R4DS

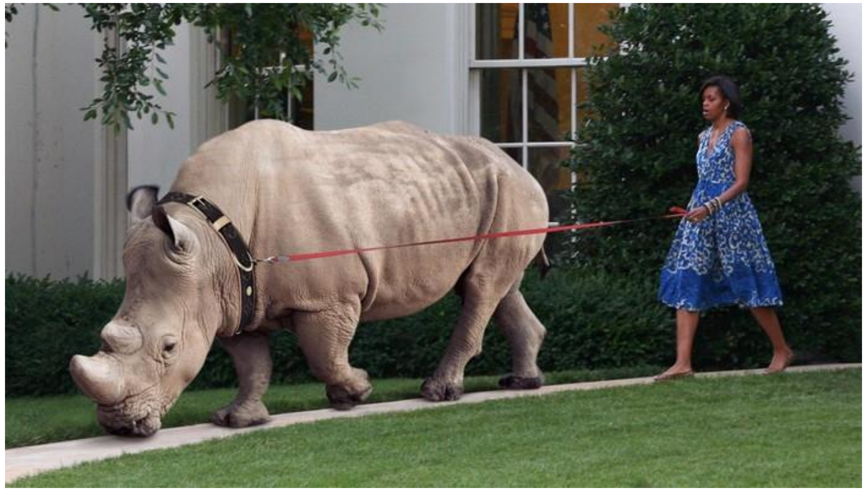
Cohort 4

Wed 6:00 – 7:00 US Central

Twitter: @Rspjut

5-MINUTE ICE BREAKER

What is your pet situation?



AGENDA

- 5-Minute Ice breaker
- Quick Housekeeping Reminders
- Chapter 7 Begin
- Getting Help
- Next Week

QUICK HOUSEKEEPING REMINDERS

- Video camera is optional, but encouraged.
- I purposely err on the side of going fast. Slowing me down does not hurt my feelings.
- Take time to learn the theory (Grammar of Graphics, Tidy Data whitepaper, Relational Database theory, etc.).
- Please do the chapter exercises. Second-best learning opportunity!
- Please plan on teaching one of the lessons. Best learning opportunity!

EXPLORATORY DATA ANALYSIS

> *This chapter will show you how to use visualisation and transformation to explore your data in a systematic way, a task that statisticians call exploratory data analysis, or EDA for short. EDA is an iterative cycle. You:*
>
> *1. Generate questions about your data.*
>
> *2. Search for answers by visualising, transforming, and modelling your data.*
>
> *3.Use what you learn to refine your questions and/or generate new questions.*
>
> *- Wickham and Grolemund, Section 7.1*

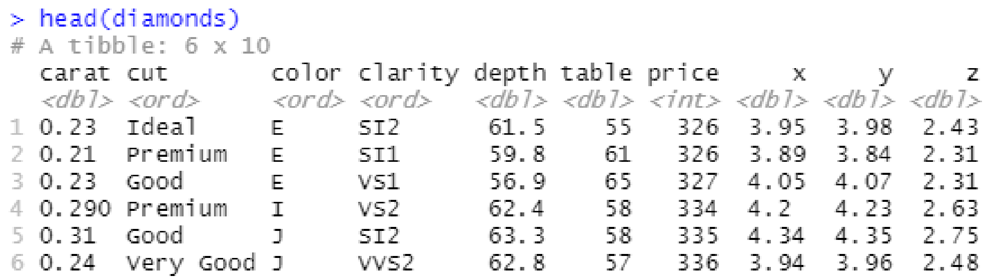
> *Your goal during EDA is to develop an understanding of your data… There is no rule about which questions you should ask to guide your research. However, two types of questions will always be useful for making discoveries within your data. You can loosely word these questions as:*
>
> *1. What type of variation occurs within my variables?*
>
> *2. What type of covariation occurs between my variables?*
>
> *- Wickham and Grolemund, Section 7.2*

DATASET USED IN CHAPTER 7: DIAMONDS

Diamonds (load `tidyverse` then `?diamonds`)

Variable	Format
price	Price in US dollars
carat	Weight of the diamond (0.2 – 5.01)
cut	Quality of the cut (Fair, Good, Very Good, Premium, Ideal)
color	Diamond color from D (best) to J (worst)
clarity	How clear. Worst = I1, SI2, SI1, VS2, VS1, VVS2, VVS1, IF
x	Length in mm
y	Width in mm
z	Depth in mm
depth	Depth percentage
table	Width of top of diamond relative to widest point

