

# R WORKSHOP – DAY 1

---

Randi L. Garcia

Smith College

July 17<sup>th</sup>, 19<sup>th</sup>, and 21<sup>st</sup>

# DAY 1

---

- Installing R and RStudio
- RStudio environment, packages, and R Markdown
- Data cleaning
- Making figures
- Descriptive stats, correlations, reliability, t-tests

# DAY 2

---

- ANOVA and regression
- Preparing APA style manuscripts
- Exploratory Factor Analysis (EFA) and Confirmatory Factor Analysis (CFA)
- Path Analysis and SEM

# DAY 3

---

- Multilevel Modeling
- Growth Curve Modeling
- Dyadic Data Analysis

# Workshop Materials

>Find the workshop schedule and data examples here:

<https://randilgarcia.github.io/smith-r-workshop/>

>Download ALL materials, including R-code, here:

<https://github.com/RandiLGarcia/smith-r-workshop>

# Installing R and RStudio

- Click links on Website...
- When we're done I'll take you on a tour of RStudio

# Data Cleaning

- The package we'll use for data cleaning is called dplyr.
- The Verbs
  - filter()
  - mutate()
  - rename()
  - arrange()
  - select()
  - summarize()
  - group\_by()

# Data Cleaning

- Each verb performs familiar operations of a dataset

| Verb        | What is does                             | ...in SPSS                      |
|-------------|------------------------------------------|---------------------------------|
| mutate()    | Creates new variables                    | COMPUTE (or transform in menu)  |
| filter()    | Filters for specific cases               | FILTER (or select data in menu) |
| arrange()   | Sorts using some logic                   | SORT                            |
| select()    | Subsets for only certain variables       | DROP                            |
| group_by()  | Groups dataset by a categorical variable | Like split file in menu         |
| summarize() | Create a summary table                   | Descriptive statistics          |

# Data Cleaning

- We will use the pipe operator to combine verbs!
- The pipe:

`%>%`

`filter(dataset, age >= 18)`

...is the same as:

`dataset %>% filter(age >= 18)`

# Data Cleaning

- Why the pipe!?!?
- Instead of reading/writing:

```
select(filter(mutate(dataset, bdi = bdi1 + bdi2 + bdi3), age >= 18) bdi, soc_sprt)
```

- We can write:

```
dataset %>%  
  mutate(bdi = bdi1 + bdi2 + bdi3) %>%  
  filter(age >= 18) %>%  
  select(bdi, soc_sprt)
```

# Data Cleaning

- If you wanted to save this two-variable dataset:

```
dataset_small <- dataset %>%  
  mutate(bdi = bdi1 + bdi2 + bdi3) %>%  
  filter(age >= 18) %>%  
  select(bdi, soc_sprt)
```

# Data Cleaning

- There are also verbs for **joining two tables**
  - Adding cases from another dataset that has the same variables
    - And also when they do not have exactly the same variables
  - Adding variables from another dataset
    - These variables can be measured at the same level or at a higher level, as long as there are index variables.
- And verbs for **transforming data** from
  - Wide-to-long
  - Long-to-wide
- We can learn these on Day 3 if you want...?

