Intro to ggplot2

Contents

```r
acitelli <- read.csv("/Users/randigarcia/Desktop/acitelli.csv", header=TRUE)
```

We first want to create a gender variable that is a character, this will make the output look nicer. We'll make use of the `ifelse()` function inside of `mutate()`.

```r
library(dplyr)

acitelli <- acitelli %>%
  mutate(Gender = ifelse(gender == -1, "Women", "Men"))

menOnly <- acitelli %>%
  filter(gender == 1) %>%
  mutate(wise_hus = Yearsmar > median(Yearsmar)) %>%
  select(-gender)
```

Visualizing Data

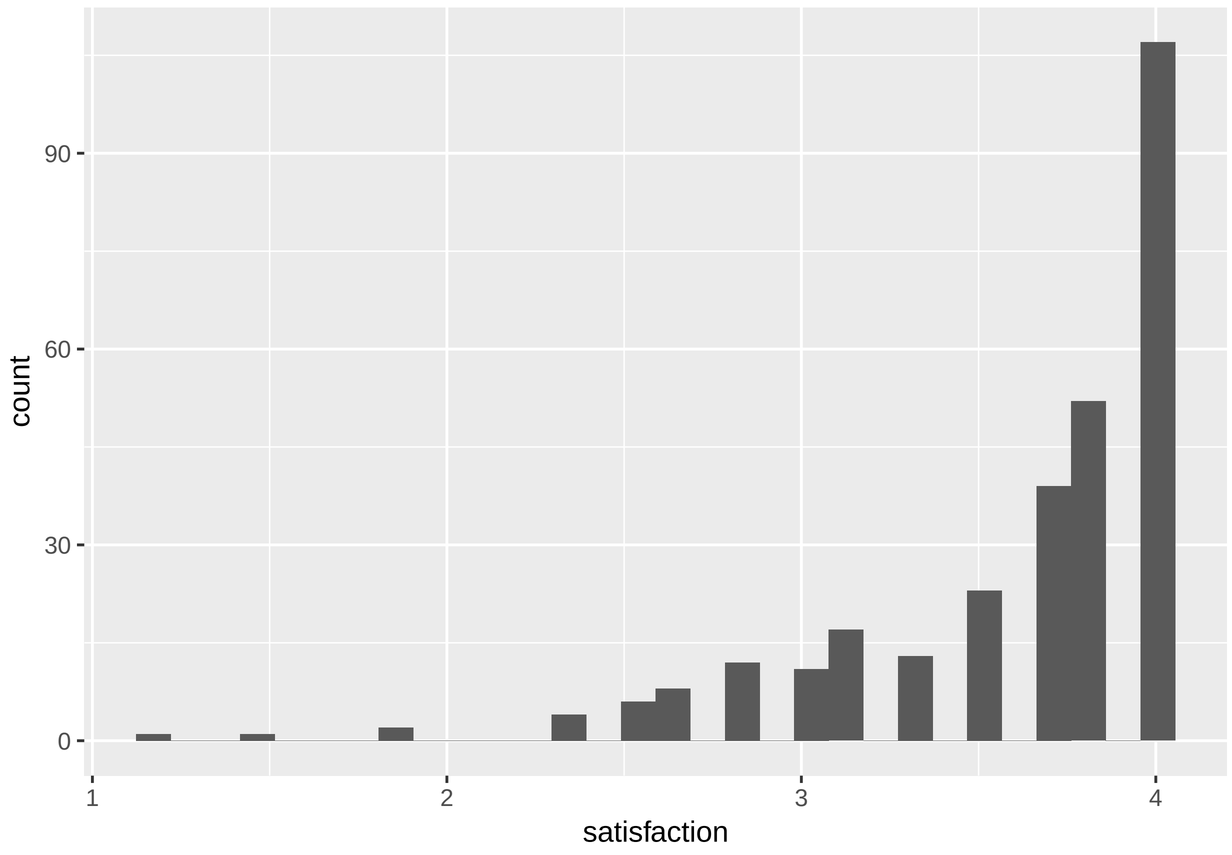
There are quite a few ways to make figures in R, we'll the popular package `ggplot2`. You can find a cheat sheet for `ggplot2` here. Be sure to install it first if you never have, or if you need to update it.

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

Quick Histogram

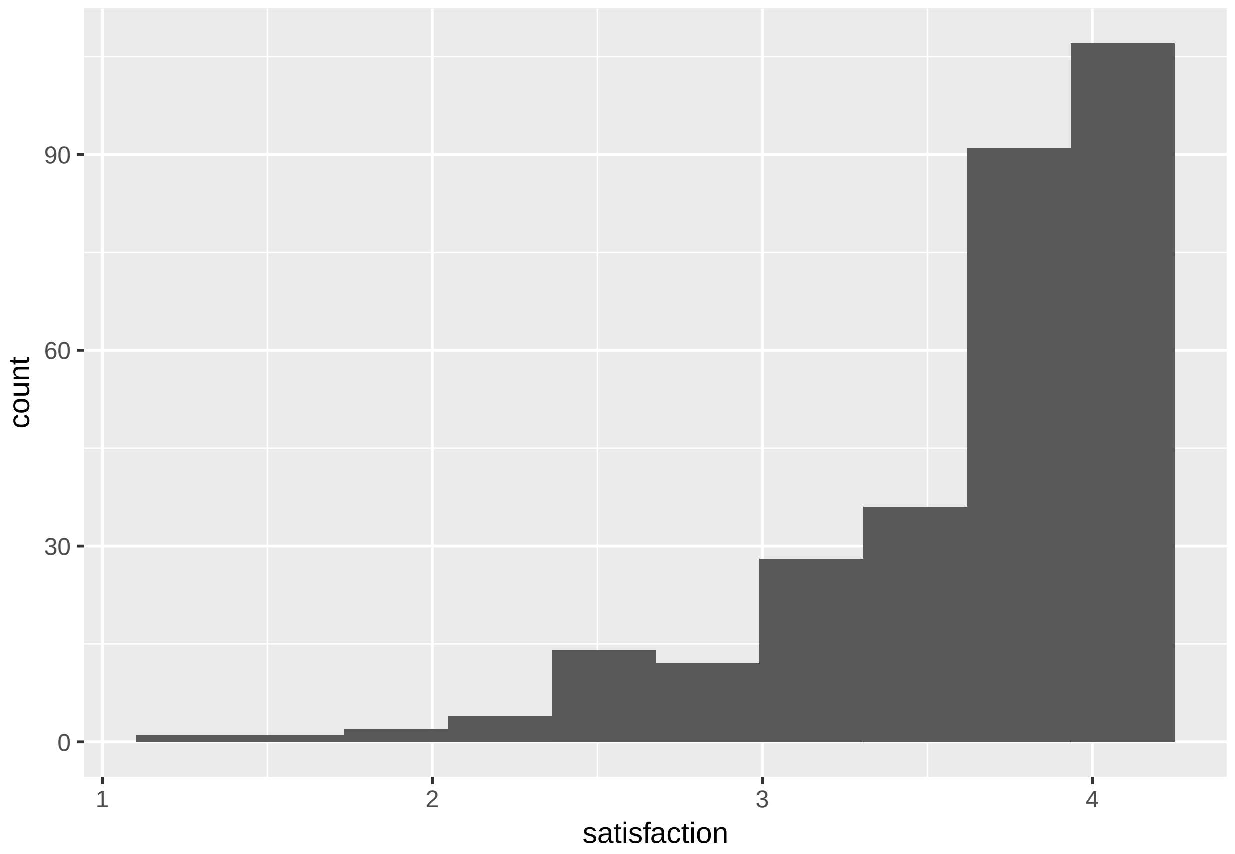
First, let's make a histogram for `satisfaction`. The easiest way to make a figure with `ggplot2` is with the `qplot()` function. This stands for *quick plot*. Notice in the code below that we did not specify anything about a histogram. `qplot()` guesses which type of plot we want based on the variable's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