`head(diamonds)`



```
> head(diamonds)
# A tibble: 6 x 10
  carat cut       color clarity depth table price     x     y     z
  <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
1 0.23  Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
2 0.21  Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
3 0.23  Good      E     VS1      56.9    65   327  4.05  4.07  2.31
4 0.290 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
5 0.31  Good      J     SI2      63.3    58   335  4.34  4.35  2.75
6 0.24  Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
```

Total Records = 53,940

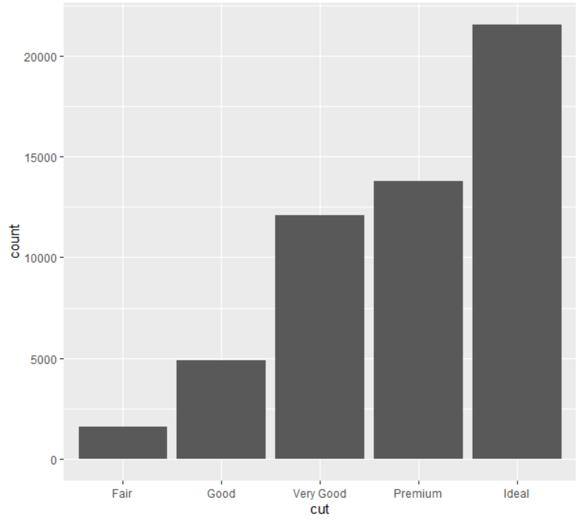
7.3.1 VISUALIZING DISTRIBUTIONS

Using `diamonds`, let's visualize the number of diamonds that belong to each value of the `cut` variable.

First, what are the values of the `cut` variable?

Variable	Format
cut	Quality of the cut (Fair, Good, Very Good, Premium, Ideal)

What type of geom would work best for this?
(`cut` is a categorical variable!)



Create the graph in R.

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

dataset

geom type

variable

7.3.1 VISUALIZING DISTRIBUTIONS

Using `diamonds`, let’s visualize the number of diamonds that belong to each value of the `carat` variable.

First, what are the values of the `carat` variable?

Variable	Format
carat	Weight of the diamond (0.2 – 5.01)

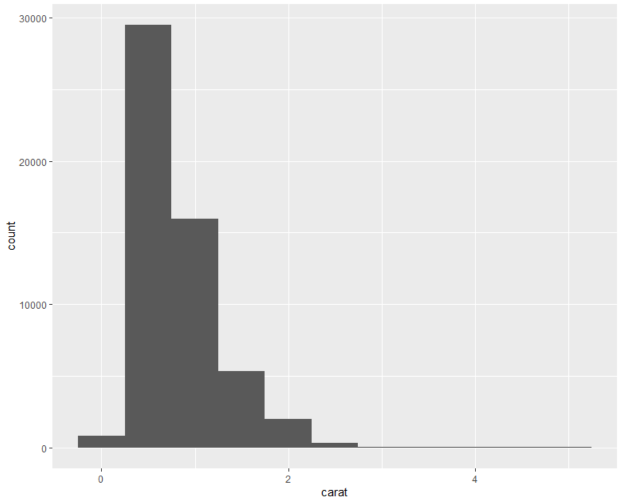
What type of geom would work best for this?
(`carat` is a continuous variable!)

- Will 2.0 carats and 3.0 carats be in the same bin?
- What about 2.1 carats and 2.2 carats?

Create the graph in R.