# R MARKDOWN FILE

---

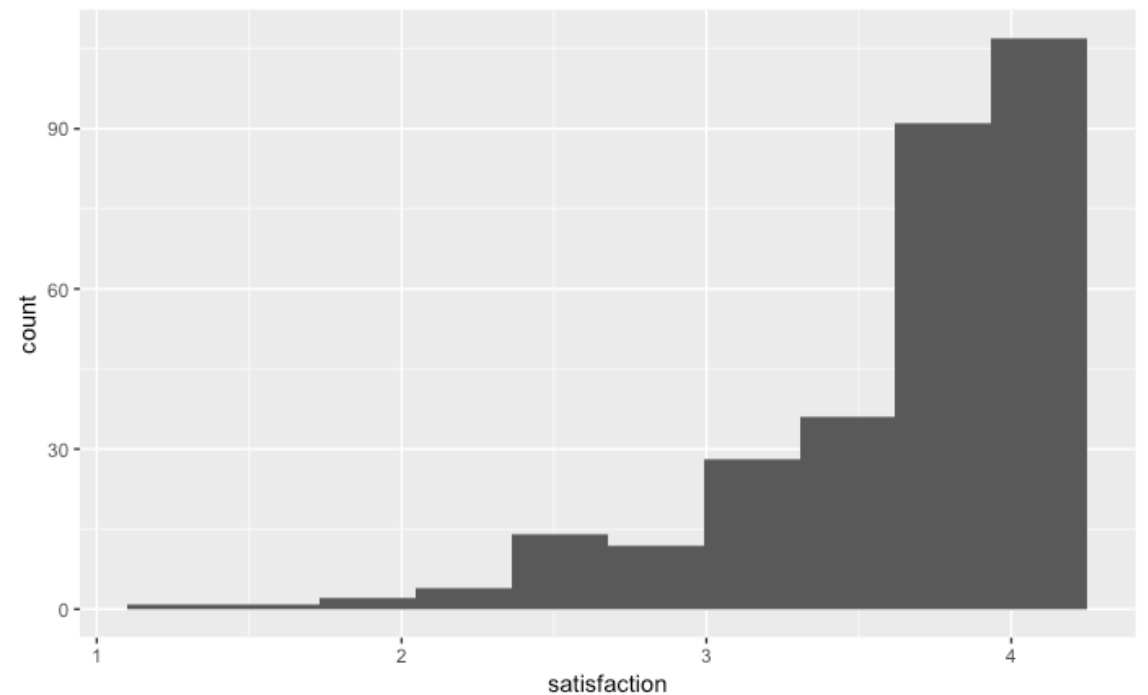
# Making Figures with ggplot2

- As with everything else, there are lots of ways to make figures in R
  - Base R
  - Lattice graphics
  - The ggplot2 package
- We'll be learning the ggplot2 package.
  - It makes beautiful visualizations
  - It's popular so there is a lot of help on the internet and companion code
  - Written by the same person who wrote dplyr and it works at the end of a pipeline
  - It is what Smith students know how to use

# Making Figures with ggplot2

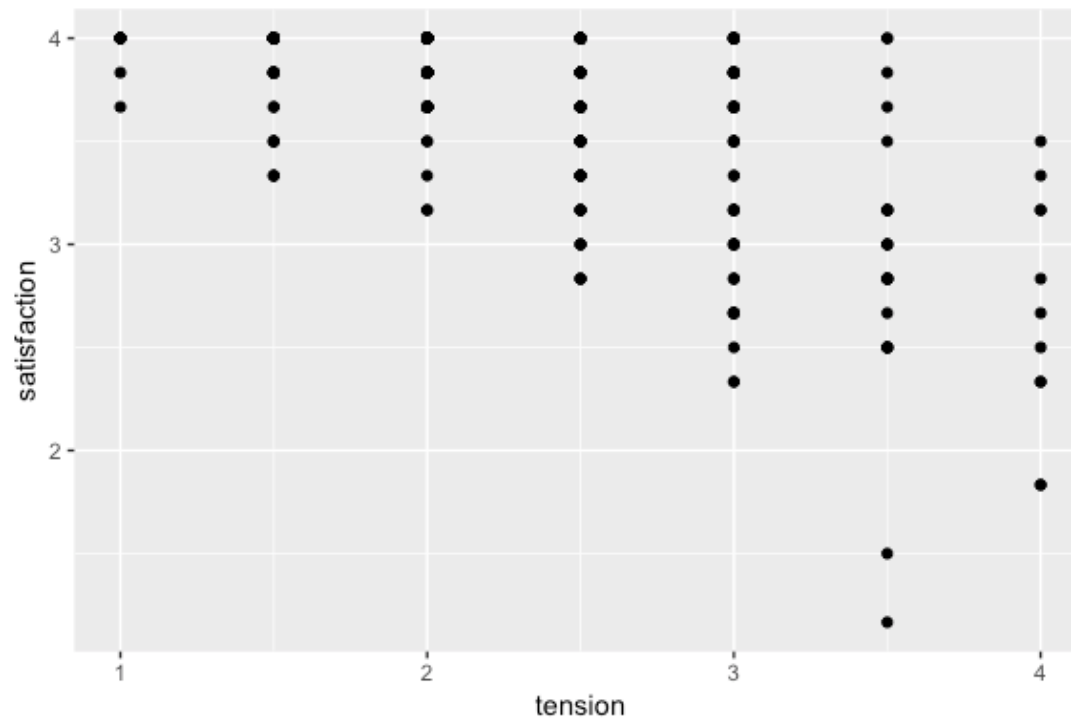
- The easiest figures are made with the `qplot()` function