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



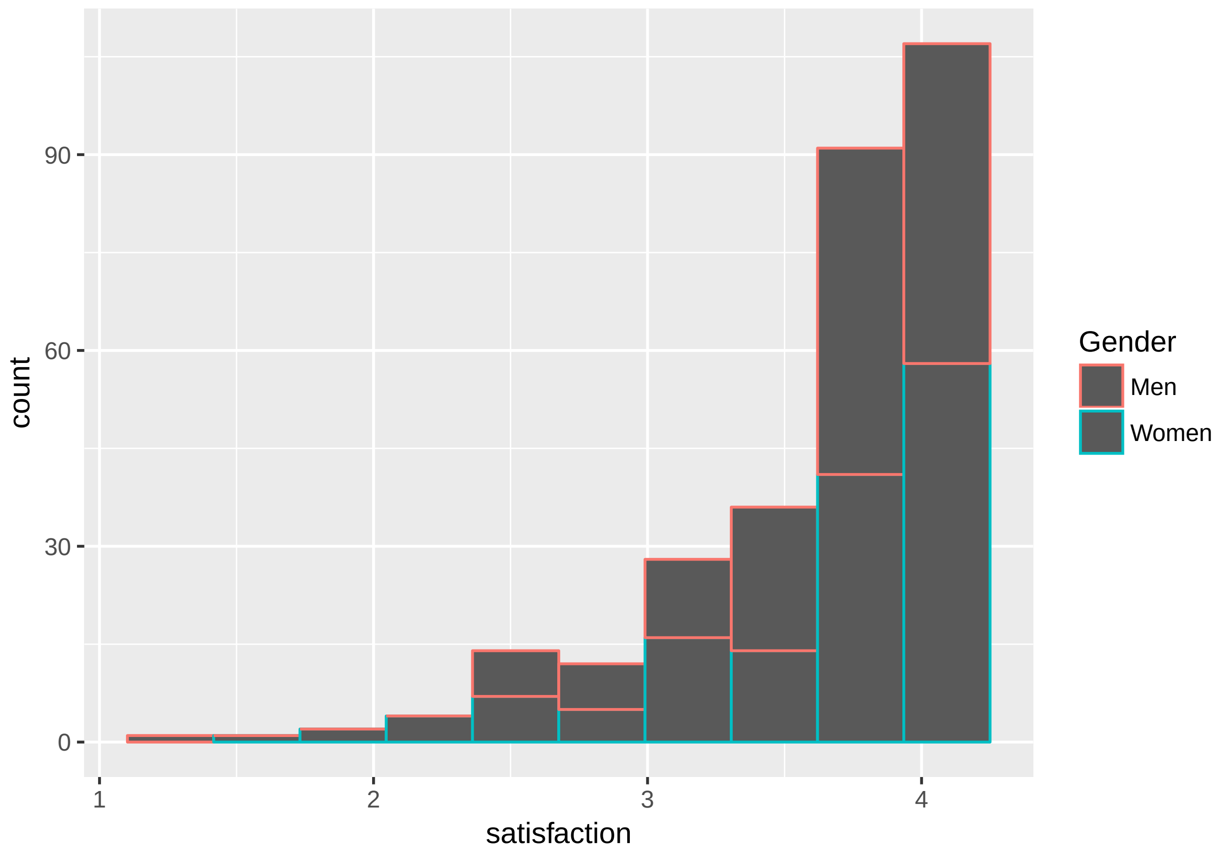
There are too many bins (it defaults to 30 bins), we can ask for a specific number by adding the `bins =` argument. Try playing around the bin number below to find the optimal plot. I put `30` in there as a placeholder.

```
qplot(satisfaction, data = acitelli, bins = 10)
```



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.

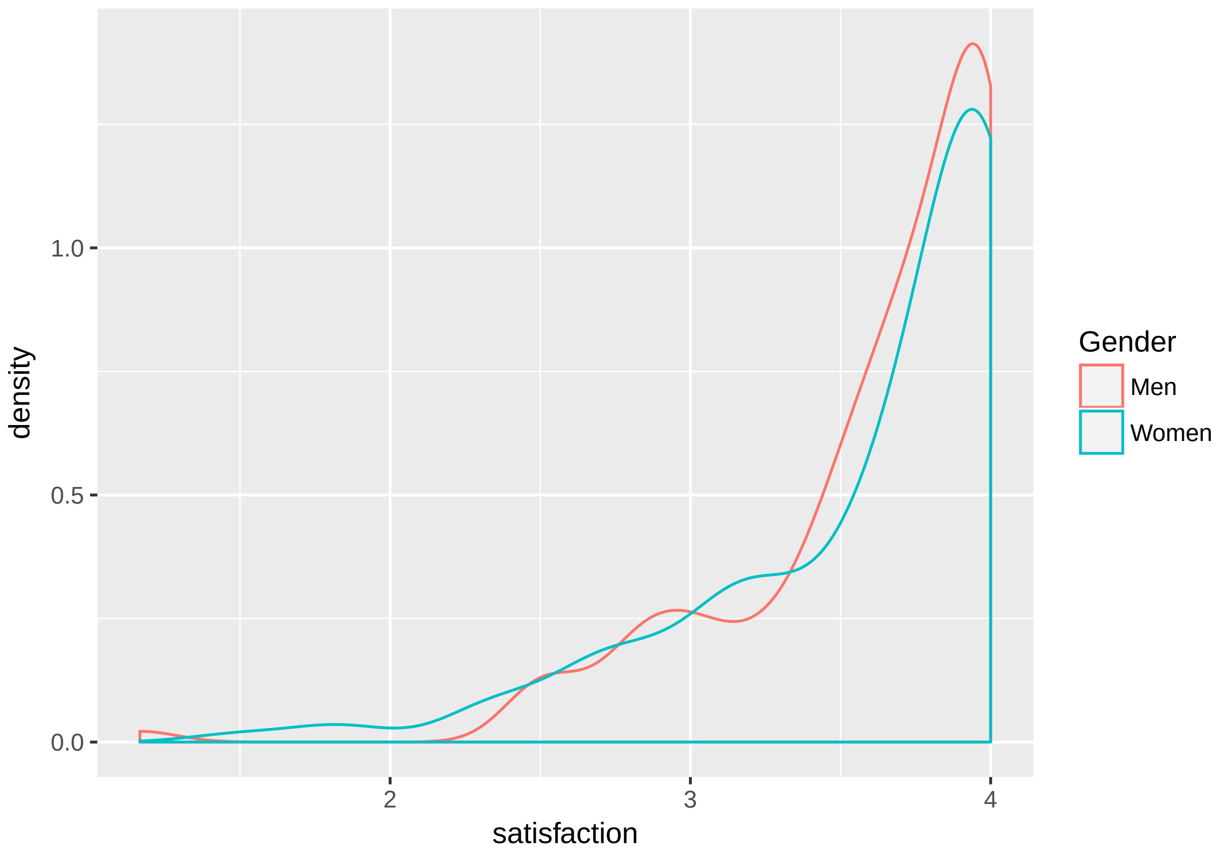
```
qplot(x = satisfaction, color = Gender, data = acitelli, bins = 10)
```



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.

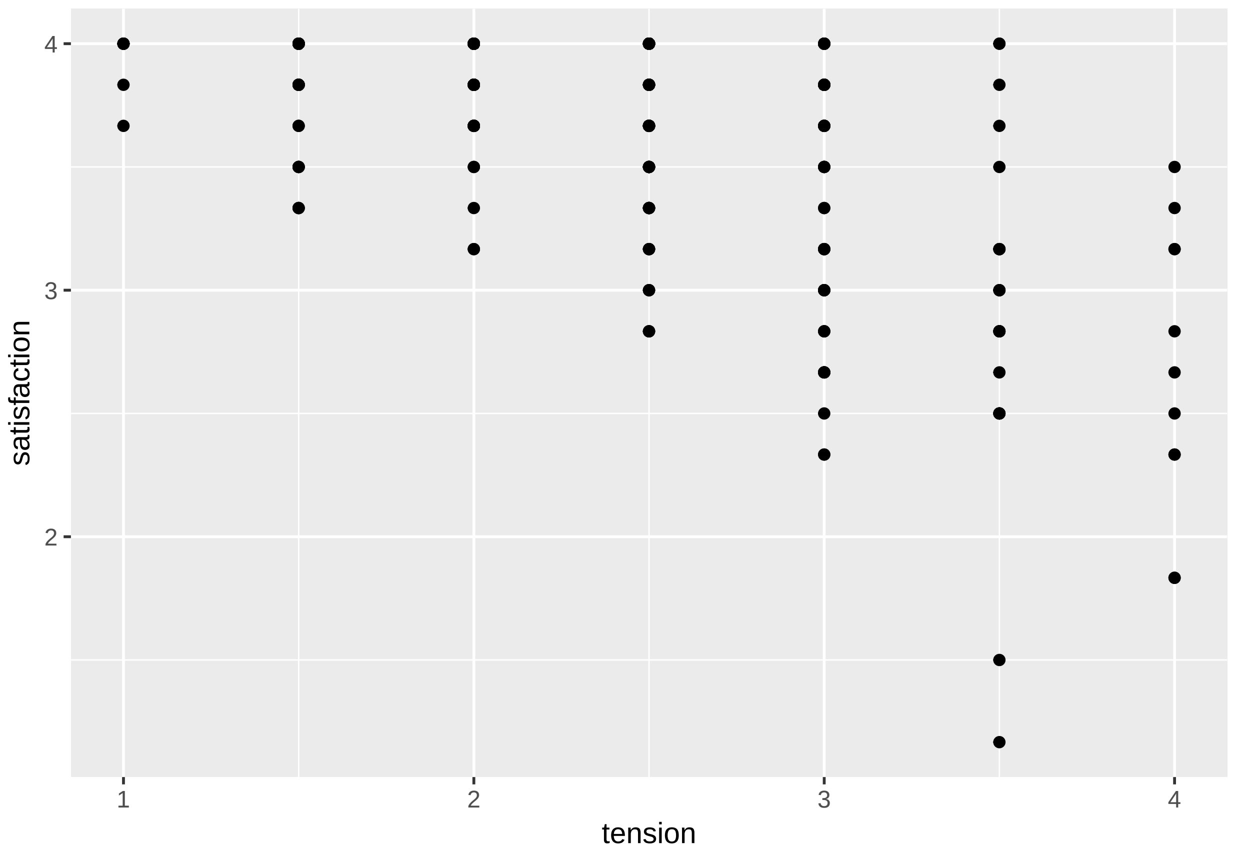
```r
qplot(x = satisfaction, color = Gender, data = acitelli, geom = "density")
```



