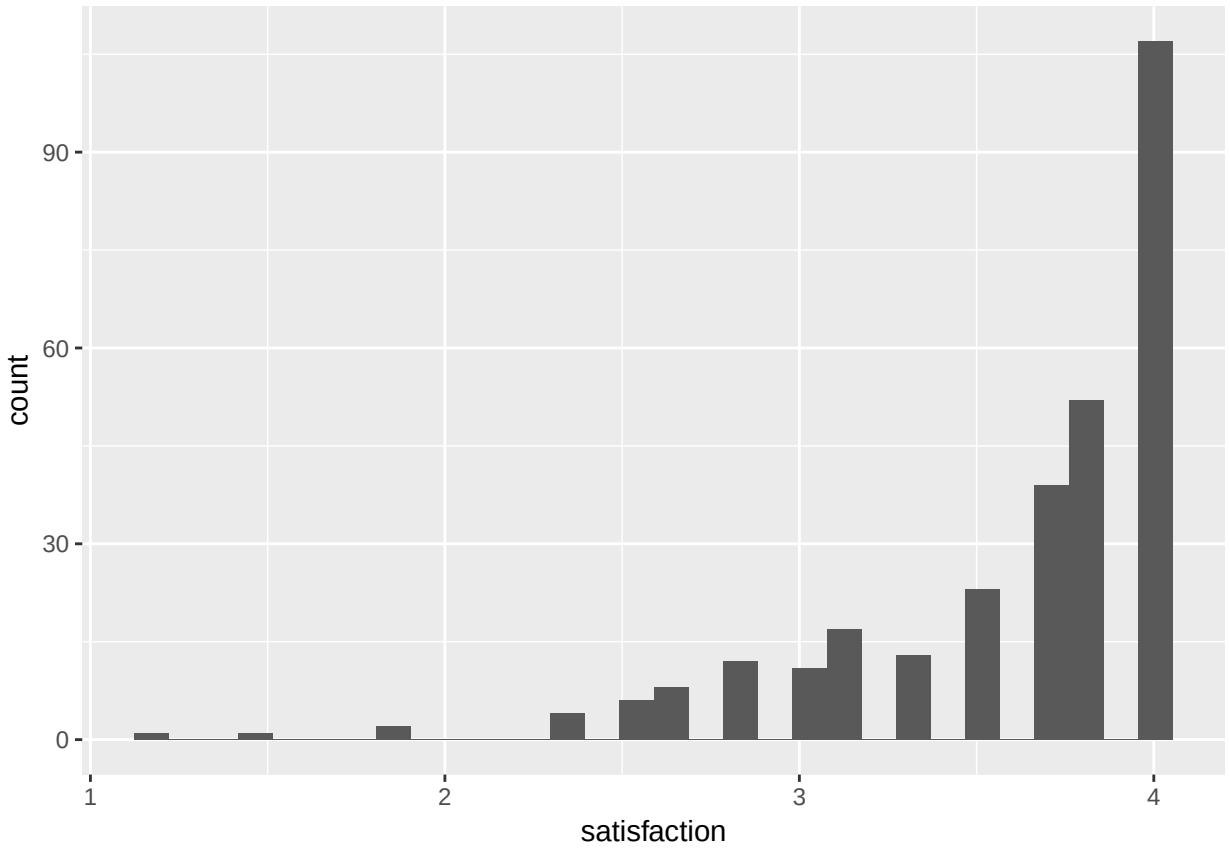
```
#install.packages("ggplot2")
library(ggplot2)
```

Visualizing Data with `qplot()`

Quick Histogram

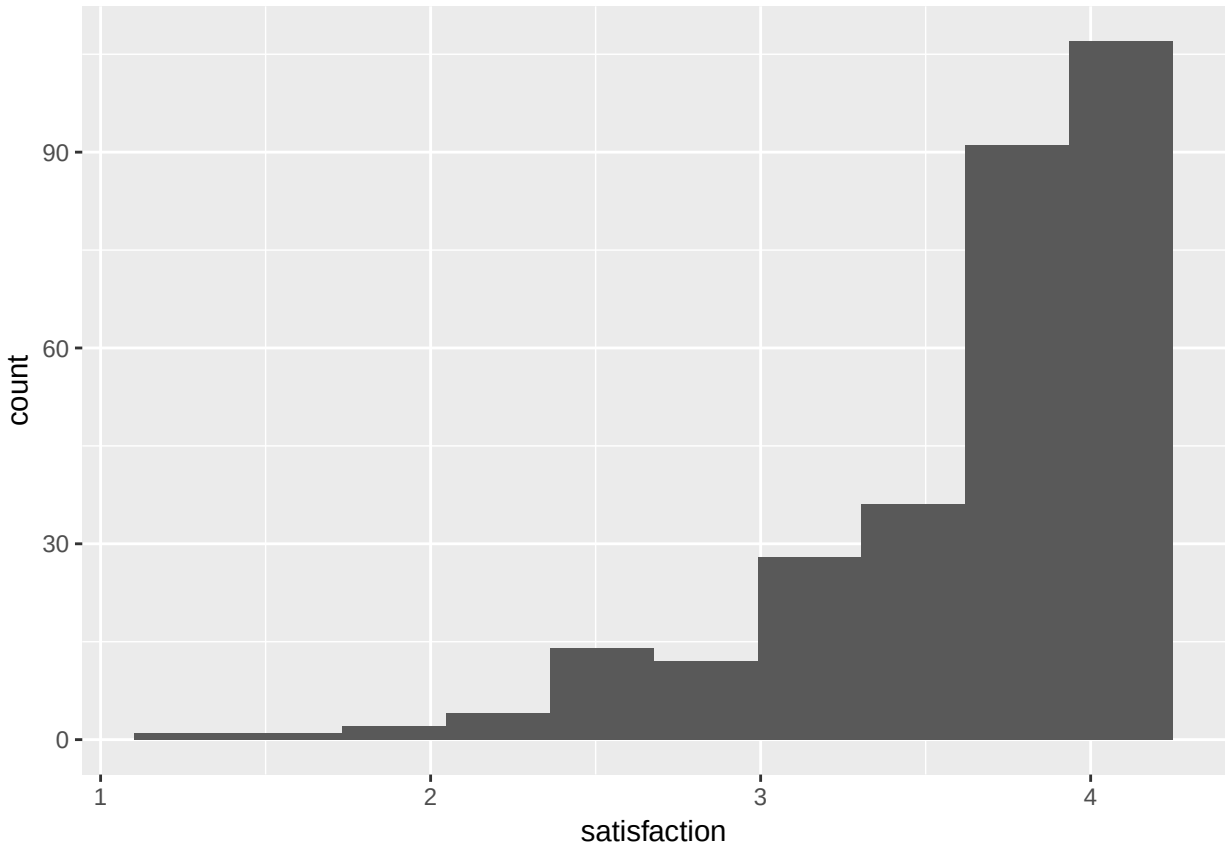
First, let's make a histogram for `satisfaction`. The easiest way to make a figure with `ggplot2` is with the `qplot()` function. This stands for *quick plot*. Notice in the code below that we did not specify anything about a histogram. `qplot()` guesses which type of plot we want based on the variable's type (i.e., integer, number, double, factor, character).

```
qplot(satisfaction, data = acitelli)
```



There are too many bins (it defaults to 30 bins), we can ask for a specific number by adding the `bins =` argument. Try playing around the bin number below to find the optimal plot. I put `10` in there as a placeholder.

```
#play around with different numbers of bins
qplot(satisfaction, data = acitelli, bins = 10)
```