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



# Making Figures with ggplot2

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



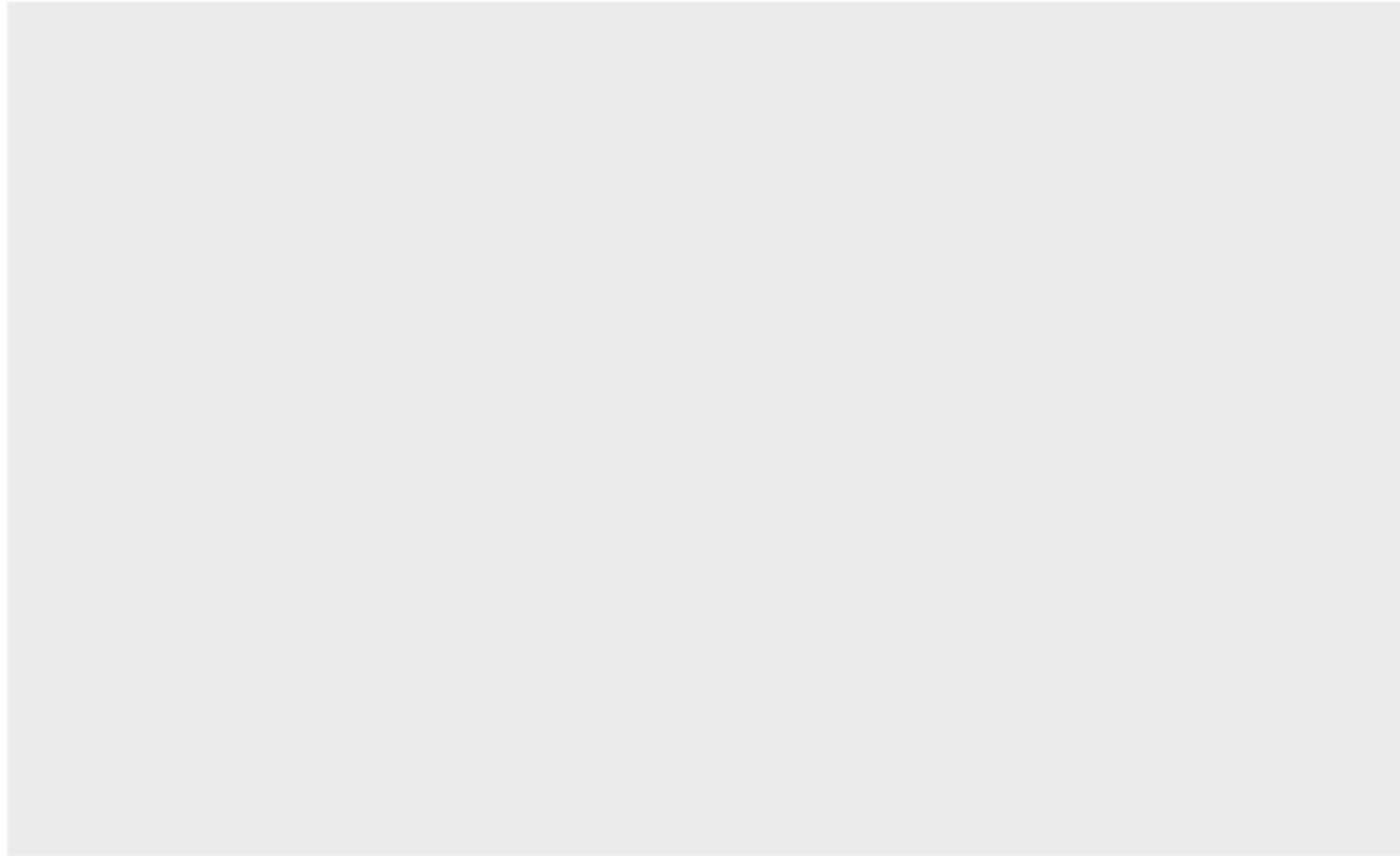
# Making Figures with ggplot2

- The `qplot()` function is good for quick visualizations
  - Good for probably 80% of what you'd want to do while analyzing data
- But, you'll use the `ggplot()` function for anything more involved, probably for making figures for publication
- The `ggplot2` packages uses the "grammar of graphics"