Quick Scatterplot

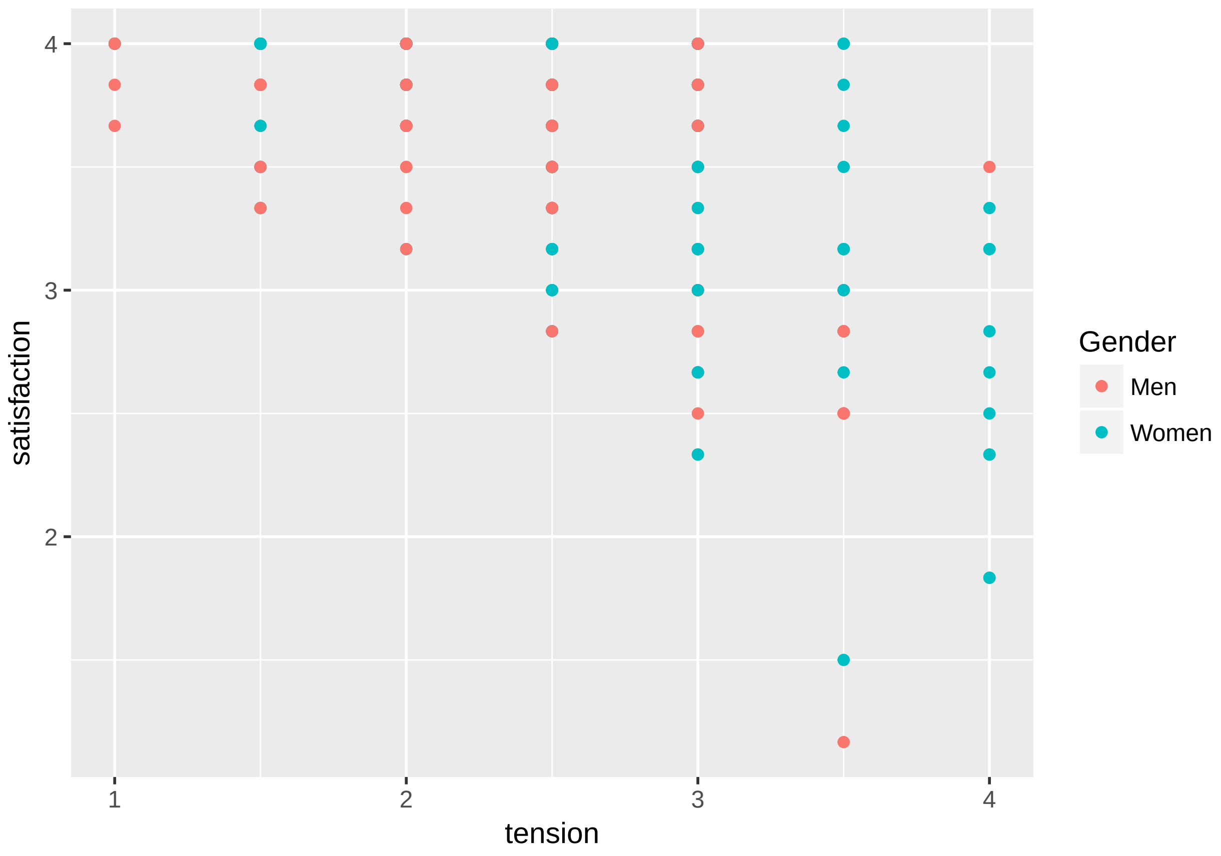
We might also want a scatter plot. Again,`qplot()` guesses what we want, but it's a good idea to specify which variable goes on the x-axis and which goes on the y-axis.

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



We can even add a third variable, mapping it to color. To get the behavior we want, gender has to be a categorical variable (a character variable is fine).

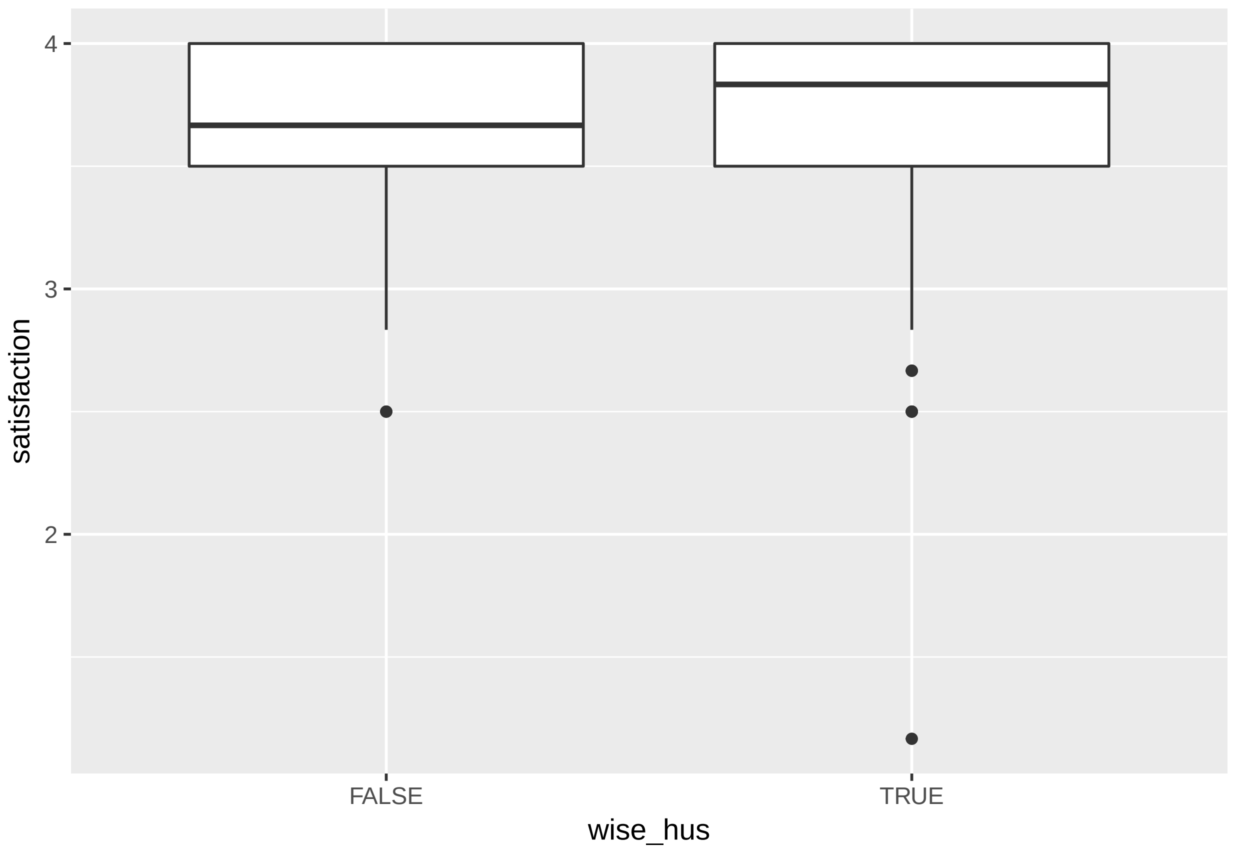
```
qplot(x = tension, y = satisfaction, color = Gender, data = acitelli)
```



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 with `geom = "boxplot"`.

```
qplot(y = satisfaction, x = wise_hus, data = menOnly, geom = "boxplot")
```