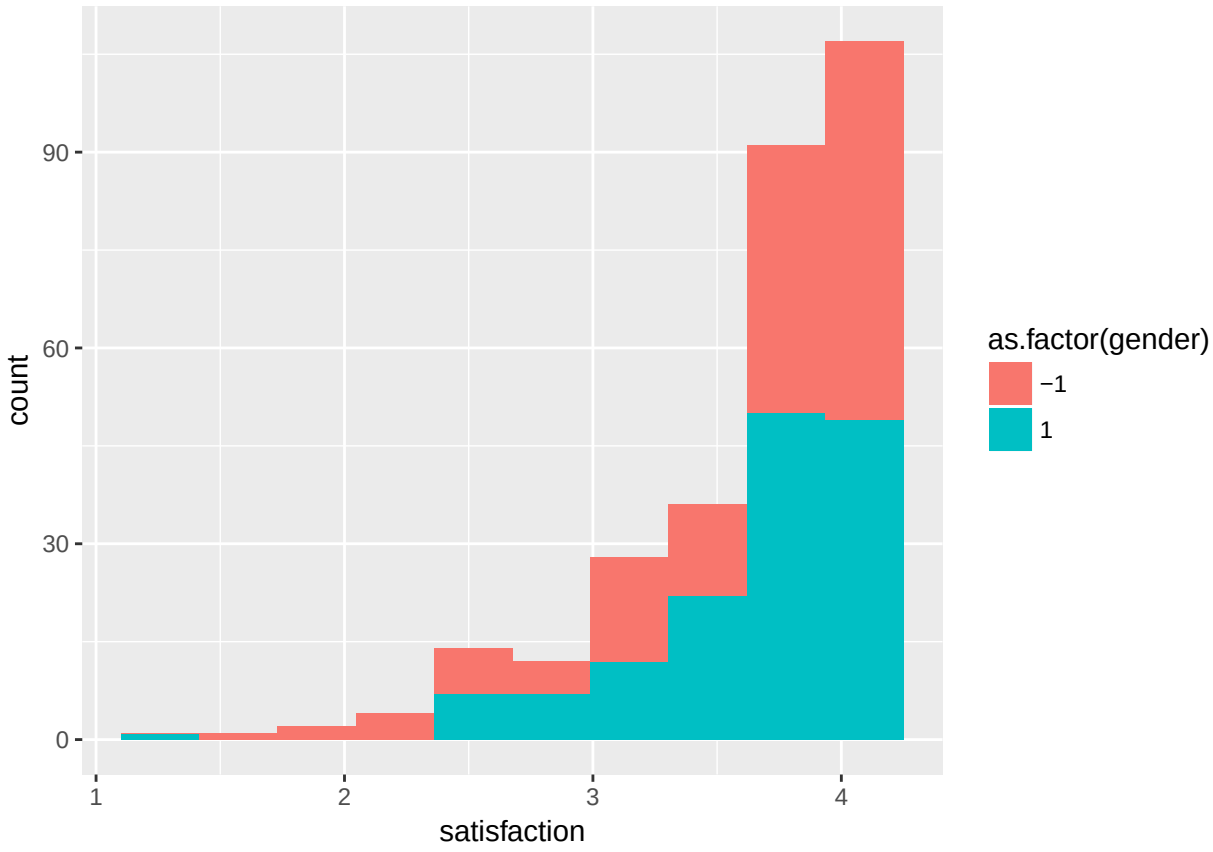


Remember that you can run the `?` code to see the help file for any function.

```
?qplot
```

We might want to see if the distributions are different for men and women. We can do this by mapping `Gender` to the fill aesthetic. Note that we could used `color =` if we want a hollow histogram. Try it out!

```
qplot(x = satisfaction, fill = as.factor(gender), data = acitelli, bins = 10)
```

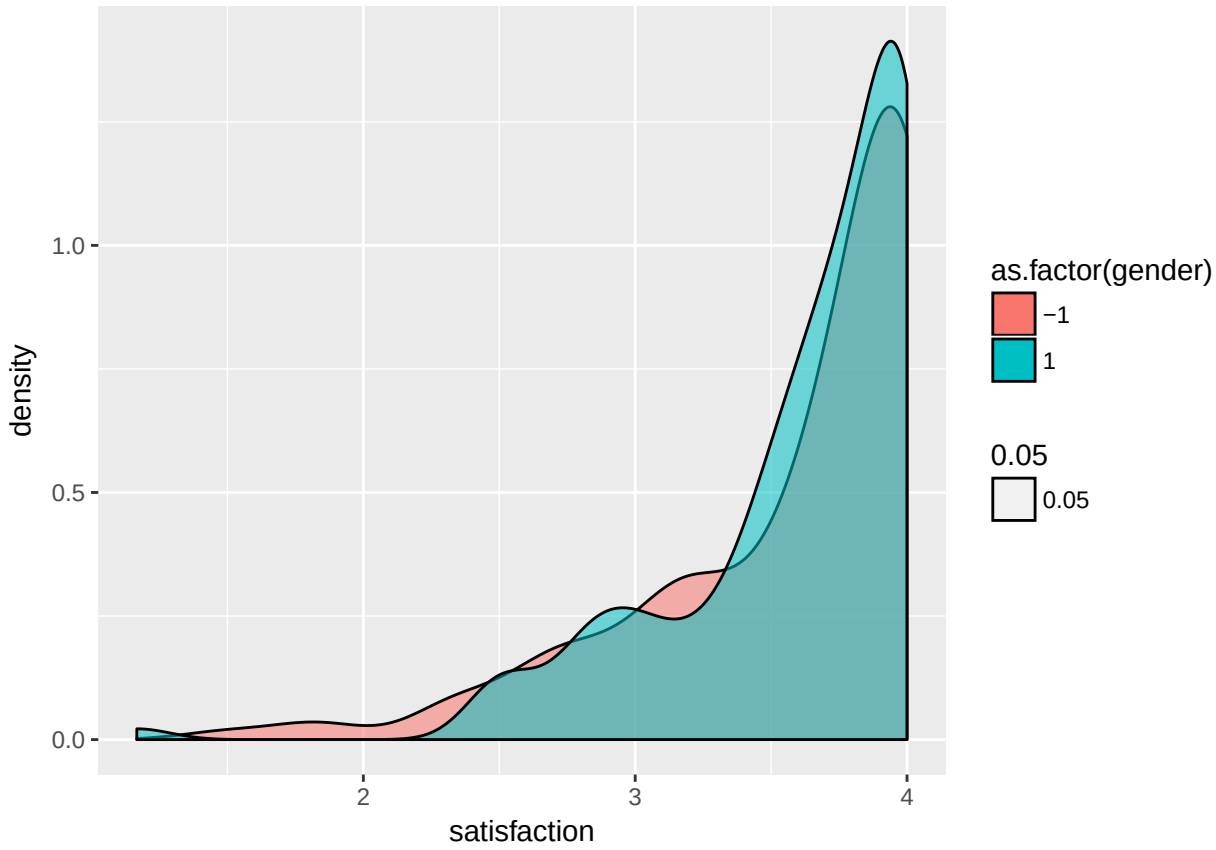


Why did we have to change the variable type of gender? What happens if you don't?

Quick Density Plot

An alternative to the histogram is the density plot. It displays a smoothed distribution and the area under the curve always sums to 1, thus, it's good for comparing two groups with different n's.

```
qplot(x = satisfaction, fill = as.factor(gender), data = acitelli, alpha = .05, geom = '
```



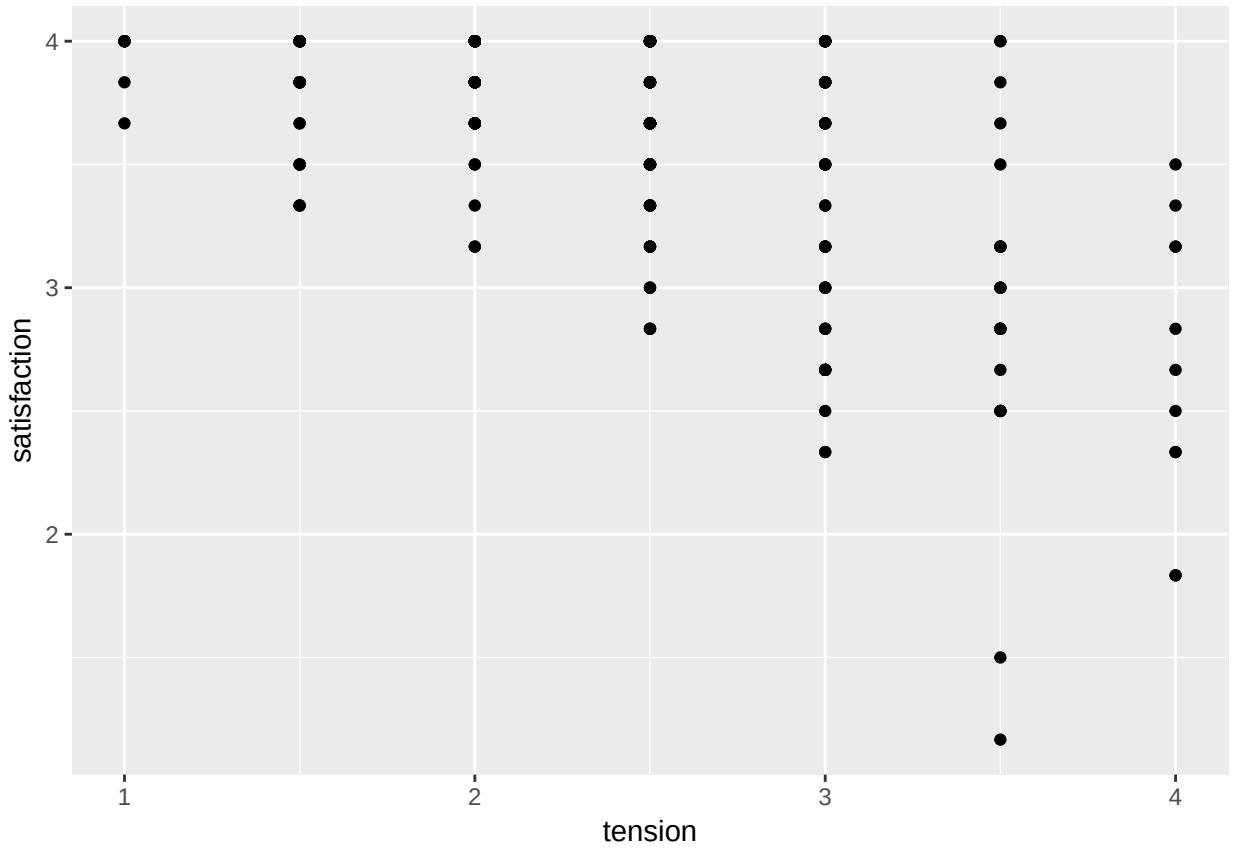
What would you say the `alpha =` argument is doing? What happens if you take it away? What if you change the value?

```
#play with alpha here
```

