```r
library(dplyr)

rladies_global %>%
  filter(city == 'Austin')
```

R-Ladies

R FOR DATA SCIENCE: Exploratory Data Analysis and Workflow

Hello!

Welcome to R-Ladies

1. Introduction

R language, RStudio,
R4DS Workshop series

Three things
you’ll need to install

1. **Install R** -- this is the open-source programming language we’ll use (download via CRAN -- Comprehensive R Archive Network)
2. **Install RStudio** -- this is the most popular IDE for R and will make your life a lot easier (download from rstudio.com/download)
3. **Install the tidyverse** -- this is the group of packages we’ll use within R to work with data. Install with one line of code in R:

```
install.packages(“tidyverse”)
```

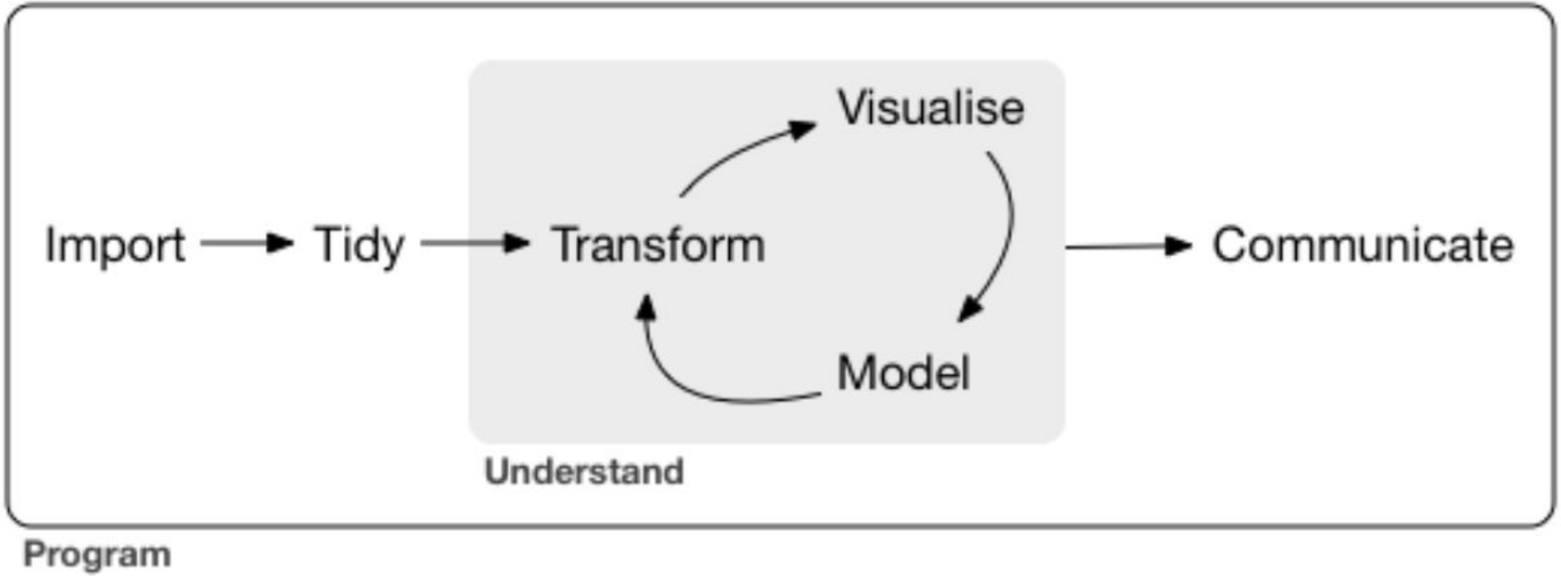
1b. Introduction

R for Data Science Workshop Series

R4DS
Workshop Series

- Exploring Data with ggplot2 + dplyr [COMPLETE-see github]
- Exploratory Data Analysis and Workflow [today]
- Data Wrangling in the Tidyverse [November 29]
- Programming -- Functions, Vectors, and Iteration [December 13]
- Modeling with modelr, purrr, and broom [January 24]
- Communicating Results with rmarkdown and ggplot2 [February 21]