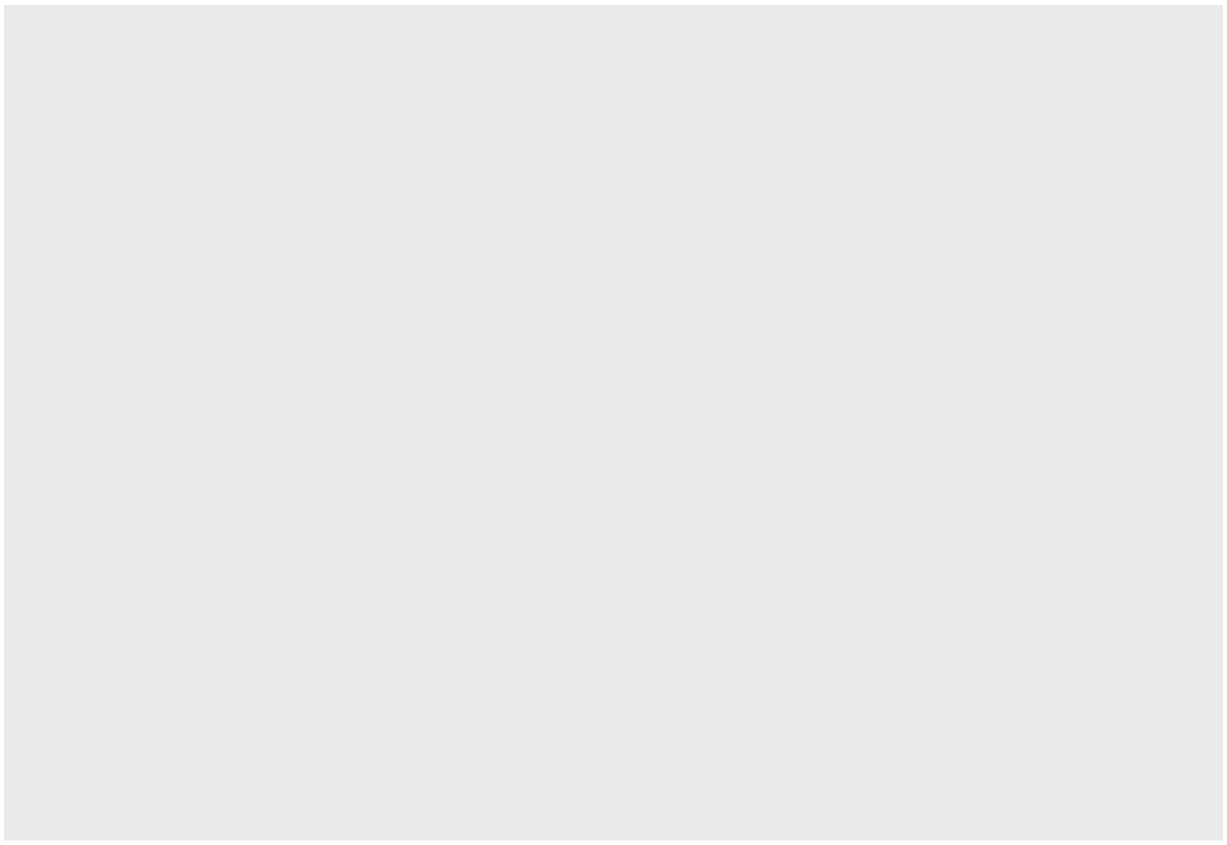

4
3
2
satisfaction
FALSE
TRUE
wise_hus

ggplot()

Histogram

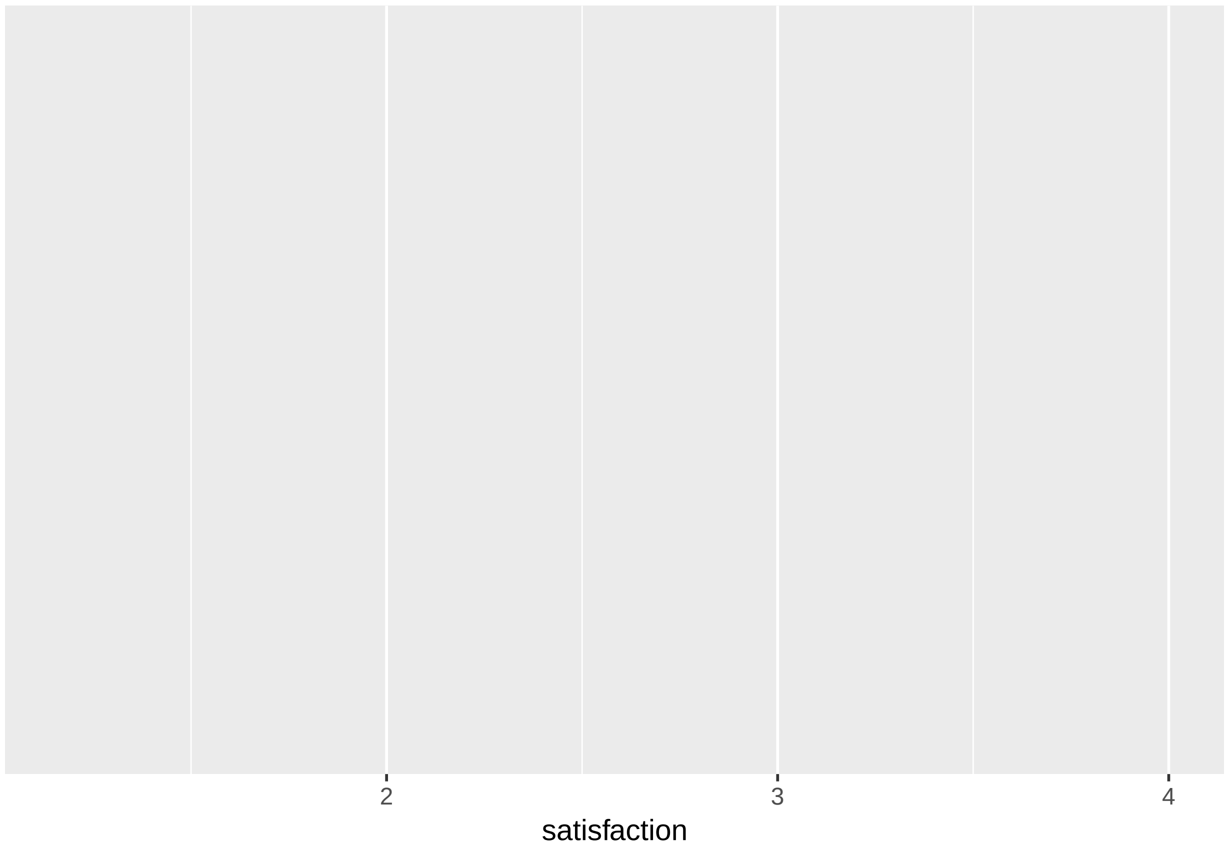
For more complex figures we will need to move away from using the qplot() function in favor of the heavy duty ggplot() function. To get a sense of how ggplot() builds plots, first we will just run the empty function.

```r
ggplot()
```



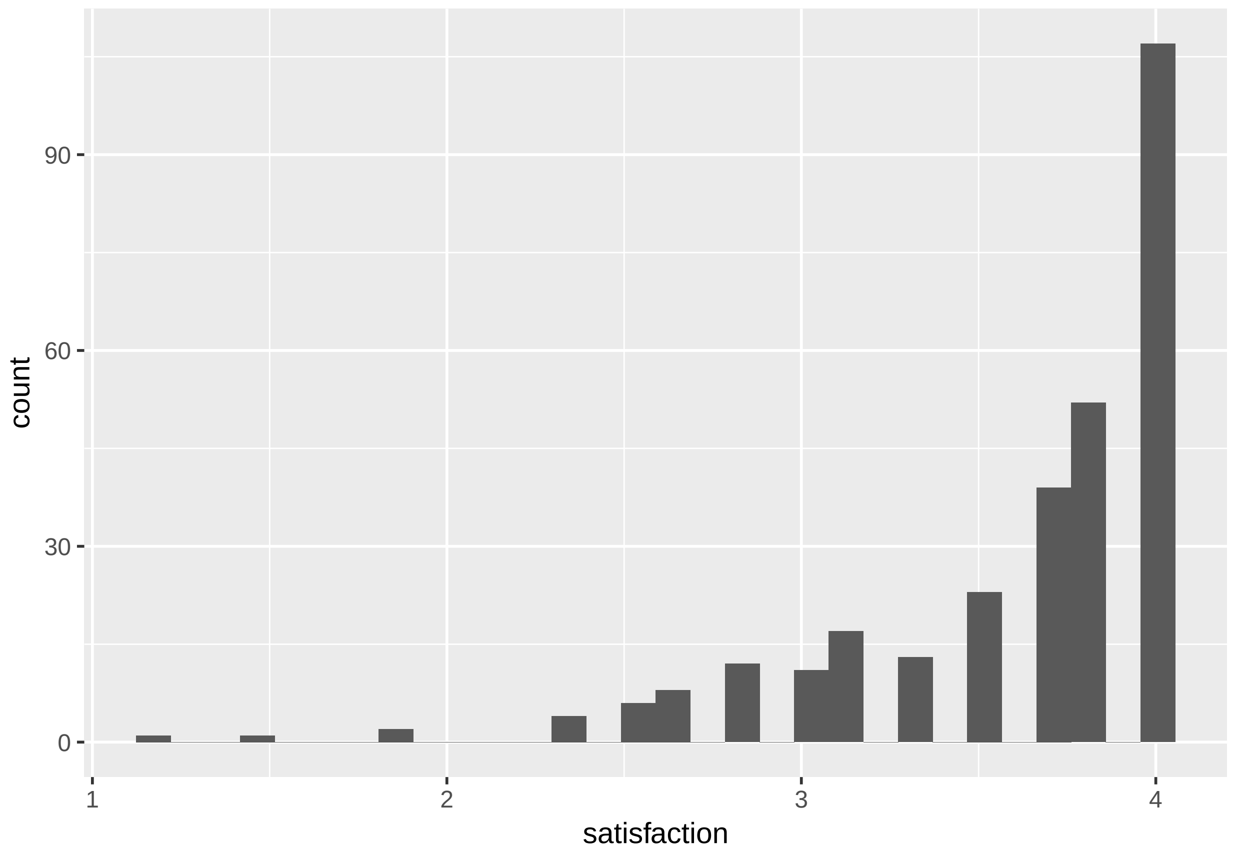
Next, we can add the data and start mapping variables to aesthetics.