Quick Scatterplot

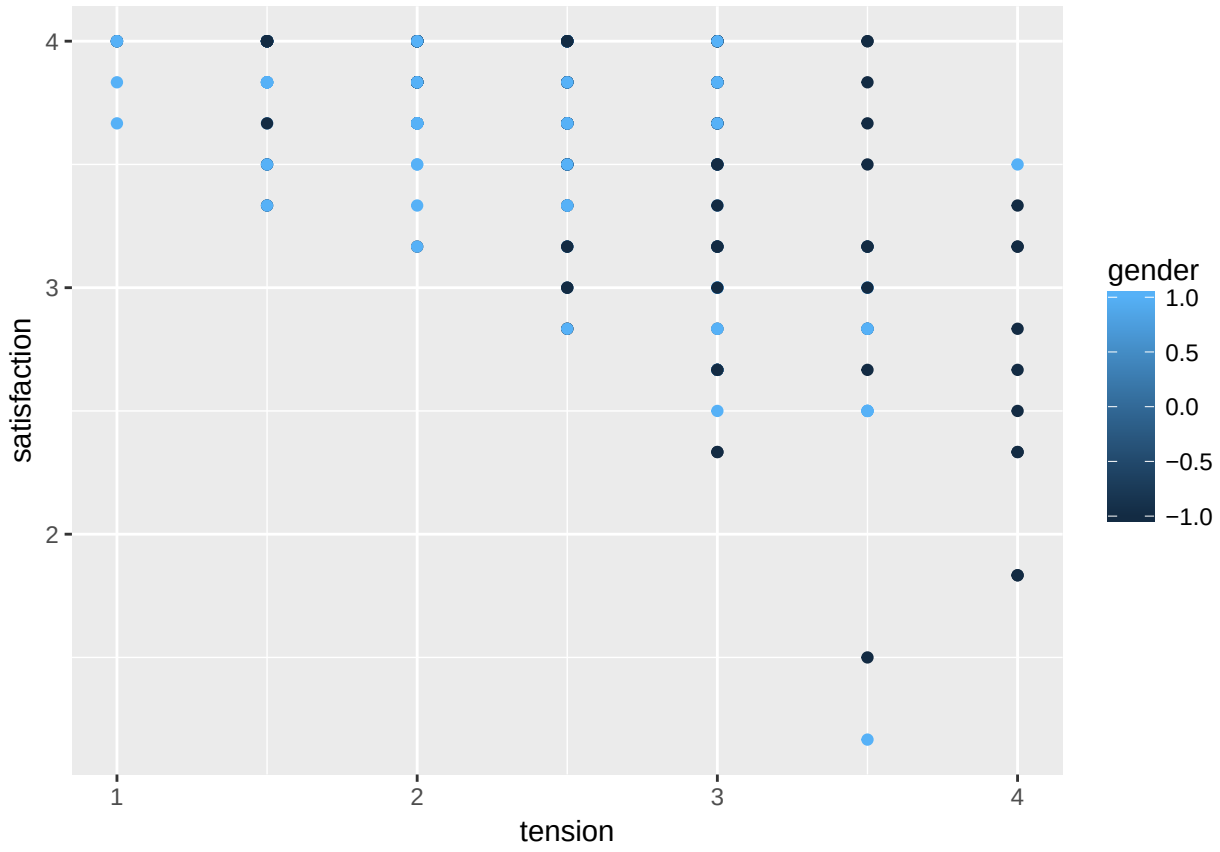
We might also want a scatter plot. Again,`qplot()` guesses what we want, but it's a good idea to specify which variable goes on the x-axis and which goes on the y-axis.

```
qplot(x = tension, y = satisfaction, data = acitelli)
```



We can even add a third variable, mapping it to color.

```
qplot(x = tension, y = satisfaction, color = gender, data = acitelli)
```

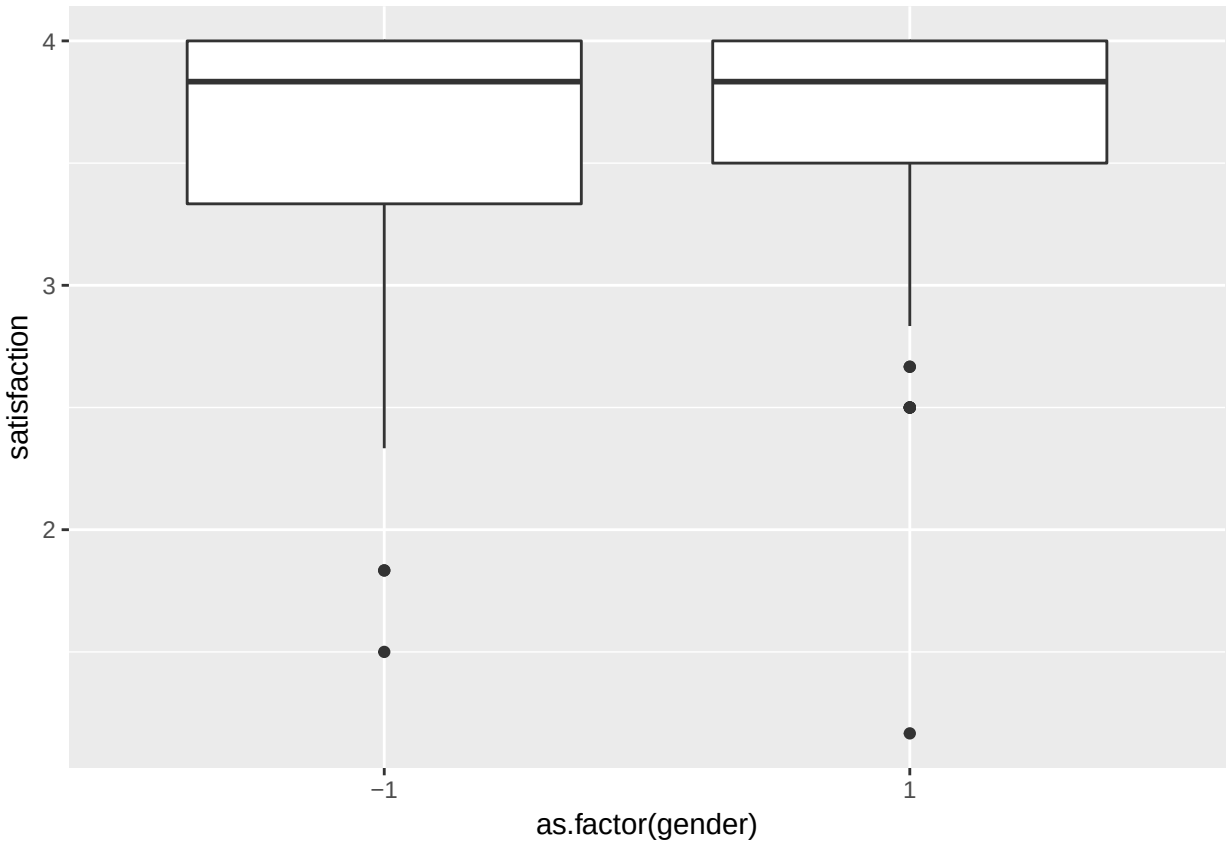


What's going on with the color scale for gender? Can you explain why this happend? What happens if you change the variable type of gender?

Quick Boxplots

We can ask for side-by-side boxplots when our x variable is categorical. In this case `qplot()` does **NOT** know what to do, so we tell it we want boxplots by adding the argument `geom = "boxplot"`.

```
qplot(y = satisfaction, x = as.factor(gender), data = acitelli, geom = "boxplot")
```



What would happen if you added `fill = as.factor(gender)` as an argument. Try it by adding a new chunk below.