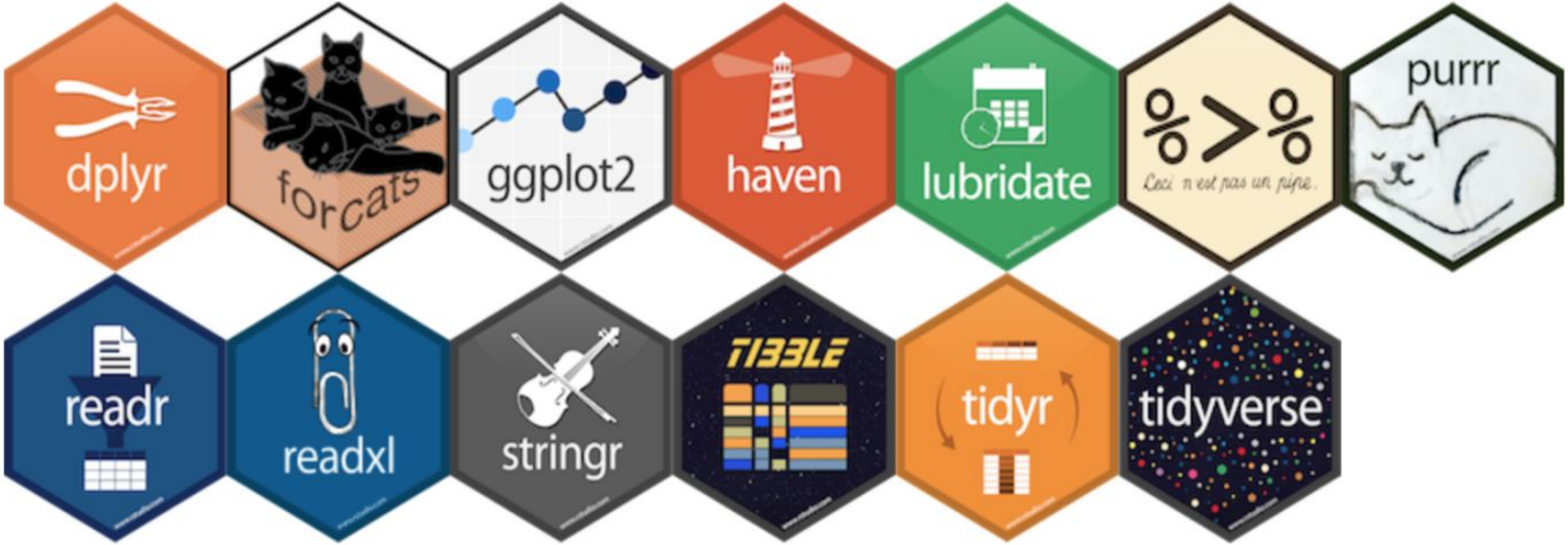
The data science process (tidied)



What is
the tidyverse?

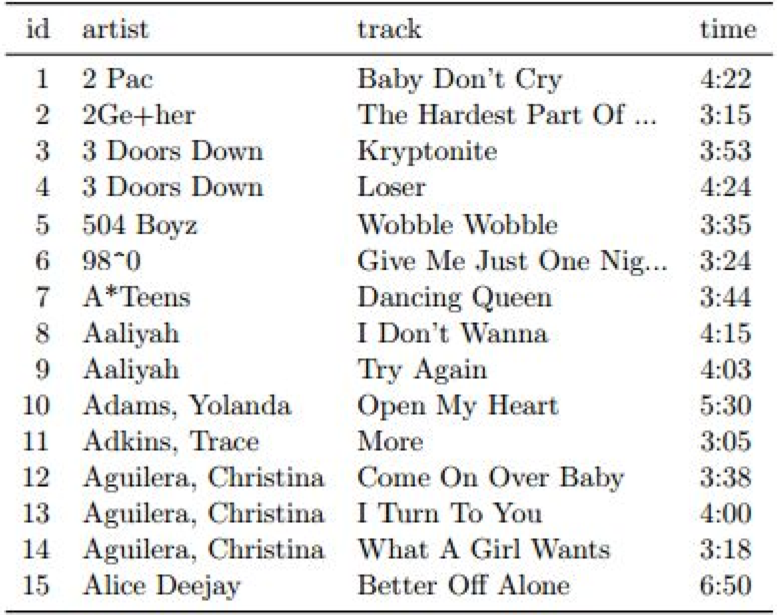
- Collection of R packages based on tidy data principles
- Designed to work together
- An easier way to code!
- AKA “Hadleyverse” (most packages written by Hadley Wickham)

What is
the tidyverse?