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



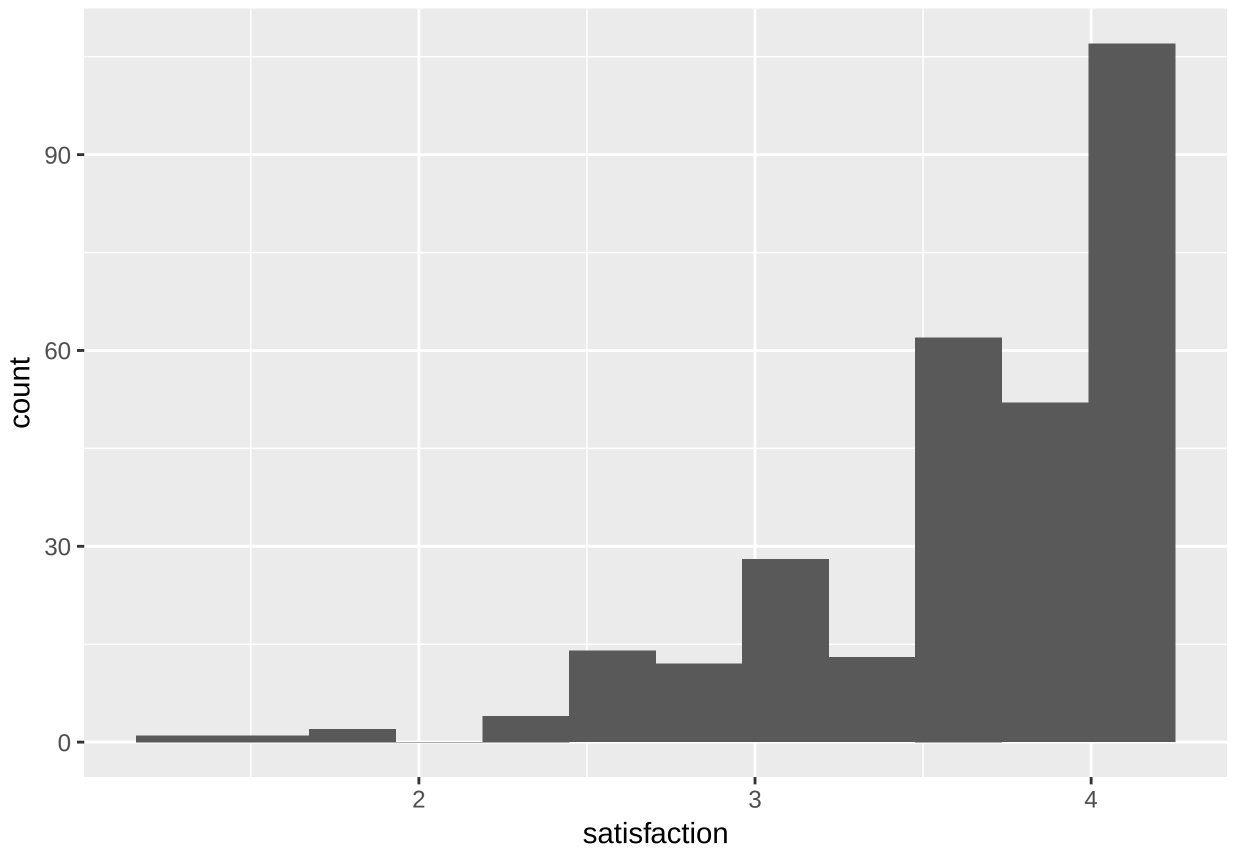
After we have specified aesthetic mappings, we can then add geoms. Notice that we make use of the `+` symbol with ggplots. The `+` needs to be on the right of each piece of the plot. We add a histogram with the `geom_histogram()` function.

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



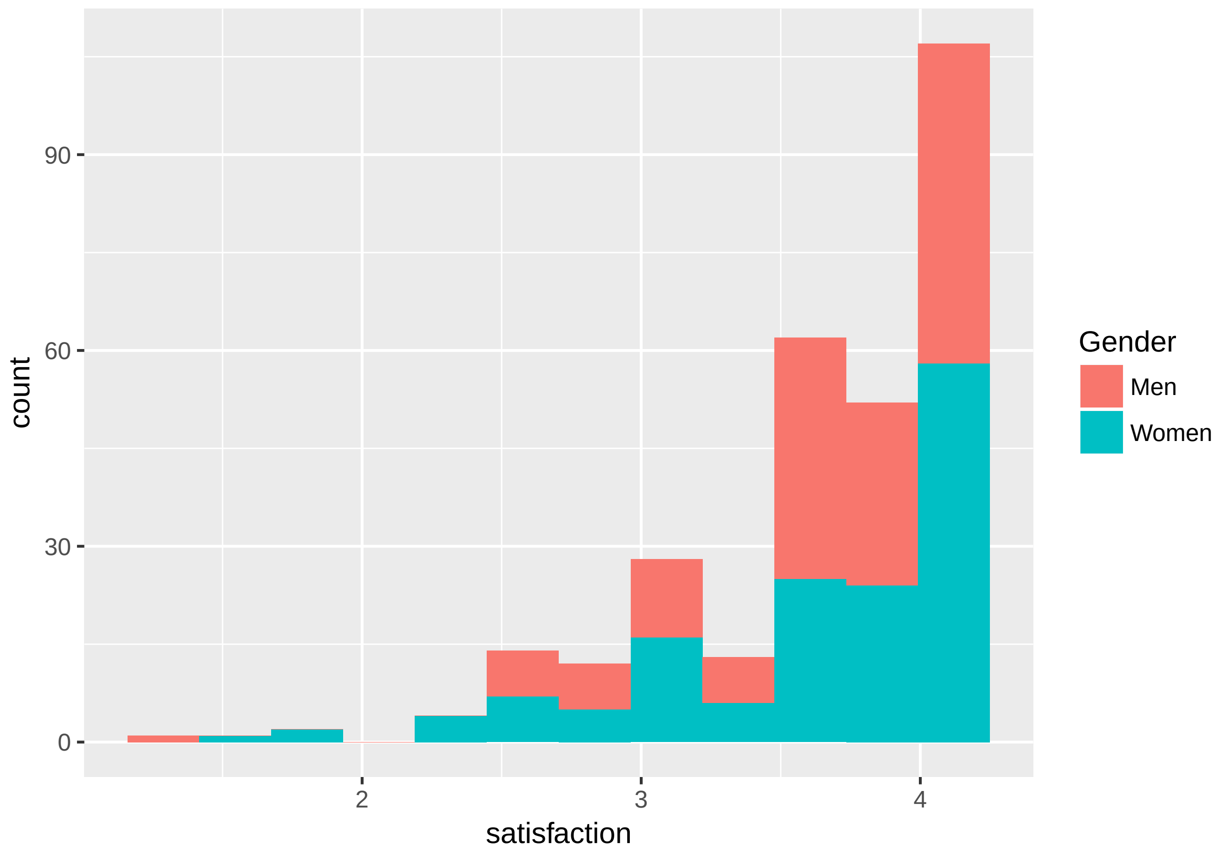
Note in the histogram above the y-axis are the counts of observations in each bin. There were some calculations involved in getting these counts. Counts are the default *statistic* when you ask for a histogram. We can change the number of bins by adding `bins =` inside of the `geom_histogram()` function.