ADD A NEW CHUNK HERE

Full Blown ggplots with `ggplot()`

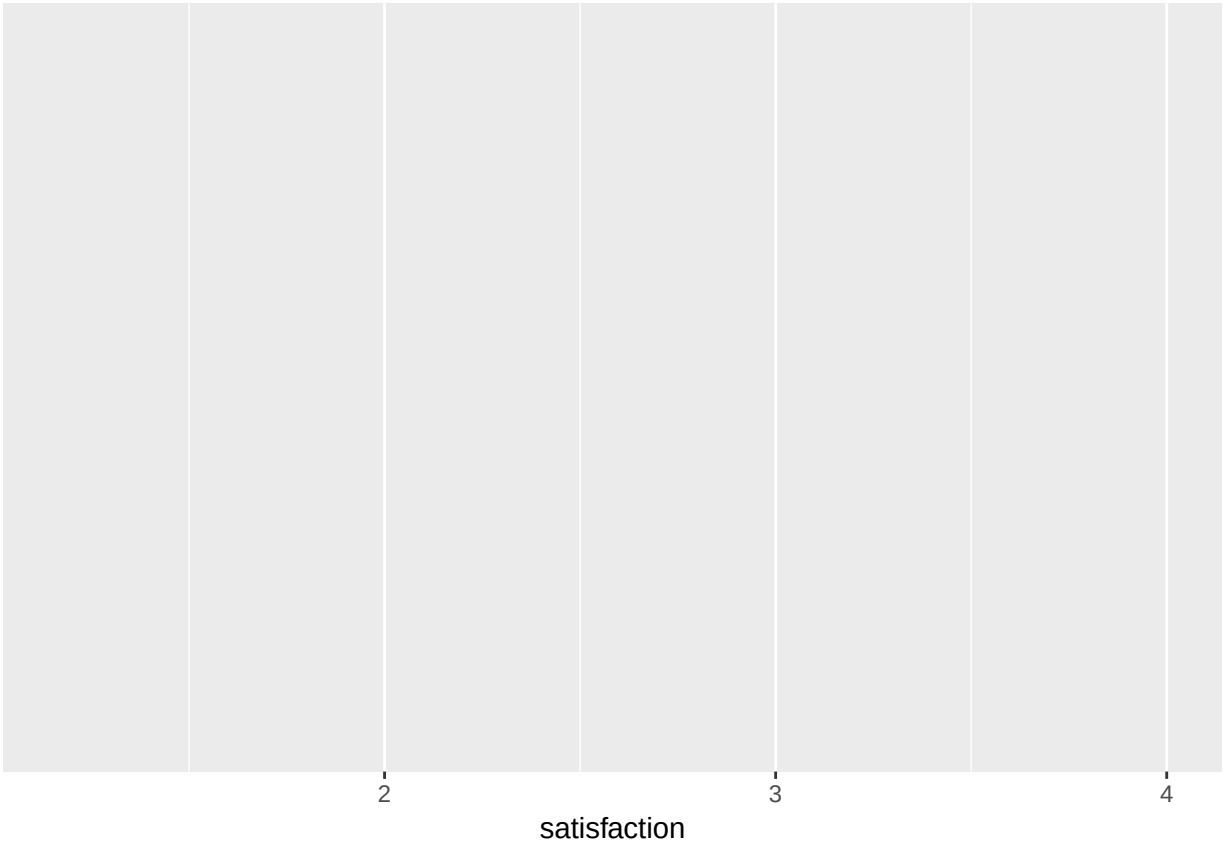
Histogram

For more complex figures we will need to move away from using the `qplot()` function in favor of the heavy duty `ggplot()` function. To get a sense of how `ggplot()` builds plots, first we will just run the empty function.

```
ggplot()
```

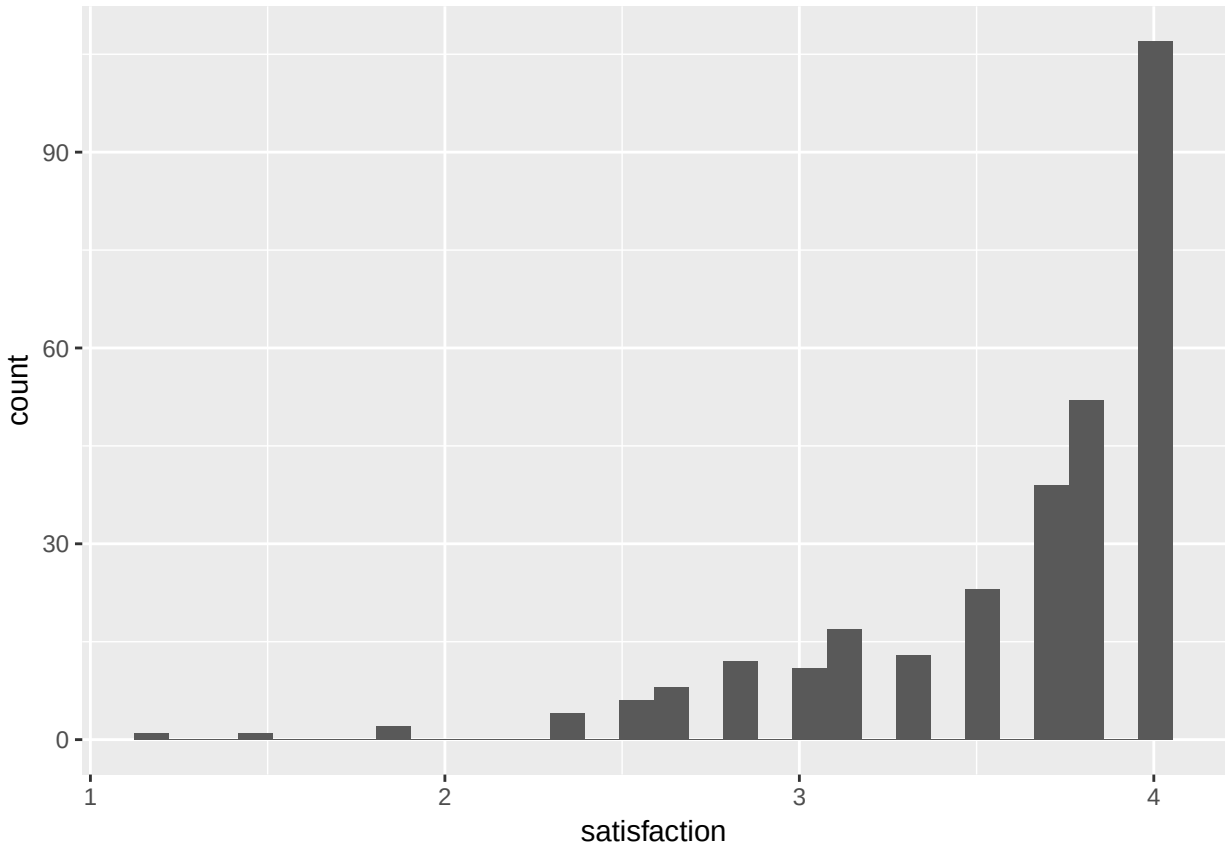
Next, we can add the data and start mapping variables to aesthetics.

```
ggplot(acitelli, aes(x = satisfaction))
```



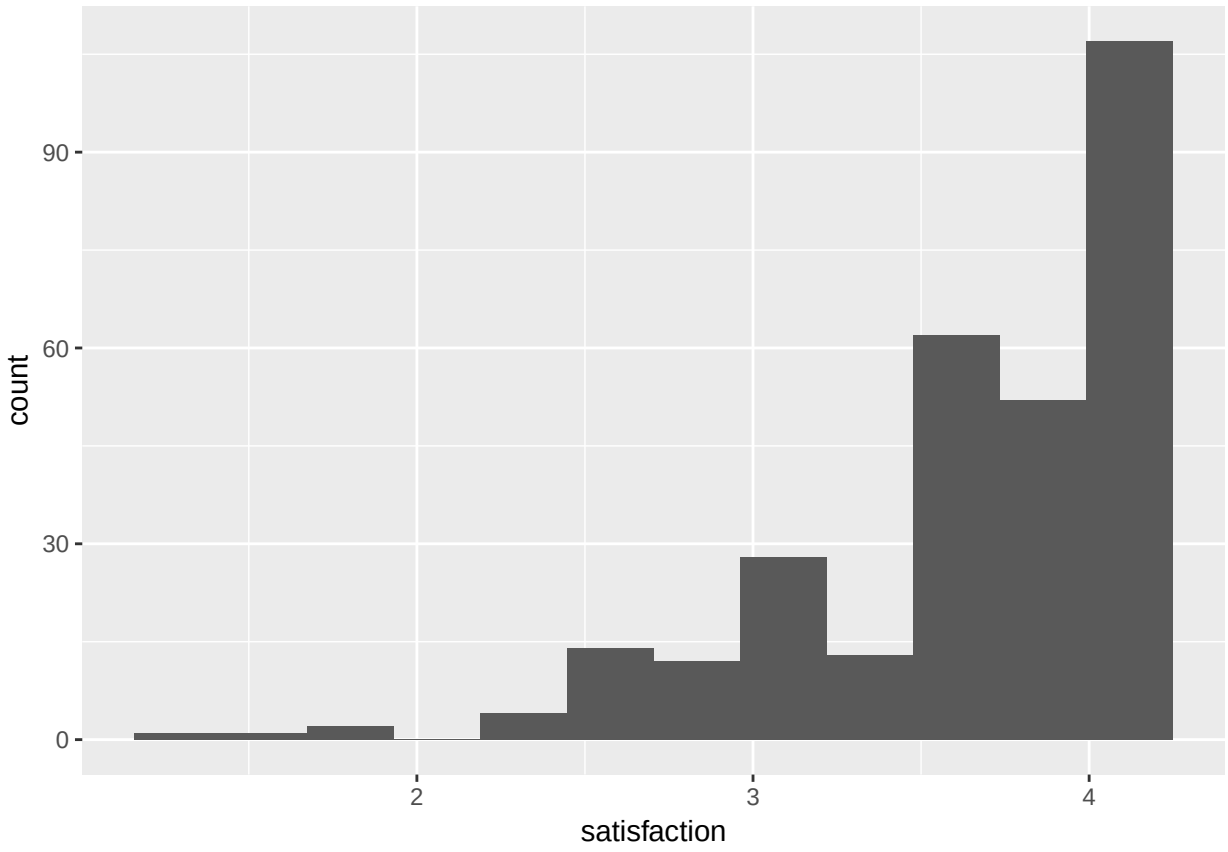
After we have specified aesthetic mappings, we can then add geoms. Notice that we make use of the `+` symbol with ggplots. The `+` needs to be on the right of each piece of the plot. We add a histogram with the `geom_histogram()` function.

```
ggplot(acitelli, aes(x = satisfaction)) +
  geom_histogram()
```