What is tidy data?

- Each variable is a column
- Each observation is a row
- Each type of observational unit is a table



id	artist	track	time
1	2 Pac	Baby Don't Cry	4:22
2	2Ge+her	The Hardest Part Of ...	3:15
3	3 Doors Down	Kryptonite	3:53
4	3 Doors Down	Loser	4:24
5	504 Boyz	Wobble Wobble	3:35
6	98ˆ0	Give Me Just One Nig...	3:24
7	A*Teens	Dancing Queen	3:44
8	Aaliyah	I Don't Wanna	4:15
9	Aaliyah	Try Again	4:03
10	Adams, Yolanda	Open My Heart	5:30
11	Adkins, Trace	More	3:05
12	Aguilera, Christina	Come On Over Baby	3:38
13	Aguilera, Christina	I Turn To You	4:00
14	Aguilera, Christina	What A Girl Wants	3:18
15	Alice Deejay	Better Off Alone	6:50

2.
Workflow

Workflow: Basics

You can use R as a calculator:

```
10 / 2 * 5
```

Create new objects with:

object_name <- value

```
x <- 3 * 4
```

<- has an RStudio keyboard shortcult!

`Alt--` (the minus sign)]

Workflow: Basics

Naming

The Rules:

1. Objects must start with a letter
2. Names can only contain letters, numbers, _, and .
3. Object names are case-sensitive!

```
this != THIS
```

Your Preference:

```
i_use_snake_case

otherPeopleUseCamelCase

Some.people.use.periods

And_some.People_AREweird
```

Workflow: Basics
Calling Functions

Functions are called like this:

```
    function_name(arg1 = val1, arg2 = val2,
…)
```

- Text (string) arguments go in quotes
- Use tab for auto-complete (less typing!)
- If you see “+” output, you’re probably missing a parenthesis or a quote

Workflow: Scripts

Scripts are good for code that:

- You want to re-use
- Is long or complicated

Tips:

- Start script with packages to use (`library(package_name)`)
- Script editor will highlight syntax with red squiggly line; hover to see what the issue is

Workflow: Projects

“Capture all important interactions in your code”

- Your environment is hard to replicate by memory (All packages used, the order they’re loaded, your working directory, etc.)
- Important to save everything if you want to share code

Workflow: Projects
Paths and Directories

Working directory is where R looks for files to load, and where it will write out any files you want to save.

- `getwd()` prints your current working directory
- `setwd()` allows you to set a working directory

Best practices for paths:

- Forward slashes in paths are great because a backslash is a special character for R
- Avoid absolute paths because it makes it hard to share

Workflow: Projects

Project Architecture

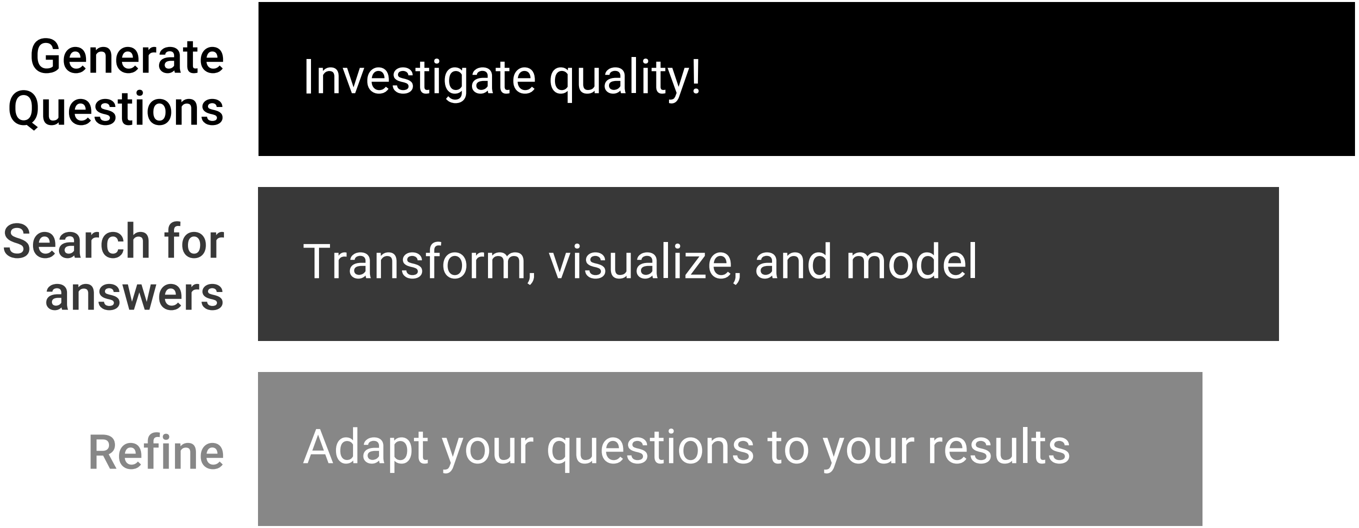
Keep all files associated with a project together -- including input files, R scripts, analytical results and deliverables.