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

dataset geom type variable argument



7.3.1 VISUALIZING DISTRIBUTIONS

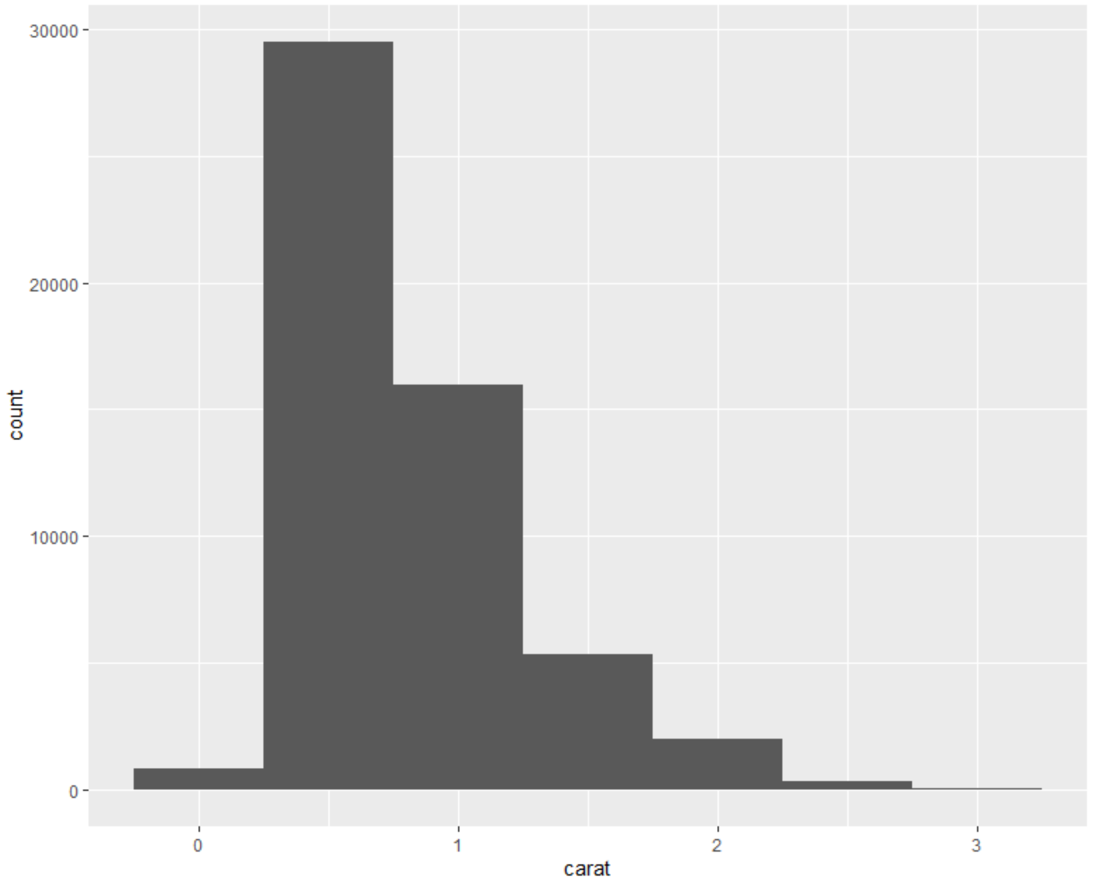
Notice that there are very few diamonds larger than 2.5 carats.

Let’s filter our dataset to only include diamonds under 3.0 carats.

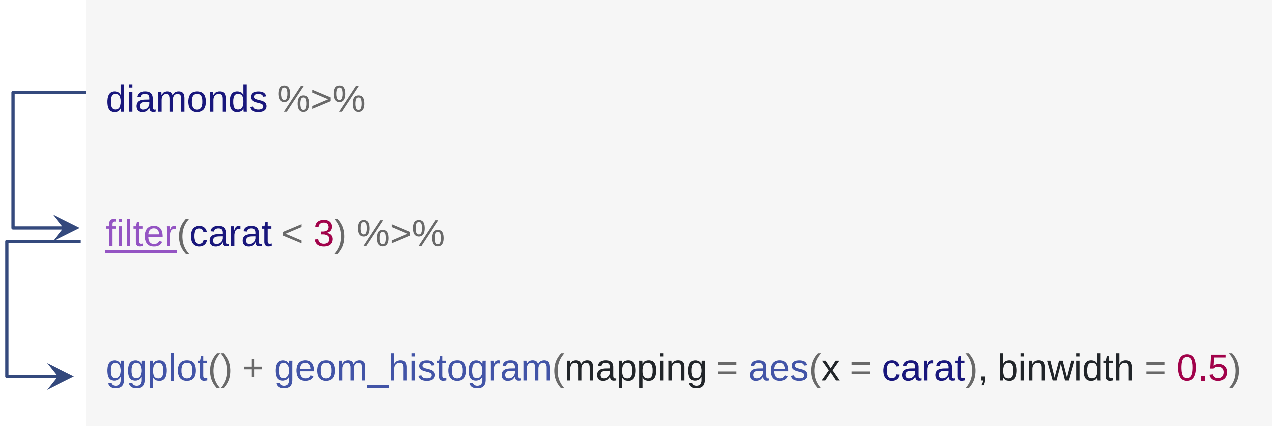
Think about how to do this with the pipe operator (`%>%`).