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



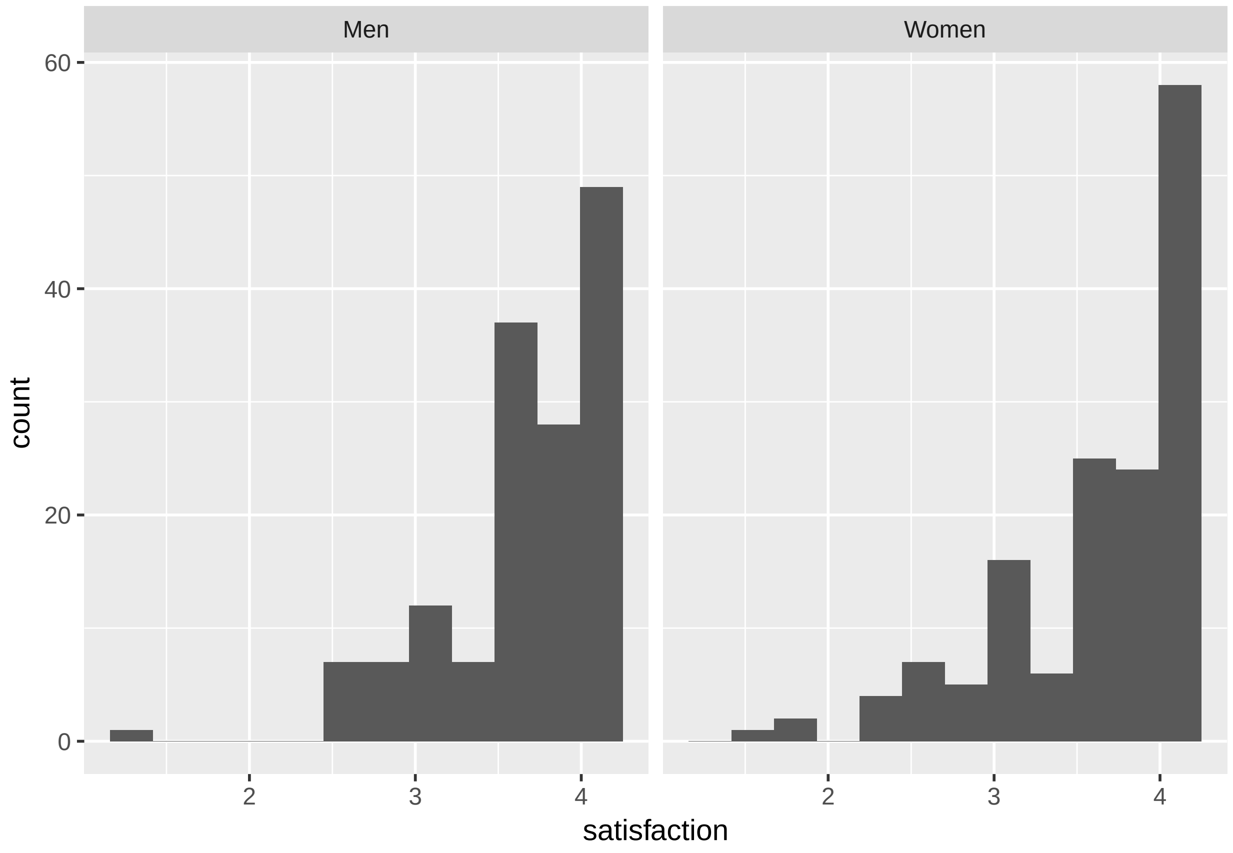
Just as before, if we want overlayed histograms, we can map gender to the fill aesthetic.

```r
ggplot(acitelli, aes(x = satisfaction, fill = Gender)) +
  geom_histogram(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.

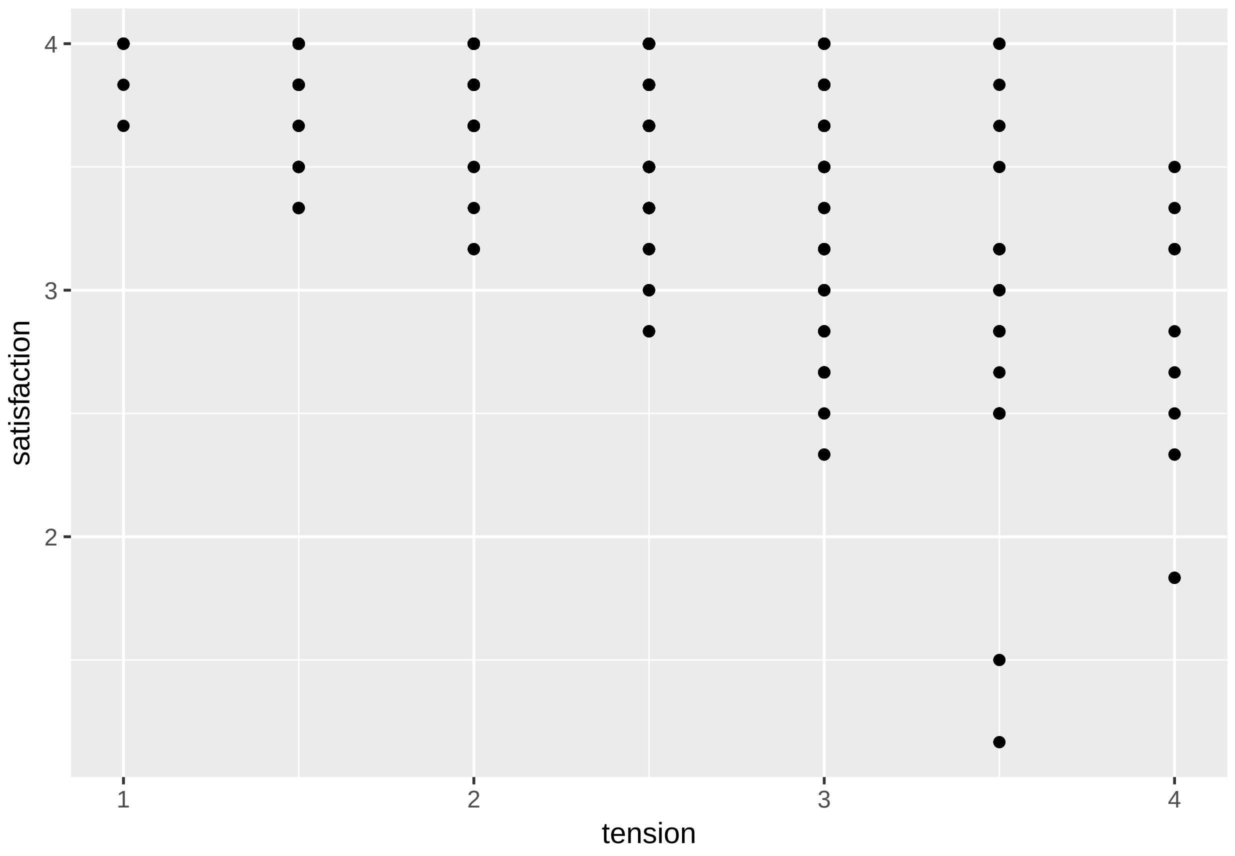
```r
ggplot(acitelli, aes(x = satisfaction)) +
  geom_histogram(bins = 12) +
  facet_wrap(~Gender)
```



Scatter Plot

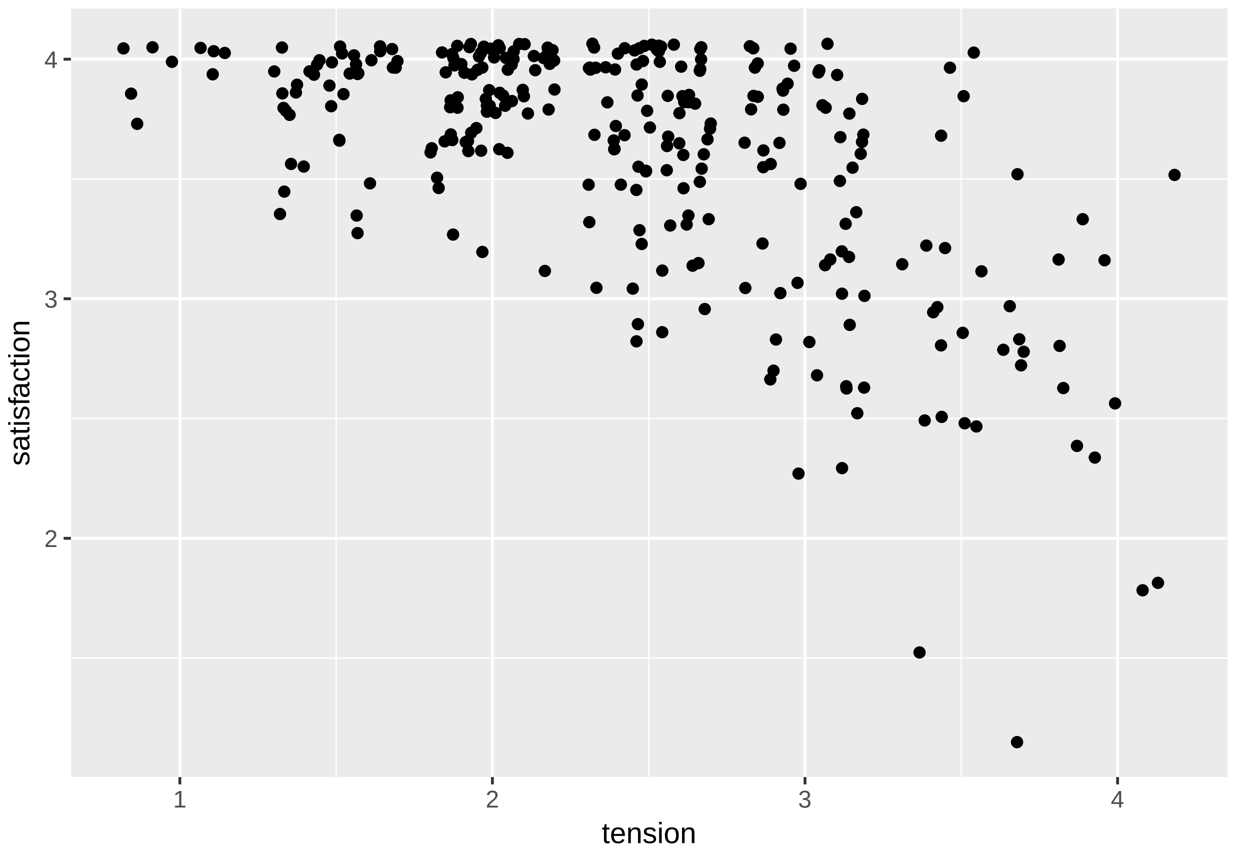
Next we'll make that scatter plot again. We'll map `tension` to the x-axis and `satisfaction` to the y-axis. Then we'll add `geom_point()`.

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



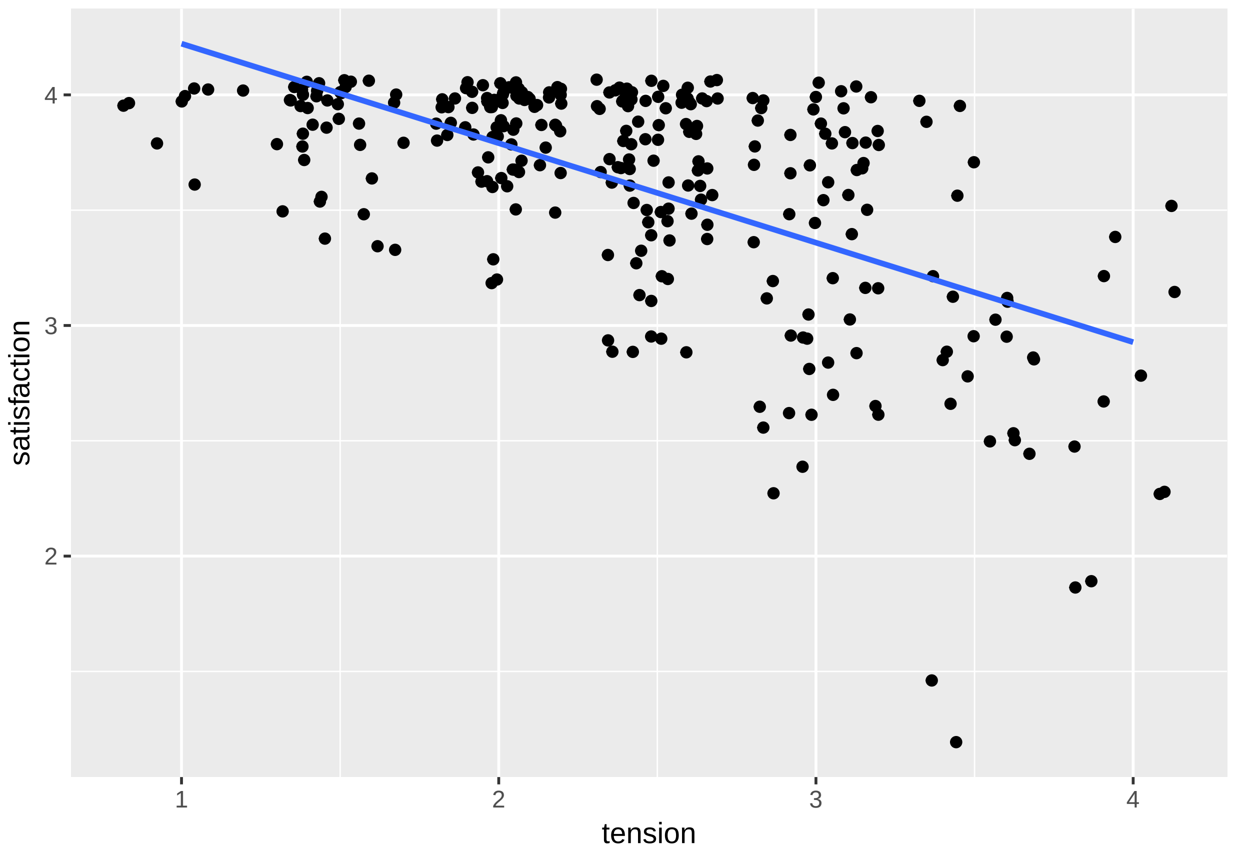
Why does it appear as though there is far less data than there really is? Check out the plot when we use `geom_jitter()`. What do you think `geom_jitter()` does?

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



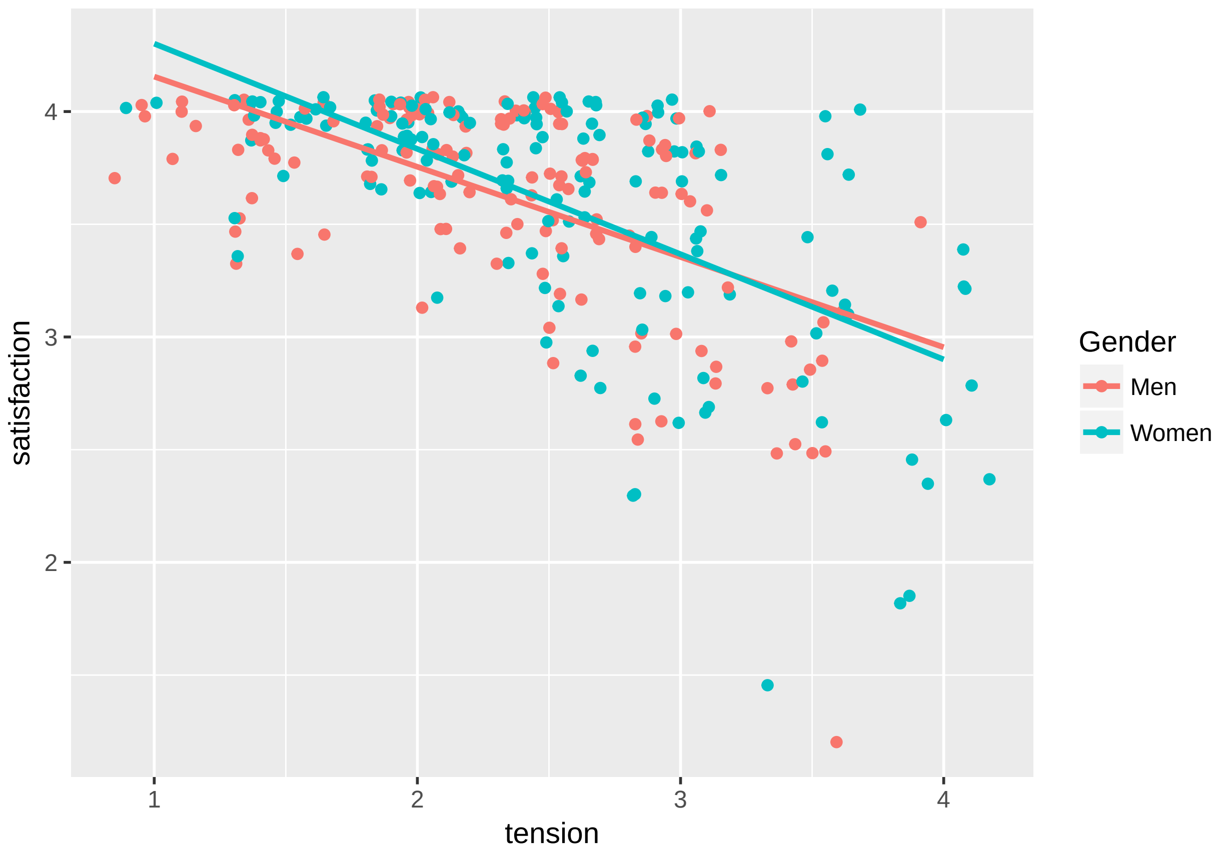
We can add more than one geom. To the jittered scatter plot we can add a least squares regression line with `geom_smooth()`. Inside of geom smooth we need to specific `method = "lm"`, the lm stand for *linear model*. We can also turn off the standard errors with se = 0.