- Can do this using RStudio Projects
- ProjectTemplate() is my favorite package for creating folders and subfolders for organizing projects

3. Exploratory data analysis

What is exploratory data analysis (EDA)?

A state of mind--are your data what you expect them to be?



Generate Questions — Investigate quality!

Search for answers — Transform, visualize, and model

Refine — Adapt your questions to your results

What you need

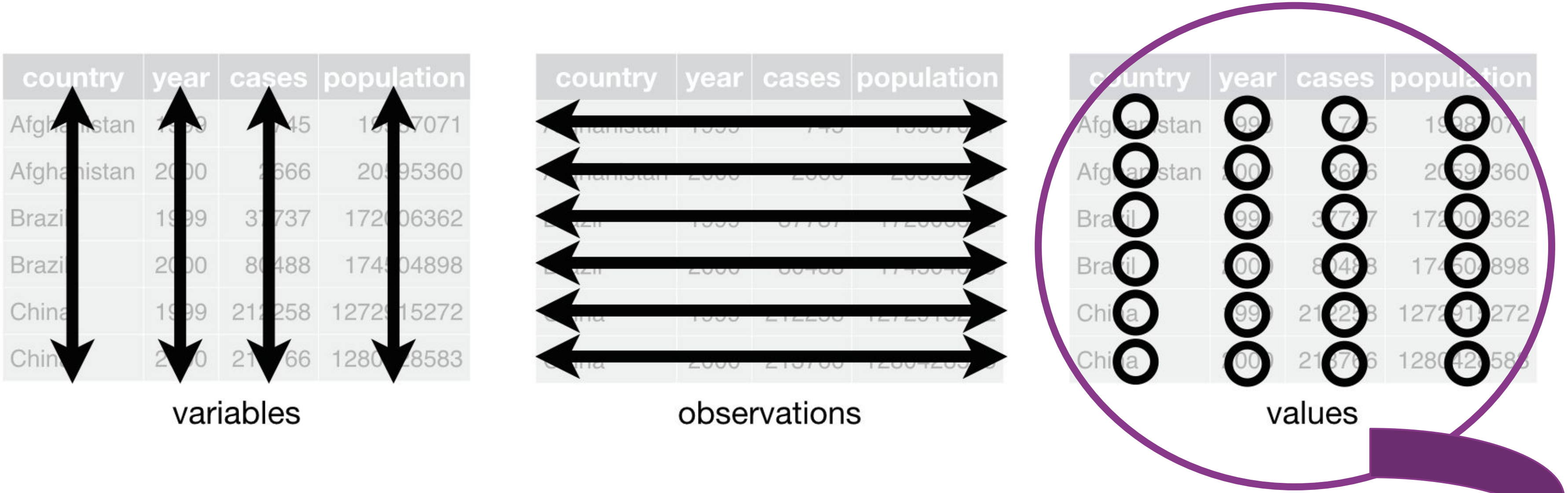
```
library(tidyverse)
```

ggplot2

dplyr

What are we looking for?