- What dataset would you start with?
- What do you do to it next (i.e., “and then” = pipe)?
- What dplyr verbs do you need?
- How do you research the arguments your verb needs?
- Graphing with `ggplot` can follow a pipe operator!

```
diamonds %>%
filter(carat < 3) %>%
ggplot() + geom_histogram(mapping = aes(x = carat), binwidth = 0.5)
```



THE PIPE %>% IN ACTION

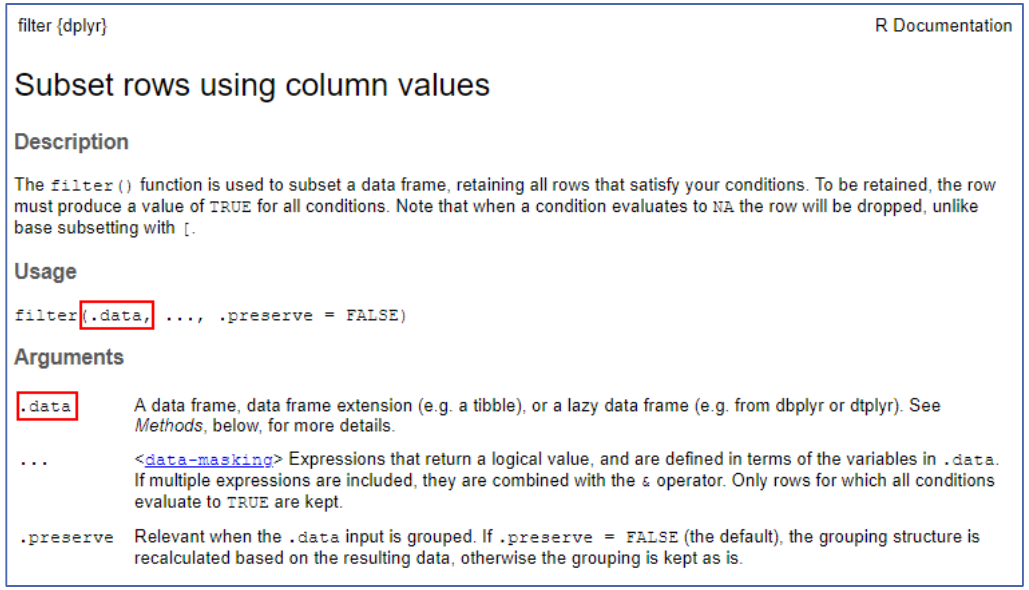


```
diamonds %>%

filter(carat < 3) %>%

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

Pipe moves results to next step.

Don't repeat the data argument.

What arguments does the `filter` verb take?



filter {dplyr} R Documentation

Subset rows using column values

Description

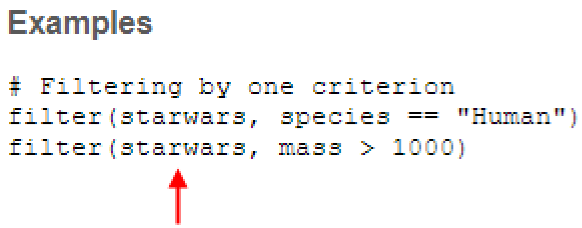
The `filter()` function is used to subset a data frame, retaining all rows that satisfy your conditions. To be retained, the row must produce a value of `TRUE` for all conditions. Note that when a condition evaluates to `NA` the row will be dropped, unlike base subsetting with `[`.

Usage

```
filter(.data, ..., .preserve = FALSE)
```

Arguments

`.data`	A data frame, data frame extension (e.g. a tibble), or a lazy data frame (e.g. from dbplyr or dtplyr). See *Methods*, below, for more details.
`...`	<`data-masking`> Expressions that return a logical value, and are defined in terms of the variables in `.data`. If multiple expressions are included, they are combined with the `&` operator. Only rows for which all conditions evaluate to `TRUE` are kept.
`.preserve`	Relevant when the `.data` input is grouped. If `.preserve = FALSE` (the default), the grouping structure is recalculated based on the resulting data, otherwise the grouping is kept as is.



Examples