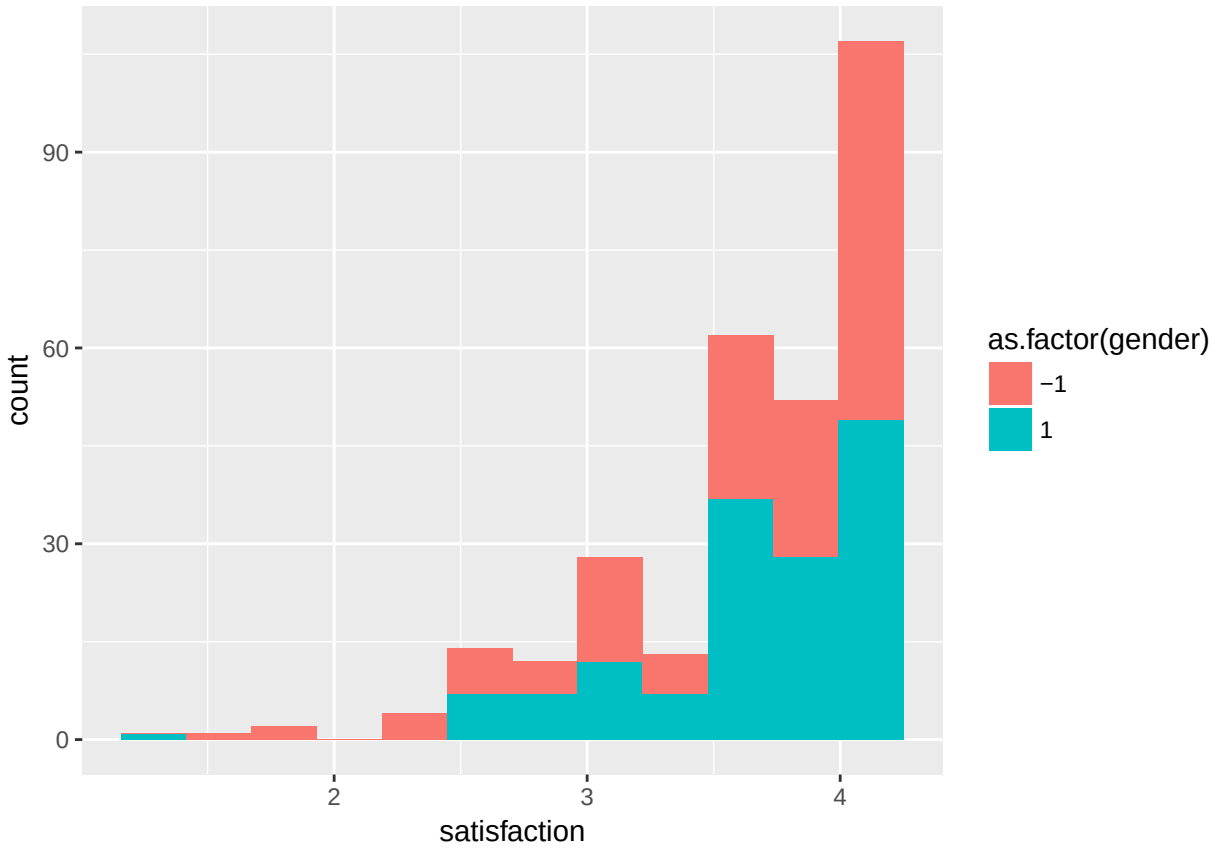
Note in the histogram above the y-axis are the counts of observations in each bin. There were some calculations involved in getting these counts. Counts are the default *statistic* when you ask for a histogram. We can change the number of bins by adding `bins =` inside of the `geom_histogram()` function.

```
ggplot(acitelli, aes(x = satisfaction)) +
  geom_histogram(bins = 12)
```



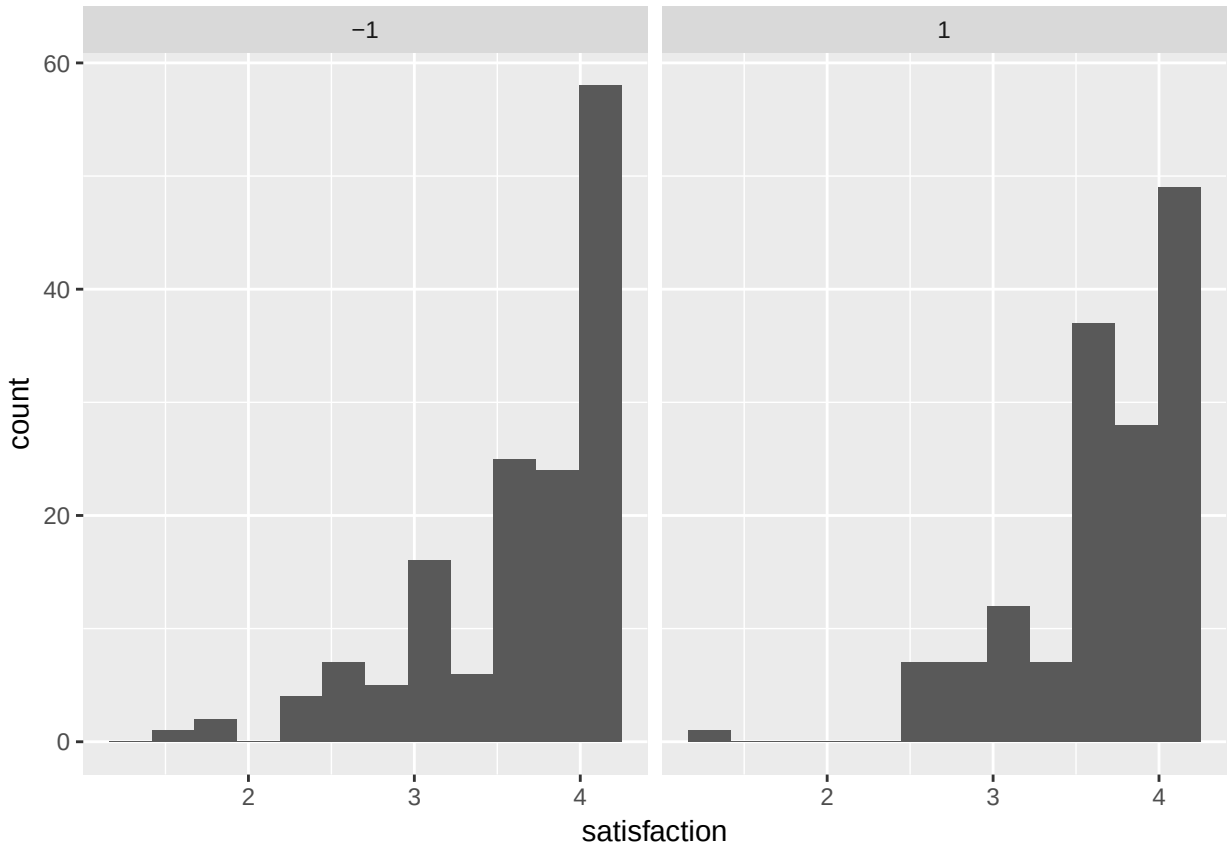
Just as before, if we want overlayed histograms, we can map gender to the fill aesthetic.

```
ggplot(acitelli, aes(x = satisfaction)) +
  geom_histogram(aes(fill = as.factor(gender)), bins = 12)
```



Alternatively, we can ask for separate facets for each level of the Gender variable with the `fact_wrap()` function. Notice that there is a `~` before Gender inside of this function.