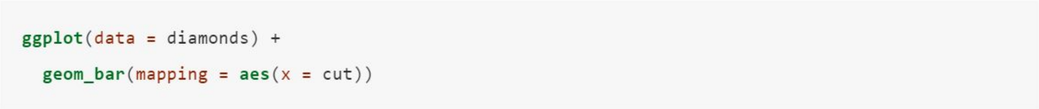
- Creative process
    a. No rules….
- In general:
    a. Variation in variables
    b. Covariation in variables (relationships)


country
year
cases
population
variables
observations
values

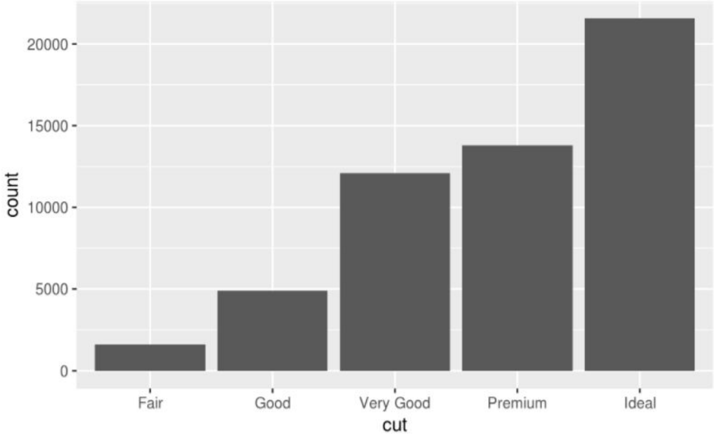
variation!

Variation- Visualizing distributions

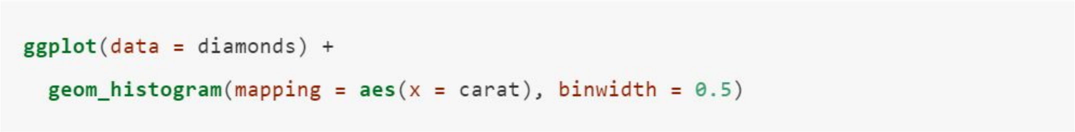
- Categorical variables = one set of values
  - In R, saved as character or factor



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



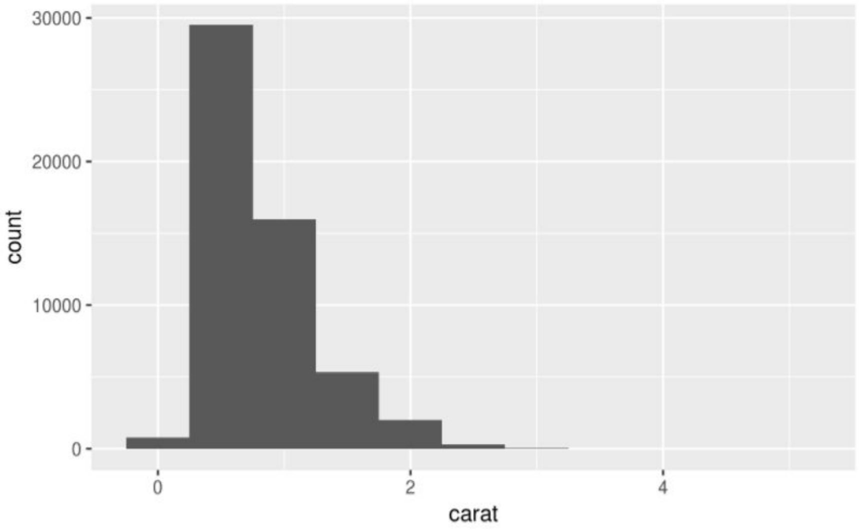
Variation-Visualizing distributions



```
ggplot(data = diamonds) +
  geom_histogram(mapping = aes(x = carat), binwidth = 0.5)
```

Continuous variable = any infinite set of ordered values

E.g. numbers, datetime



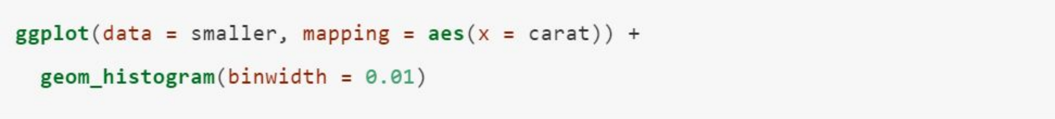
Variation-Typical values

What is common?