```
# Filtering by one criterion
filter(starwars, species == "Human")
filter(starwars, mass > 1000)
```

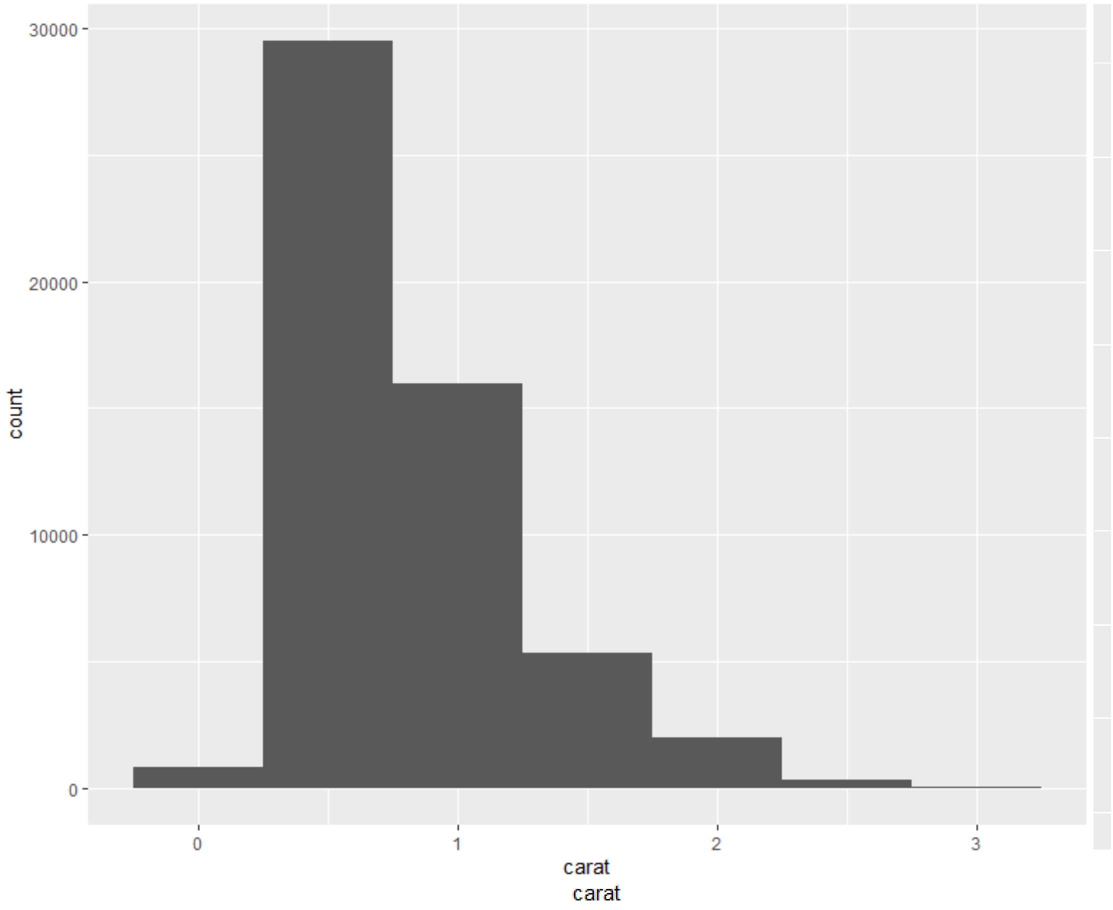
7.3.1 VISUALIZING DISTRIBUTIONS

What if you reduce the `binwidth` from `0.5` to `0.1`?