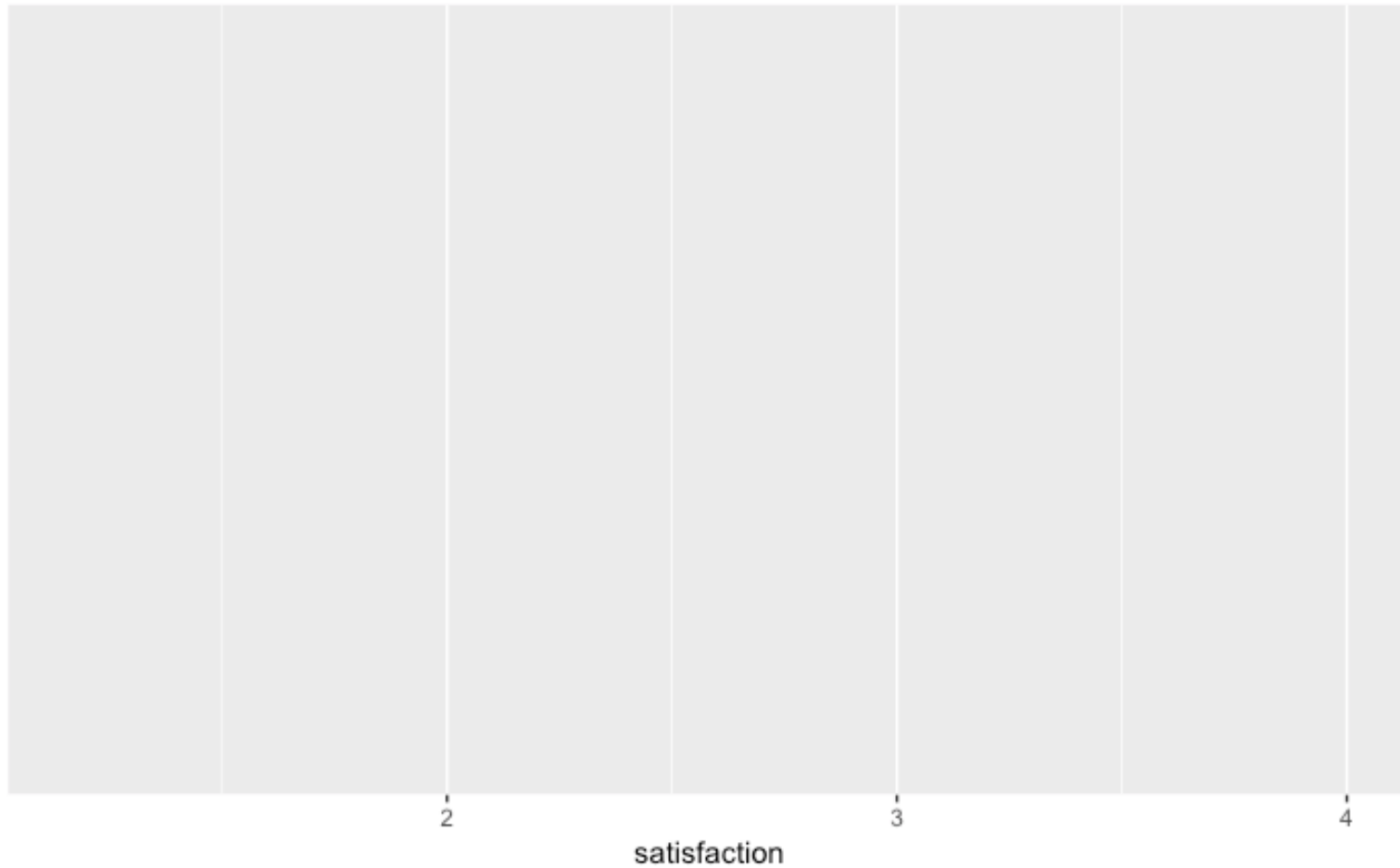
# Making Figures with ggplot2

- We independently specify pieces of the graph using the “grammar of graphics”
- Building blocks:
  - **Data**
  - **Geometric objects** (the actual things we’ll draw: points, lines, boxplot, histograms, etc.)
  - **Aesthetic mappings** (what and where we’ll draw: x-axis, y-axis, color, fill, shape, size, linetype, etc.)
  - **Statistics** (implied or specified computing to be done)
  - **Scales** (range of values, colors, or shapes)
  - **Facets** (the panes—there can be more than 1, layers in SPSS)
  - **Guides** (legends—what the humans see)