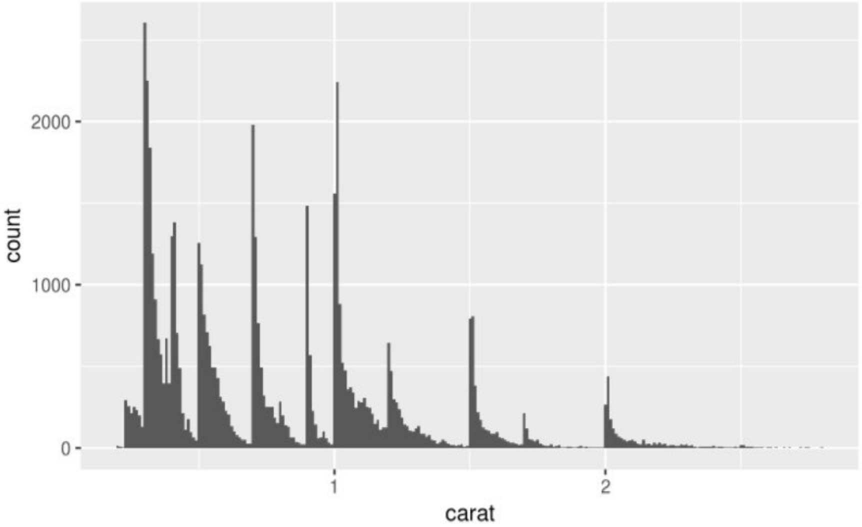
What is rare?

Any unusual patterns?

---> likely leads to questions to explore relationship *between* vars



```
ggplot(data = smaller, mapping = aes(x = carat)) +
  geom_histogram(binwidth = 0.01)
```

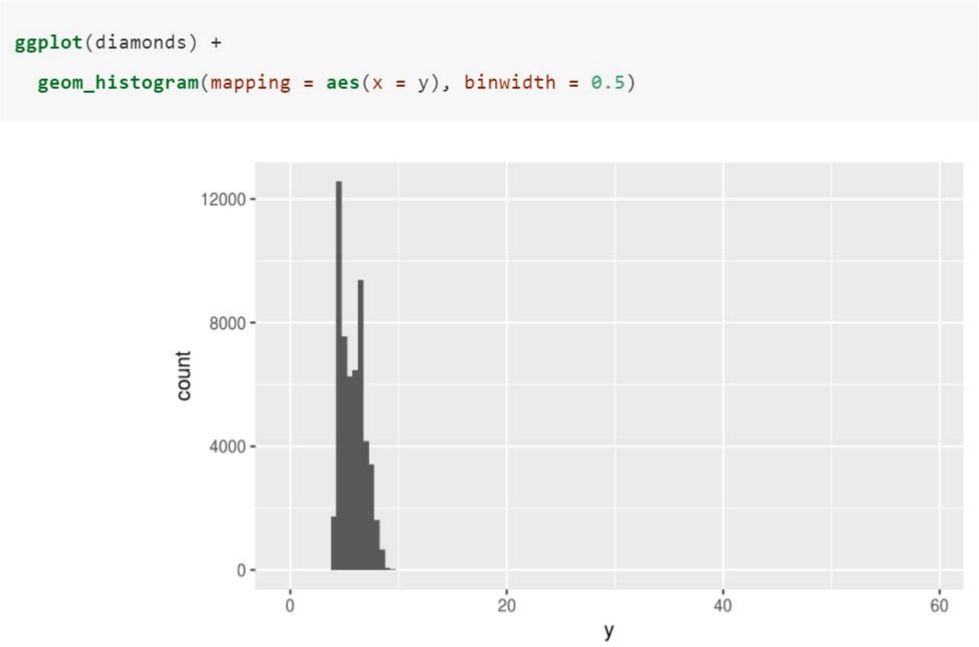


Variation-Unusual values

Outliers = unusual observations

---> can be errors, can be important

Repeat analysis without them-what happens?



```
ggplot(diamonds) +
  geom_histogram(mapping = aes(x = y), binwidth = 0.5)
```

Variation-Missing values

What do to with unusual values?