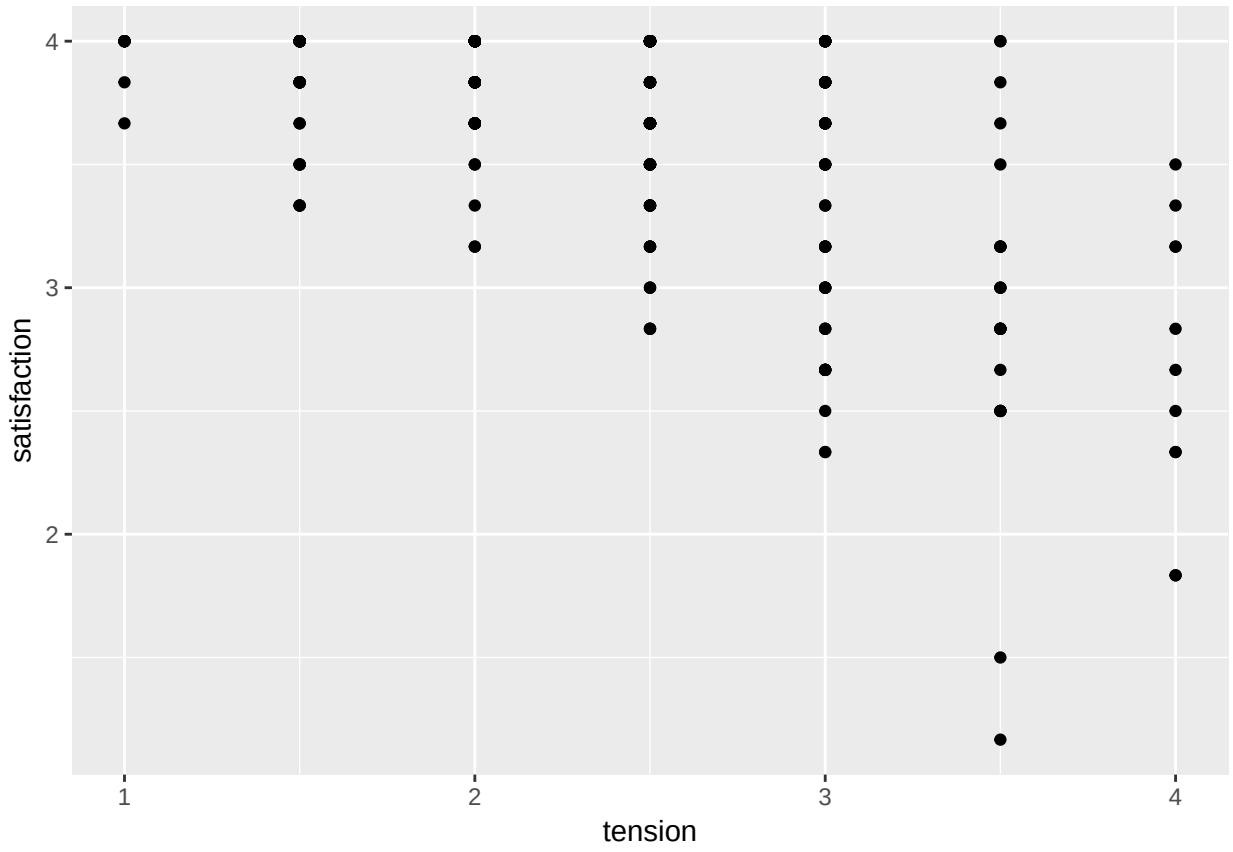
```
ggplot(acitelli, aes(x = satisfaction)) +
  geom_histogram(bins = 12) +
  facet_wrap(~as.factor(gender))
```



Scatter Plot

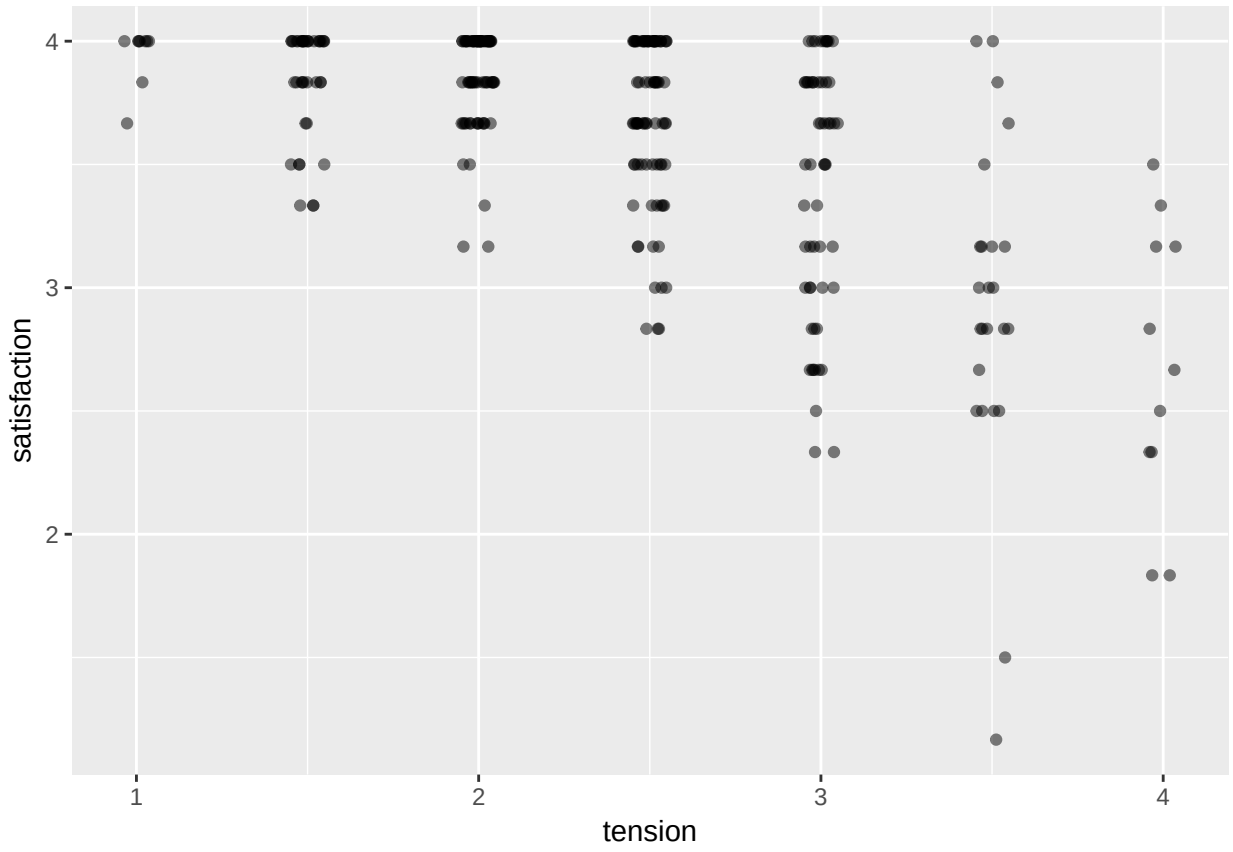
Next we'll make that scatter plot again. We'll map `tension` to the x-axis and `satisfaction` to the y-axis. Then we'll add `geom_point()`.

```
ggplot(acitelli, aes(x = tension, y = satisfaction)) +
  geom_point()
```