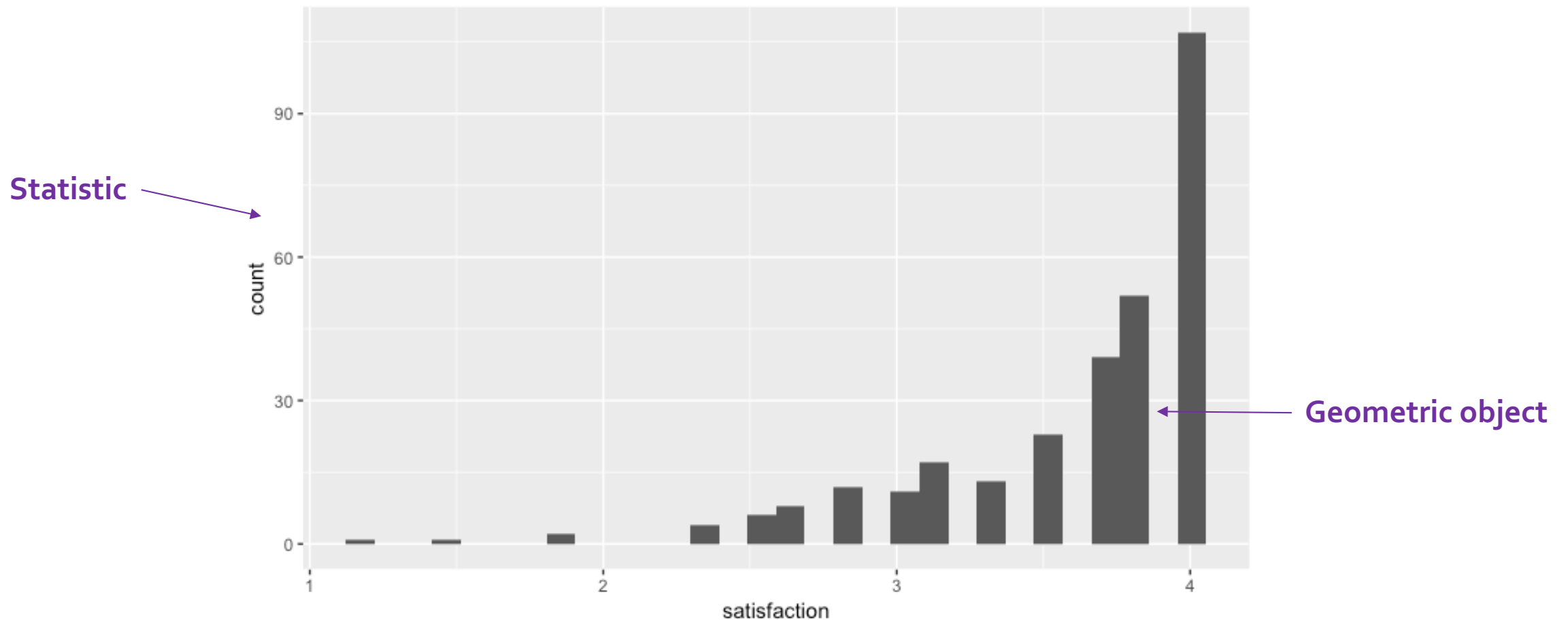
# ggplot()