```r
ggplot(acitelli, aes(x = tension, y = satisfaction)) +
  geom_jitter() +
  geom_smooth(method = "lm", se = 0)
```



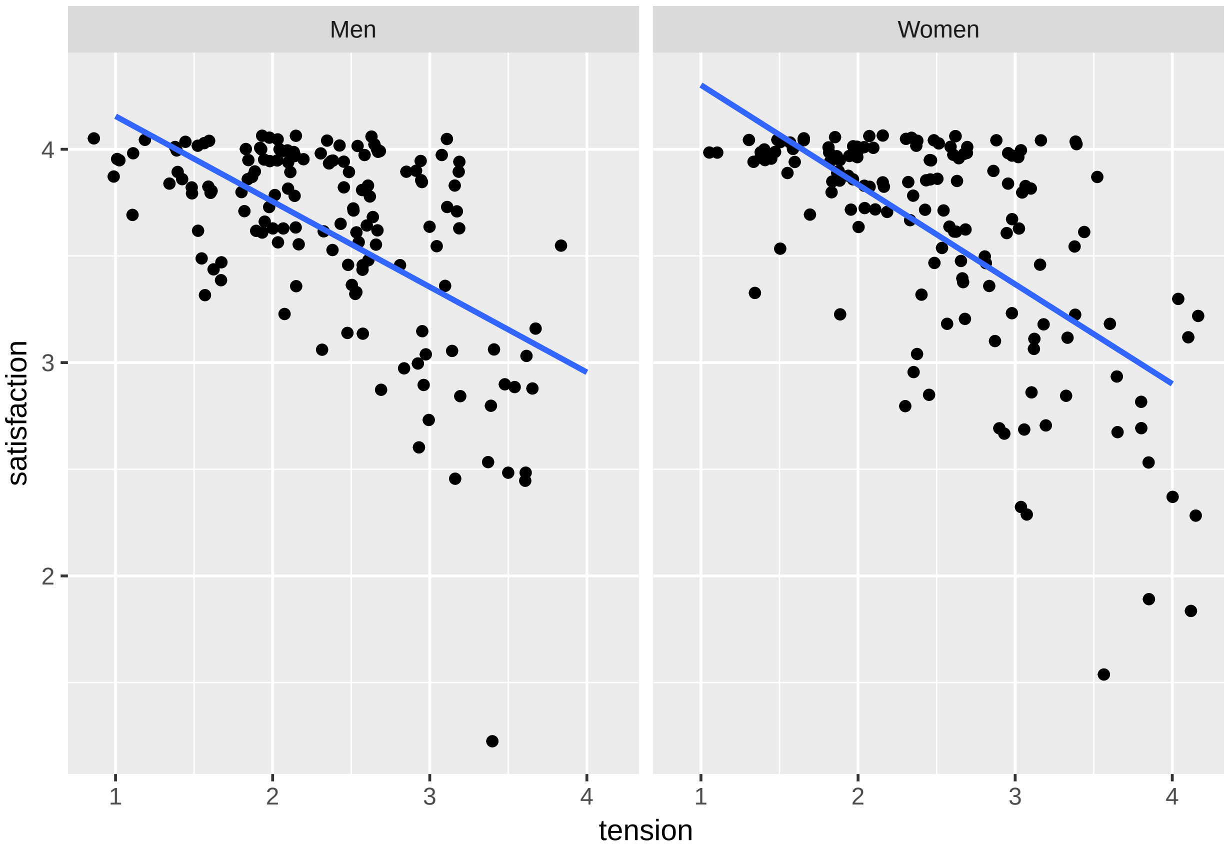
Again, we can map gender to the color aesthetic.

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



Or use `facet_wrap()`.