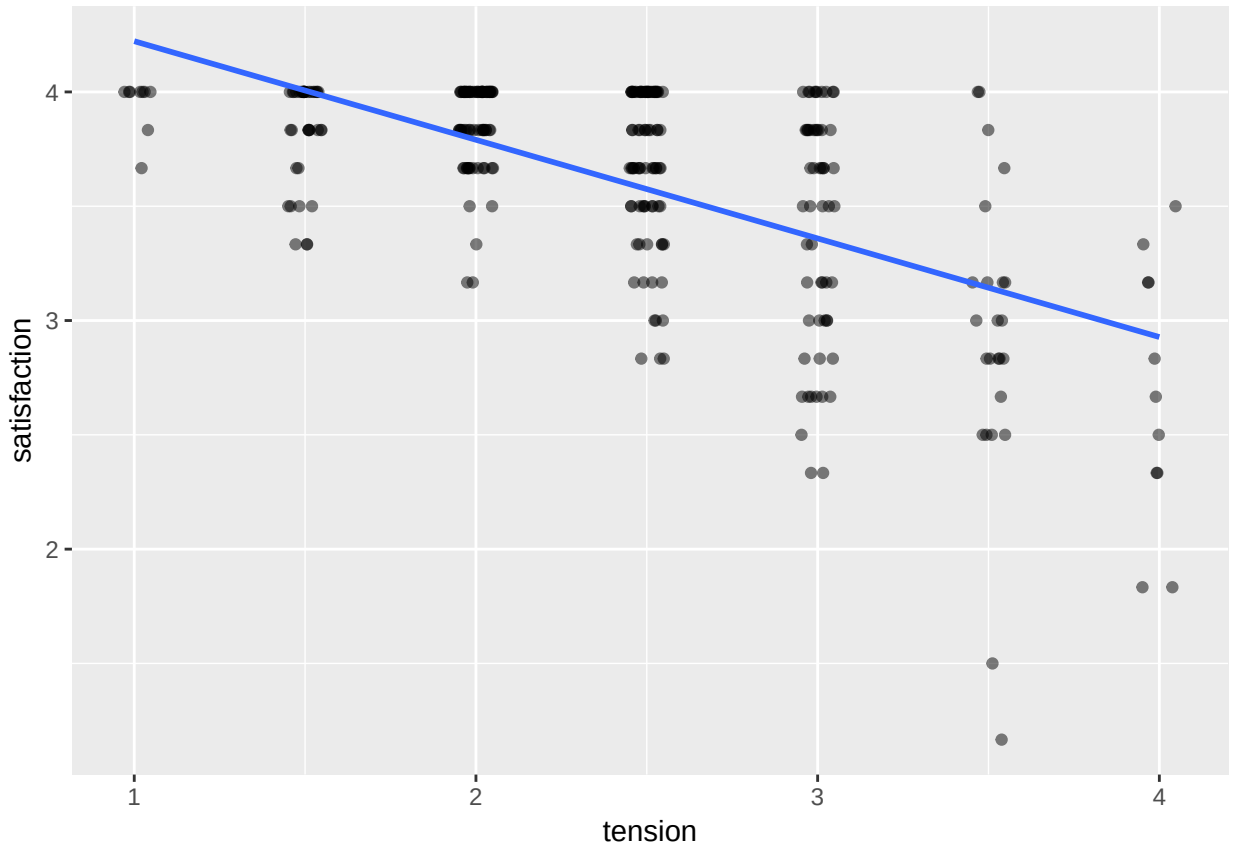
Why does it appear as though there is far less data than there really is? Check out the plot when we use `geom_jitter()`. What do you think `geom_jitter()` does?

```
ggplot(acitelli, aes(x = tension, y = satisfaction)) +
  geom_jitter(height = 0, width = .05, alpha = 0.5)
```



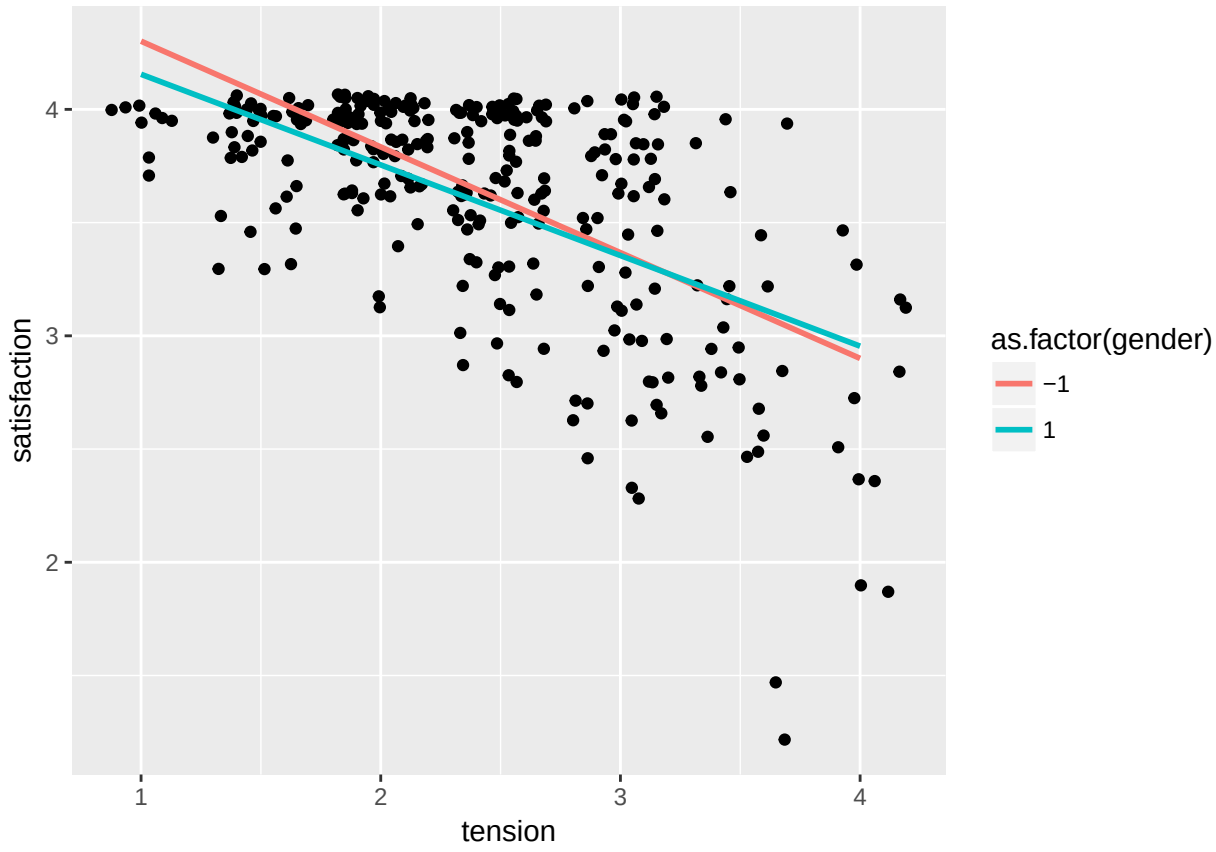
We can add more than one geom. To the jittered scatter plot we can add a least squares regression line with `geom_smooth()`. Inside of geom smooth we need to specific `method = "lm"`, the lm stand for *linear model*. We can also turn off the standard errors with se = 0.

```
ggplot(acitelli, aes(x = tension, y = satisfaction)) +
  geom_jitter(height = 0, width = .05, alpha = 0.5) +
  geom_smooth(method = "lm", se = 0)
```



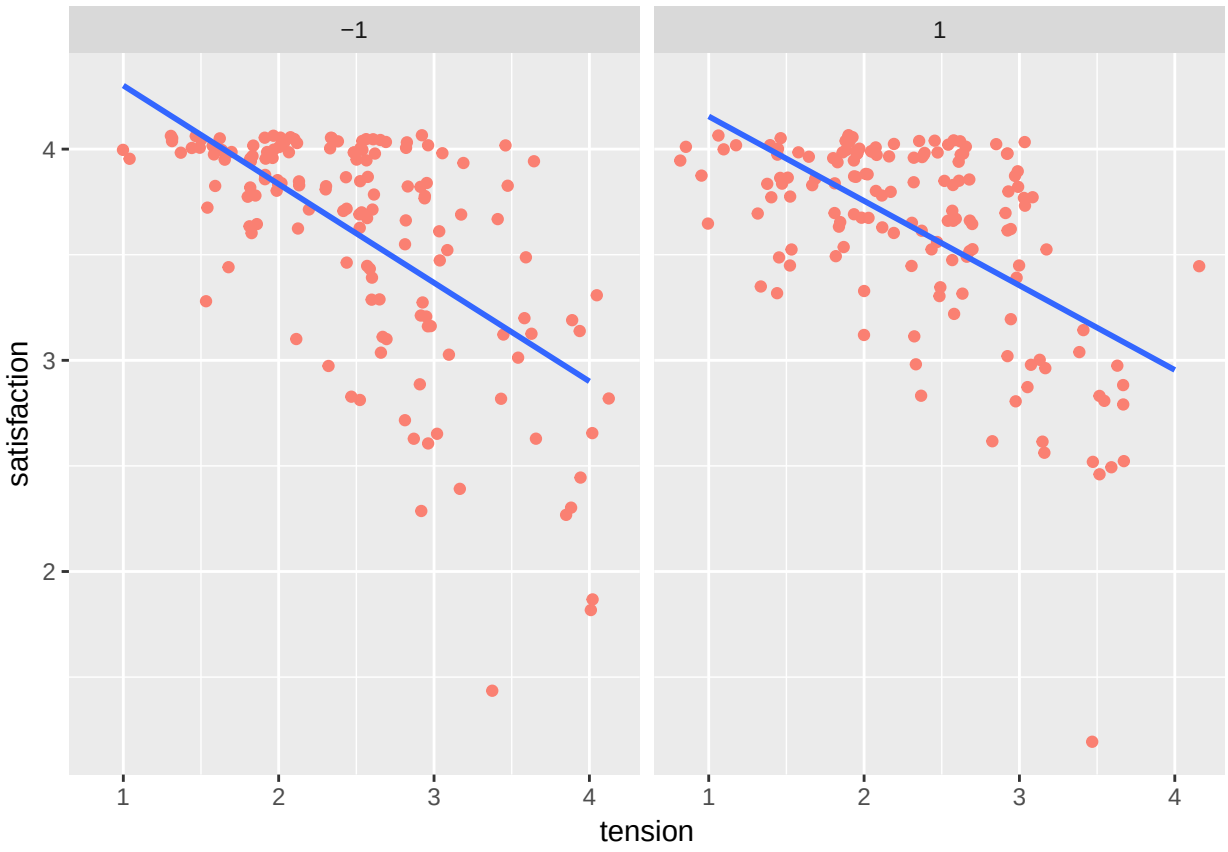
Again, we can map gender to the color aesthetic. Note: Here the mapping of gender to color is done in the `geom_smooth()` function—but what would happen if you moved it to the `aes()` function in the `ggplot()` function?

```
ggplot(acitelli, aes(x = tension, y = satisfaction)) +
  geom_jitter() +
  geom_smooth(aes(color = as.factor(gender)), method = "lm", se = 0)
```