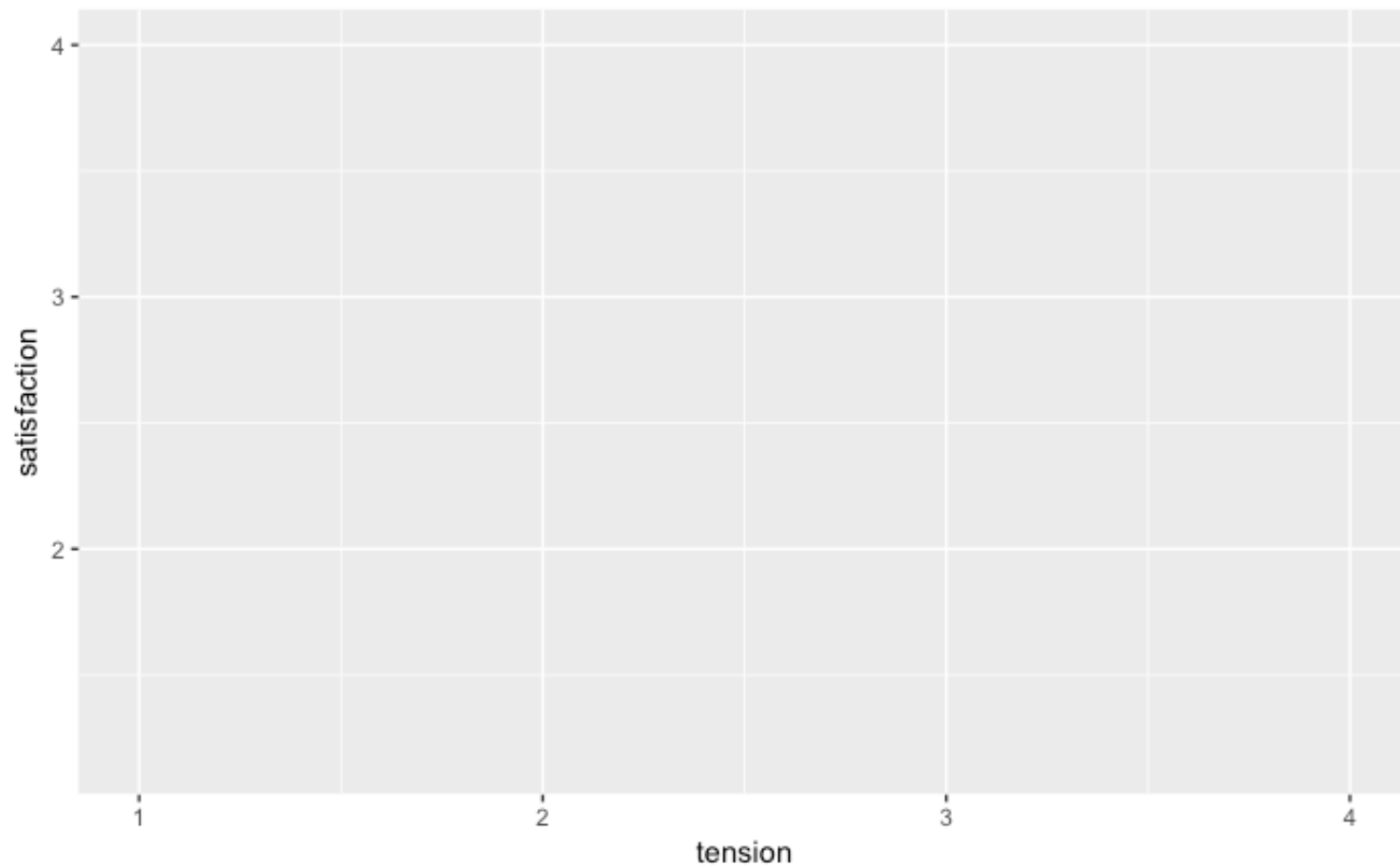
```
ggplot(acitelli, aes(x = satisfaction))
```