```
diamonds %>%
filter(carat < 3) %>%
ggplot() + geom_histogram(mapping = aes(x = carat), binwidth = 0.1)
```

Instead of each bar showing the number of diamonds from `0.0` to `0.5` carats, and `0.5` to `1.0` carats…

Each bar shows the number of diamonds from `0.0` to `0.1` carats, and `0.1` to `0.2` carats, …



7.3.1 VISUALIZING DISTRIBUTIONS

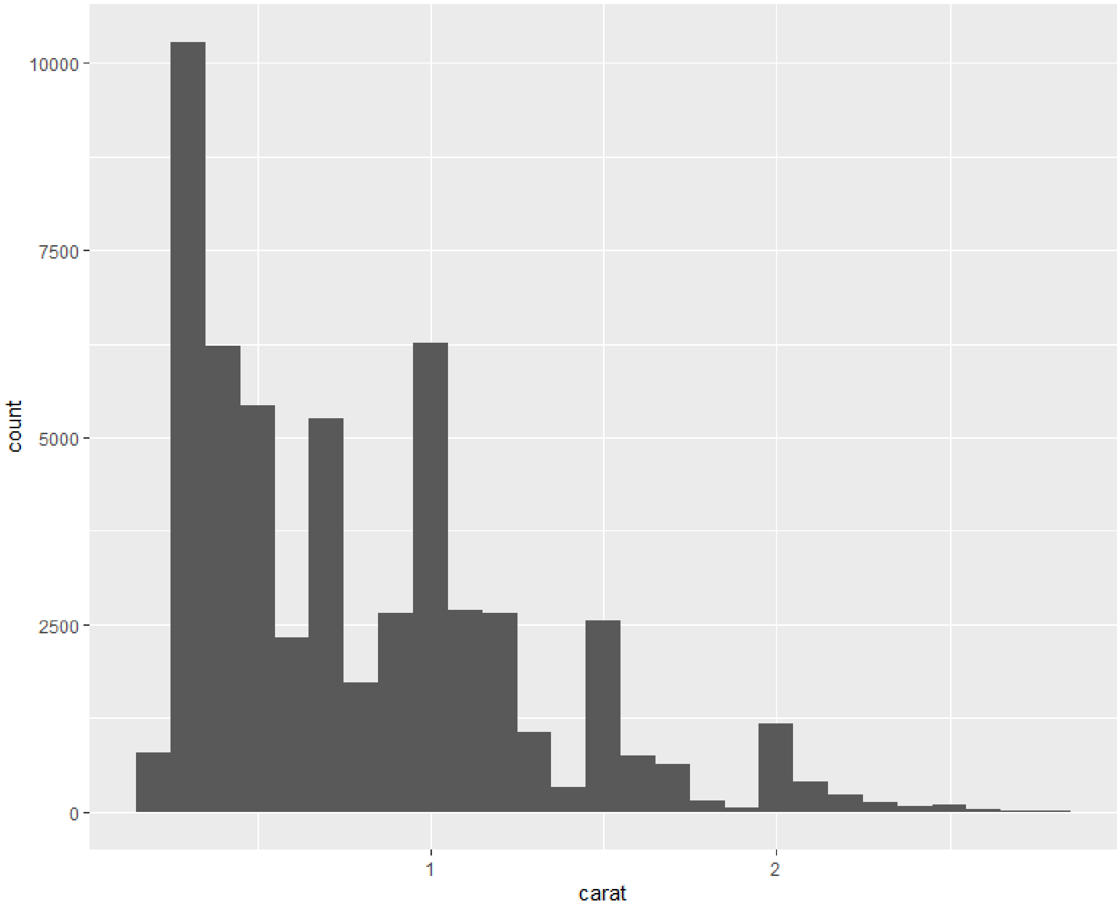
What if we want to fill the histogram bars with color based on `cut`?