Or use `facet_wrap()`. In the code below, what does `color = "salmon"` do? Could you change the regression lines to salmon?

```
ggplot(acitelli, aes(x = tension, y = satisfaction)) +
  geom_jitter(color = "salmon") +
  geom_smooth(method = "lm", se = 0) +
  facet_wrap(~as.factor(gender))
```

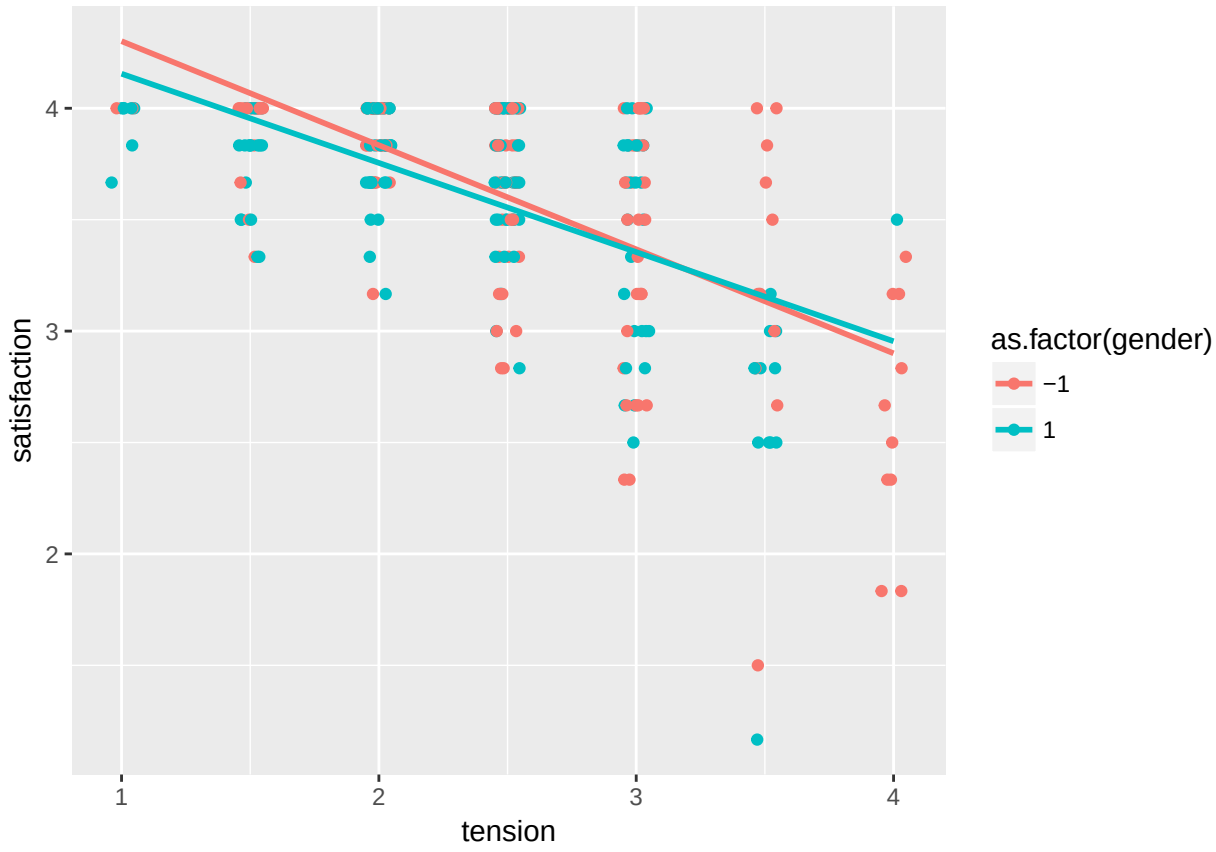


Labels and Colors

We can create a plot object with the <- symbol. Then to print the plot we'd need to run a line with the name of our plot.

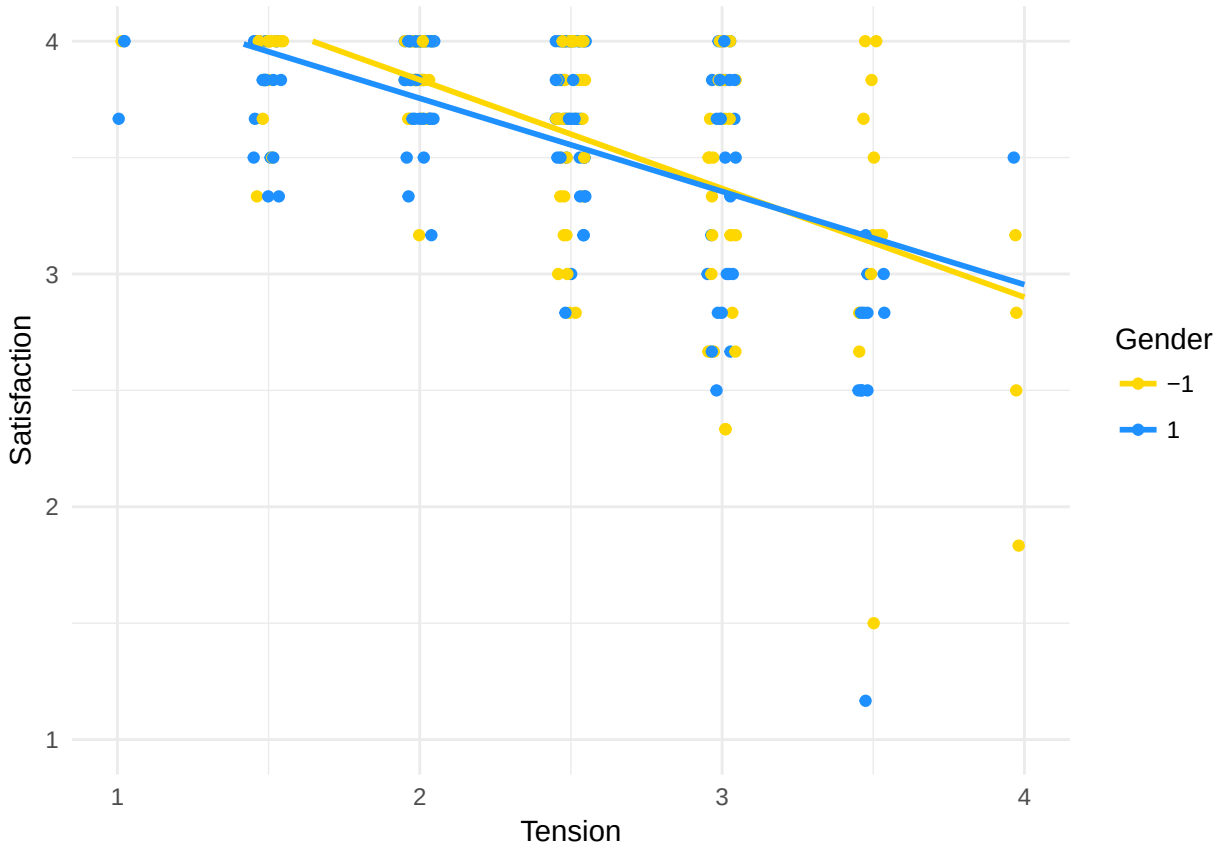
```
myplot <- ggplot(acitelli, aes(x = tension,
                               y = satisfaction,
                               color = as.factor(gender))) +
  geom_jitter(height = 0, width = .05) +
  geom_smooth(method = "lm", se = 0)

myplot
```



Then, we can add to that plot object. We can add x labels, x labels, change the colors, and the theme. There is much more that you can do with `ggplot2`!

```
myplot +
  xlab("Tension") +
  ylab("Satisfaction") +
  scale_color_manual(values = c("gold", "dodgerblue"),
                     name = "Gender") +
  ylim(1,4) +
  xlim(1,4) +
  theme_minimal()
```


Satisfaction
4
3
2
1
1
2
3
4
Tension
Gender
−1
1