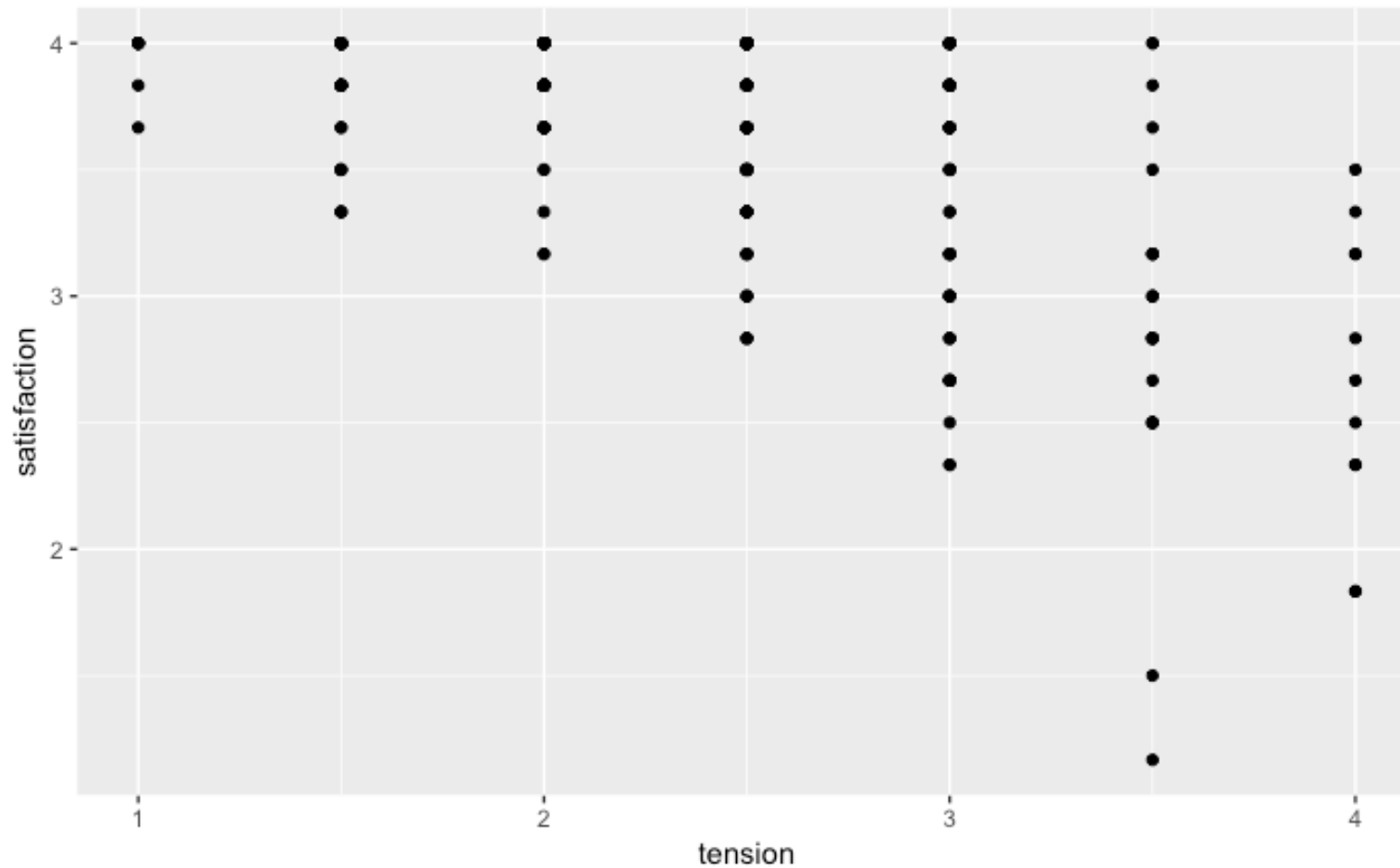
```
ggplot(acitelli, aes(x = satisfaction)) +  
  geom_histogram()
```



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



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



# R MARKDOWN FILE

---

# Correlation Matrices, Reliability, and t-Tests

- For correlation matrices and Cronbach's alpha we'll use the package called **psych**
- For t-Tests I recommend you use **mosaic** because it has that (now familiar) formula, then data, syntax

# Correlation Matrices, Reliability, and t-Tests

- Correlation Matrix

- `corr.test()`
- I like to use this with `select()`:

```
corr.test(select(acitelli, tension, self_pos, other_pos, satisfaction))
```



vars for matrix

- Reliability

- `alpha()`
- Also handy with `select()`:

```
alpha(select(acitelli, sat1, sat2, sat3, sat4, sat5, sat6))
```



items for alpha

# Correlation Matrices, Reliability, and t-Tests

- Independent samples and paired samples t-tests

```
t.test(satisfaction ~ Gender, data = acitelli)
```

```
t.test(self_pos, other_pos, data = acitelli, paired = TRUE)
```