1. Drop entire row - not recommended!
2. Replace with NA (special value type)

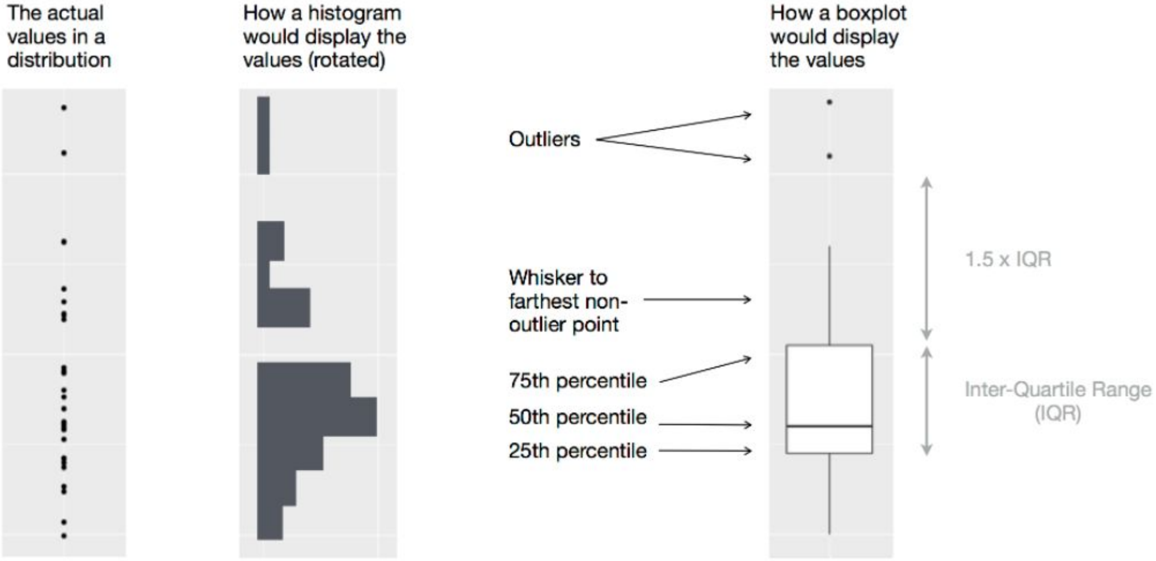
```
diamonds2<-diamonds%>%

    mutate(y= ifelse(y<3 | y>20, NA, y))
```

Note: ggplot statements will not plot NA values-you will receive a warning

Covariation- a categorical var and a continuous var

Enter the boxplot! Aka box and whisker

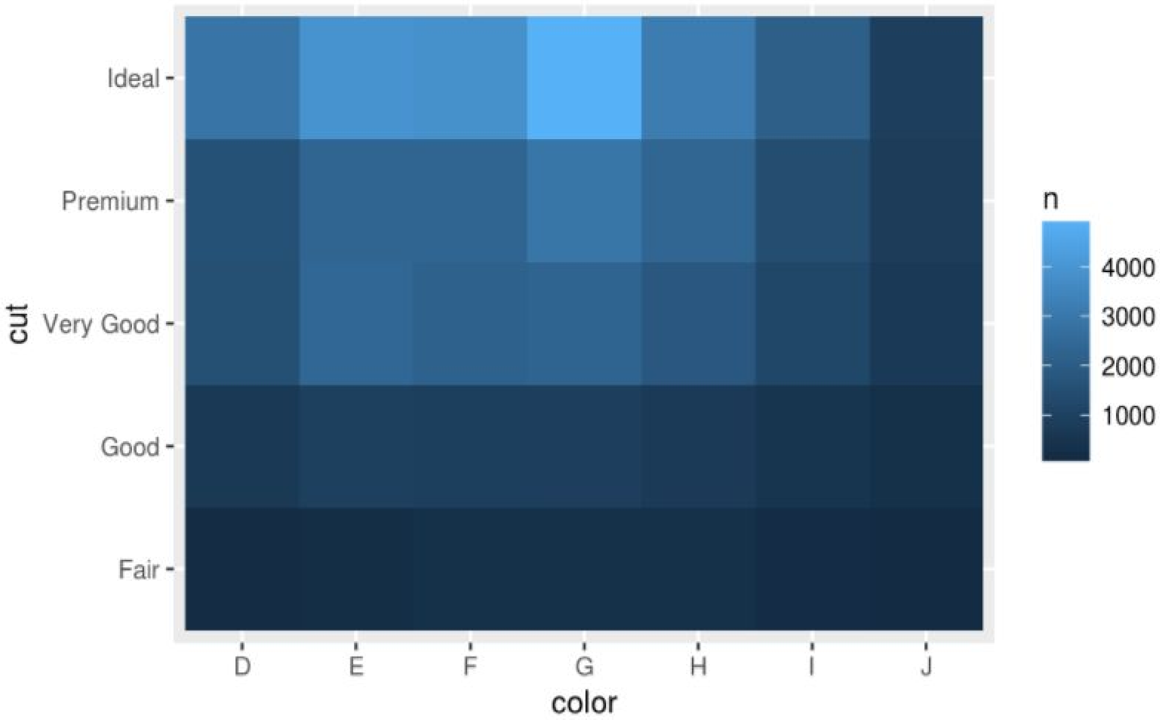
Display distribution of a continuous var broken down by a categorical var