```
diamonds %>%
filter(carat < 3) %>%
ggplot() + geom_histogram(mapping = aes(x = carat), binwidth = 0.1)
```

Q: What do you call a graphical element (like `fill`) in `ggplot`?
A: `Aesthetic`

Q: How do you assign the `fill` aesthetic to be mapped to the `cut` variable?
[✓ A: `fill = cut`] B: `cut = fill` C: `fill(cut)` D: `cut(fill)`

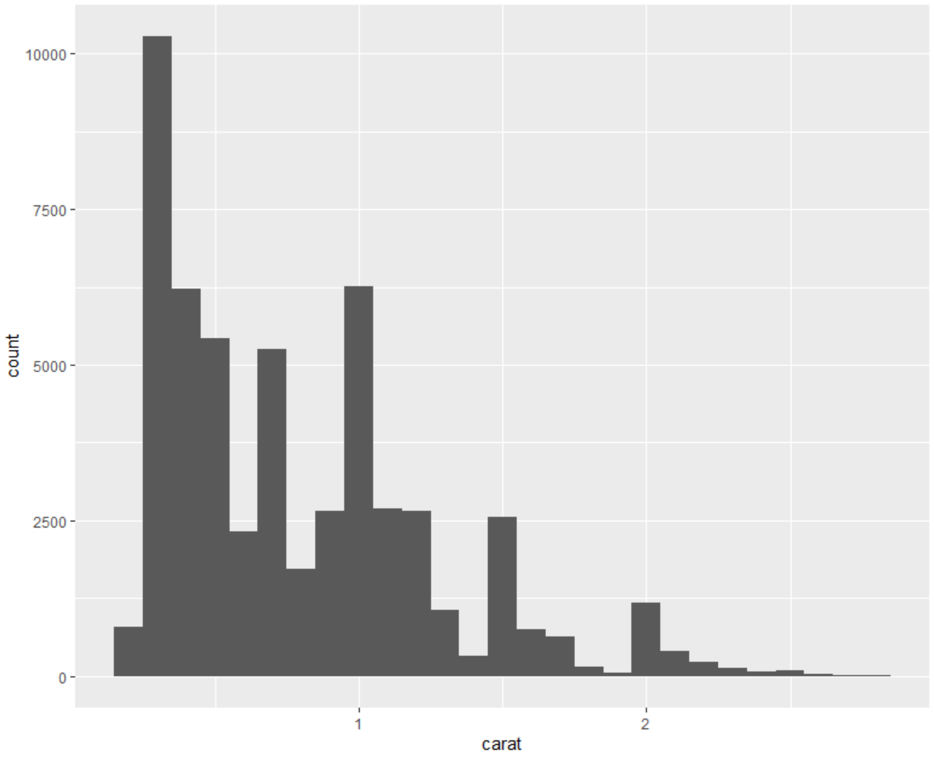
Q: Does `fill = cut` goes inside the `aes()` or outside the `aes()`? Why?
A: Inside the `aes()` because we want the fill colors to map to values of `cut`.

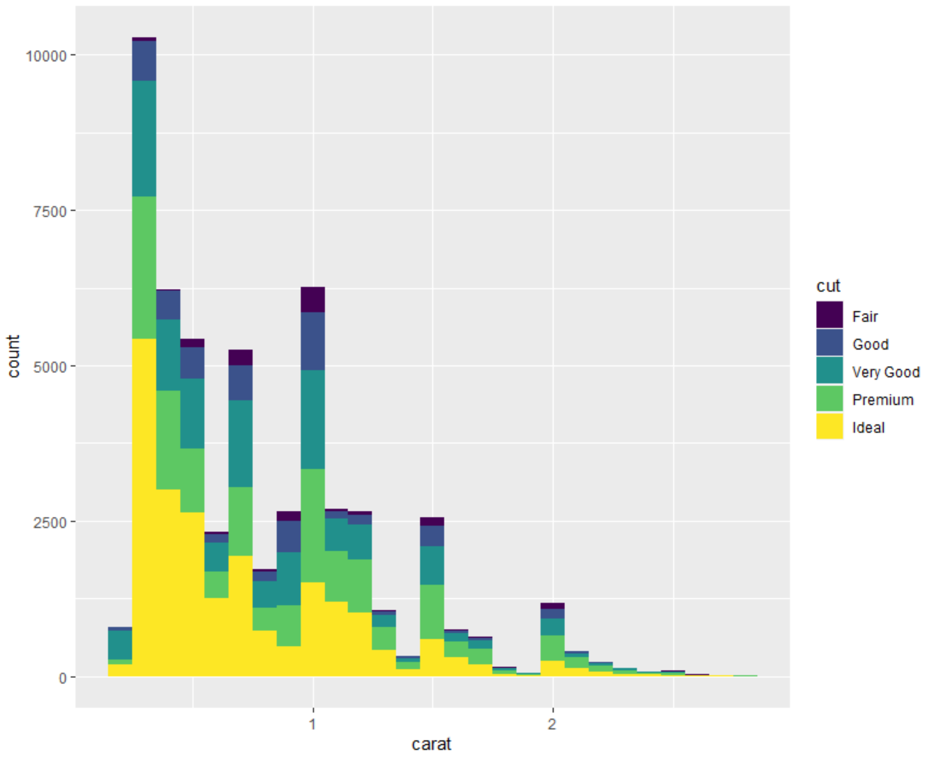


7.3.1 VISUALIZING DISTRIBUTIONS