```
ggplot(acitelli, aes(x = tension, y = satisfaction)) +
  geom_jitter() +
  geom_smooth(method = "lm", se = 0) +
  facet_wrap(~Gender)
```

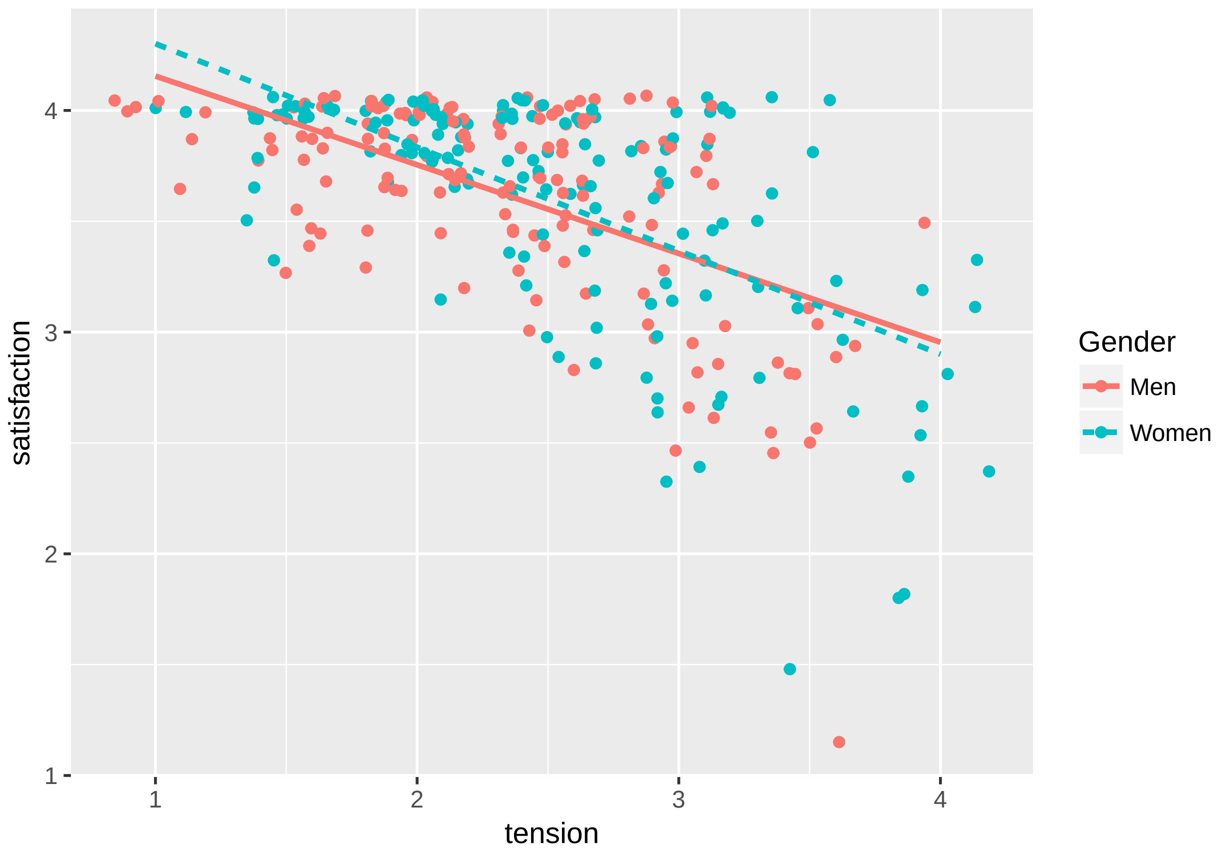


Labels and Colors

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.

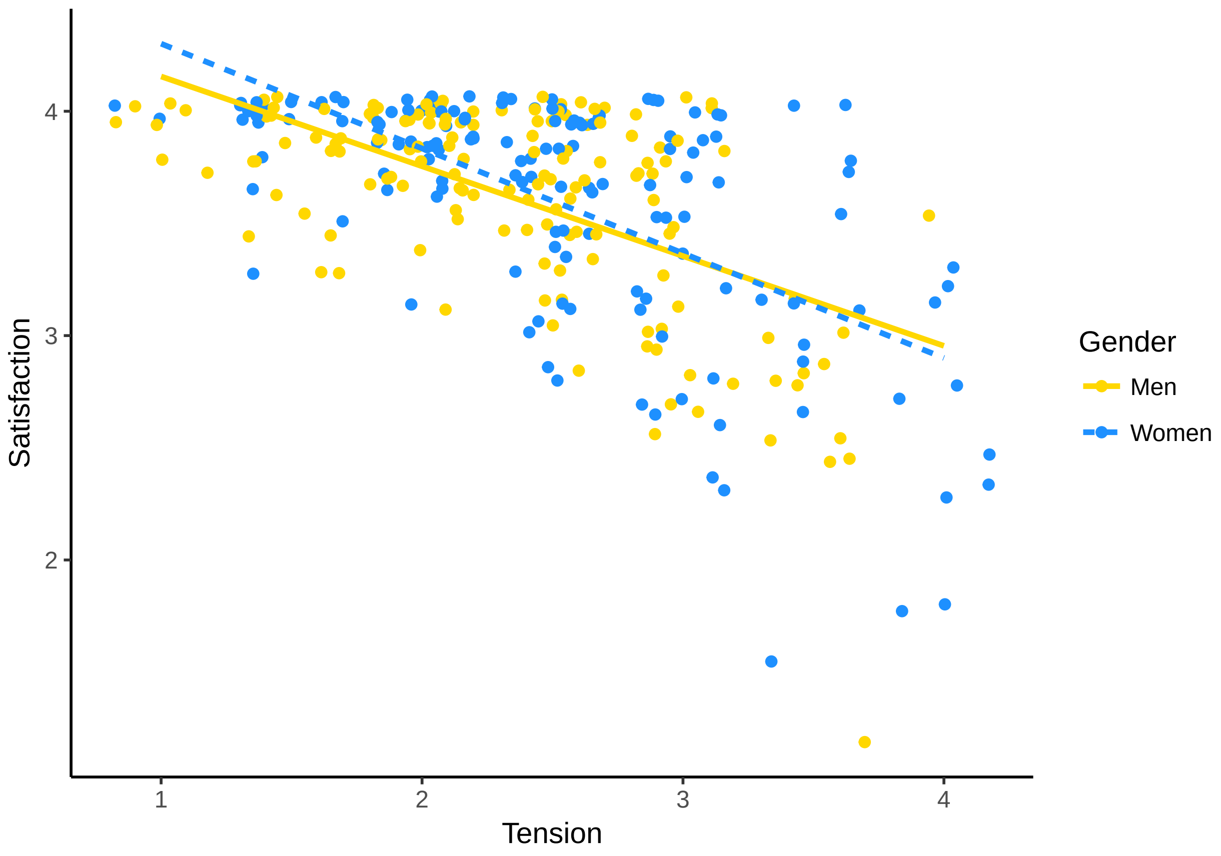
```
myplot <- ggplot(acitelli, aes(x = tension, y = satisfaction, color = Gender, linetype =
  geom_jitter() +
  geom_smooth(method = "lm", se = 0)

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")) +
  theme_classic()
```


Satisfaction
4
3
2
1
2
3
4
Tension
Gender
Men
Women