The actual values in a distribution
How a histogram would display the values (rotated)
How a boxplot would display the values
Outliers
Whisker to farthest non-outlier point
75th percentile
50th percentile
25th percentile
1.5 x IQR
Inter-Quartile Range (IQR)

Covariation- Two categorical vars

Enter the heat map!

```
diamonds%>%
    count(color, cut)%>%
    ggplot(mapping = aes(x=color, y=cut))+
    geom_tile(mapping = aes(fill=n))
```

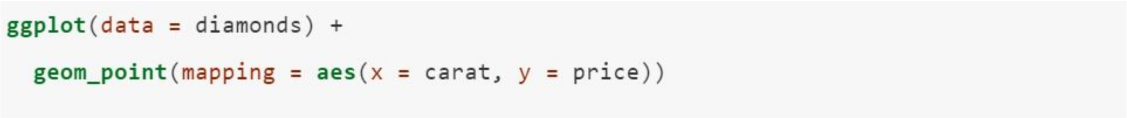

Ideal
Premium
Very Good
Good
Fair
cut
D
E
F
G
H
I
J
color
n
4000
3000
2000
1000

Covariation- Two continuous vars

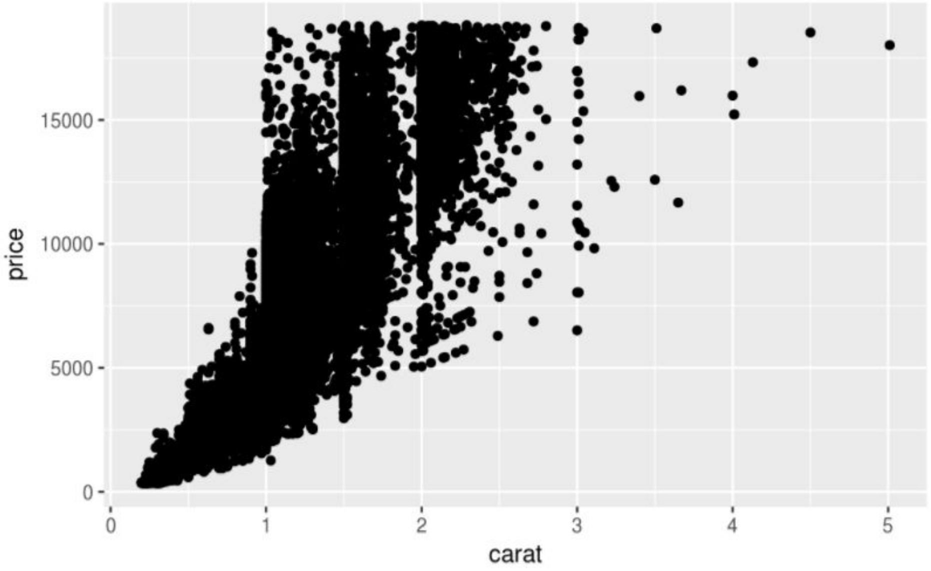
Scatterplots !

Or boxplots with

`varwidth= TRUE`



```
ggplot(data = diamonds) +
  geom_point(mapping = aes(x = carat, y = price))
```



Patterns and Models

```
library(modelr)
```

- **Models** = tool to extract patterns out of data
- View relationships once effects are adjusted for
    - Residuals = residual variation after adjusting for factors included in a model

```
mod<- lm(log(price) ~ log(carat), data=diamonds)
```

- We will learn more later!

4.
Wrap-up

Announcements, upcoming events, etc.

R-Ladie Austin
Upcoming Events

Book Club: Dear Data [November 8]

R for Data Science Workshop: Data Wrangling in the Tidyverse [Nov 29]

All The Ladies in Tech Happy Hour! [December 5]

Looking for presenters: Workshop on package development