```
diamonds %>%
filter(carat < 3) %>%
ggplot() + geom_histogram(mapping = aes(x = carat), binwidth = 0.1)
```

```
diamonds %>%
filter(carat < 3) %>%
ggplot() + geom_histogram(mapping = aes(x = carat, fill = cut), binwidth = 0.1)
```

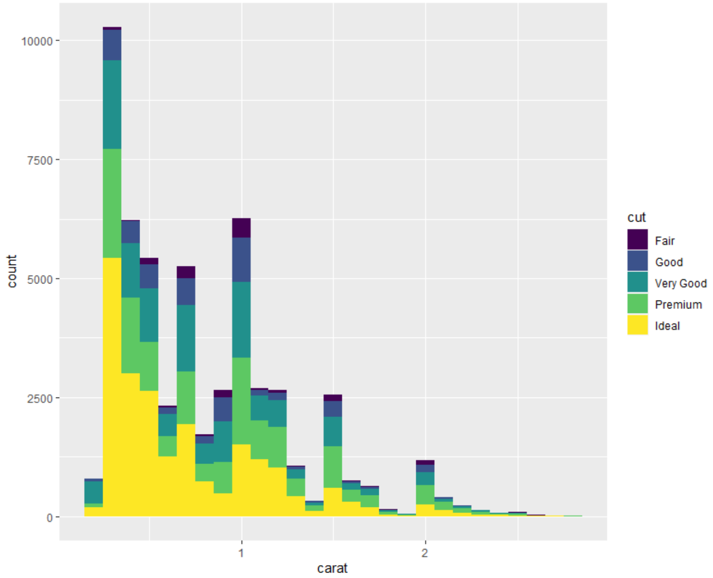




7.3.1 VISUALIZING DISTRIBUTIONS

This stacked bar chart visualization is poor because most categories of `cut` do not start at zero.

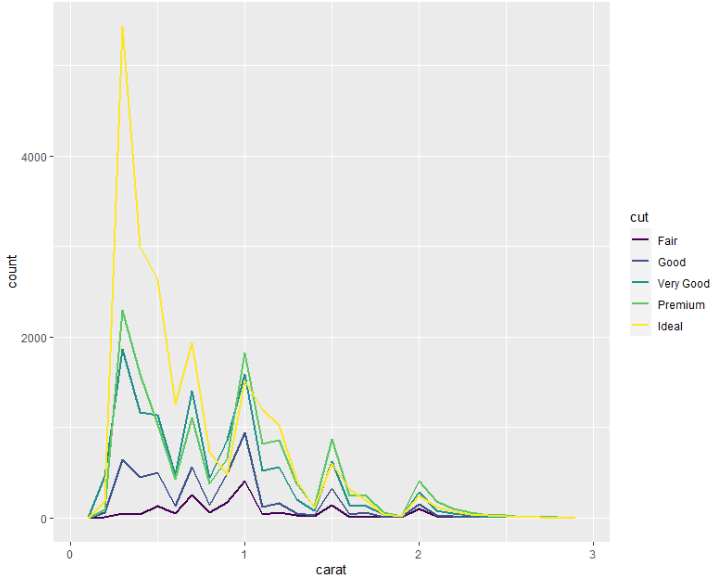
```
diamonds %>%
filter(carat < 3) %>%
ggplot() + geom_histogram(mapping = aes(x = carat, fill = cut), binwidth = 0.1)
```



The `freqpoly` visualization is better.

This time, the aesthetic name is `color` instead of `fill`.

```
diamonds %>%
filter(carat < 3) %>%
ggplot() + geom_freqpoly(mapping = aes(x = carat, color = cut), binwidth = 0.1)
```



GETTING HELP

- Ask questions during our call
- Google
- Stack Overflow
- Slack
- Office Hours r4ds.io/calendar
- Twitter #rstats
- r4ds answer keys: Jeff Arnold (preferred) or Bryan Shalloway (also good)
- Cheatsheets

NEXT WEEK…

- Continue Chapter 7: Exploratory Data Analysis
  - Section 7.3.3 on zooming in to sections of the plot
  - Carry on with Section 7.4: Missing Values

RStudio®