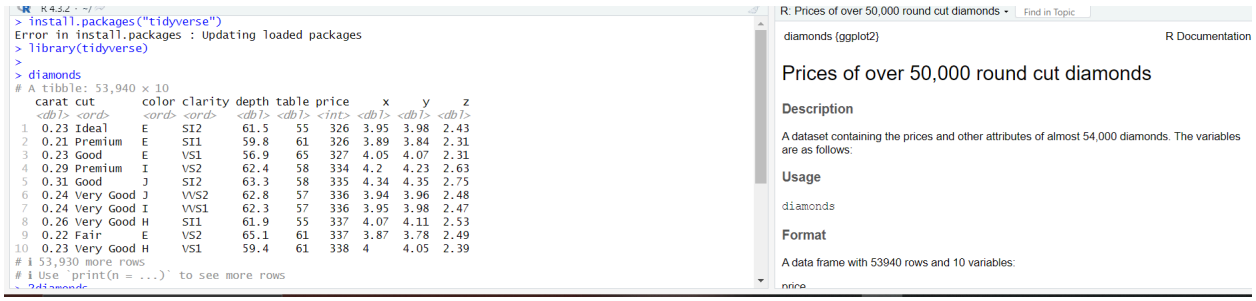
EXPLORATORY DATA ANALYSIS

DSC1105

Formative Assessment 3

BILLONES, Cristel Kaye P.
2021016541

```
install.packages("tidyverse")
library(tidyverse)

diamonds
?diamonds
head(diamonds)
```



```
> install.packages("tidyverse")
Error in install.packages : Updating loaded packages
> library(tidyverse)
>
> diamonds
# A tibble: 53,940 × 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  0.29 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
 7  0.24 Very Good I     VVS1     62.3    57   336  3.95  3.98  2.47
 8  0.26 Very Good H     SI1      61.9    55   337  4.07  4.11  2.53
 9  0.22 Fair      E     VS2      65.1    61   337  3.87  3.78  2.49
10  0.23 Very Good H     VS1      59.4    61   338  4     4.05  2.39
# ℹ 53,930 more rows
# ℹ Use `print(n = ...)` to see more rows
```

R: Prices of over 50,000 round cut diamonds — Find in Topic

diamonds {ggplot2} — R Documentation

Prices of over 50,000 round cut diamonds

Description

A dataset containing the prices and other attributes of almost 54,000 diamonds. The variables are as follows:

Usage

```
diamonds
```

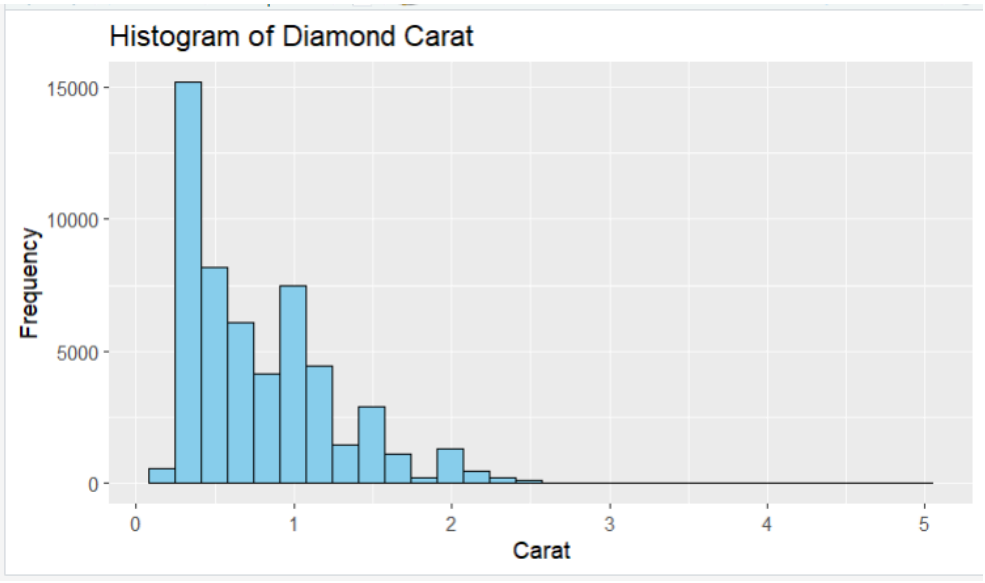
Format

A data frame with 53940 rows and 10 variables:

price

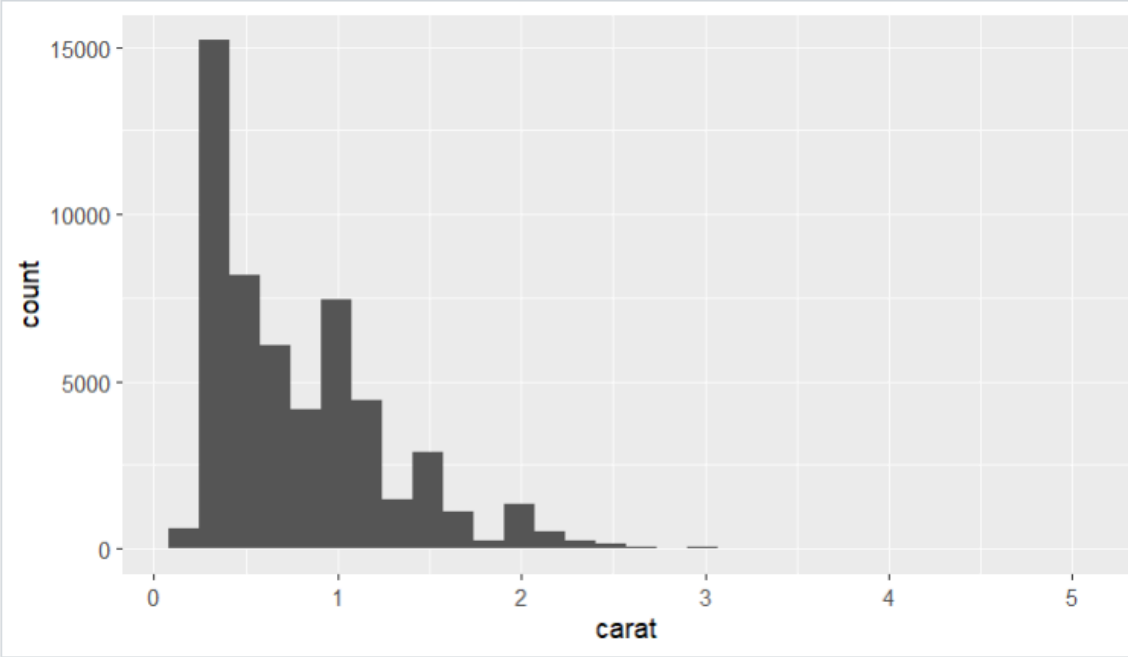
1. Create a histogram on the diamonds dataset, for example with

ggplot() + geom_histogram(aes(x = carat), data = diamonds)



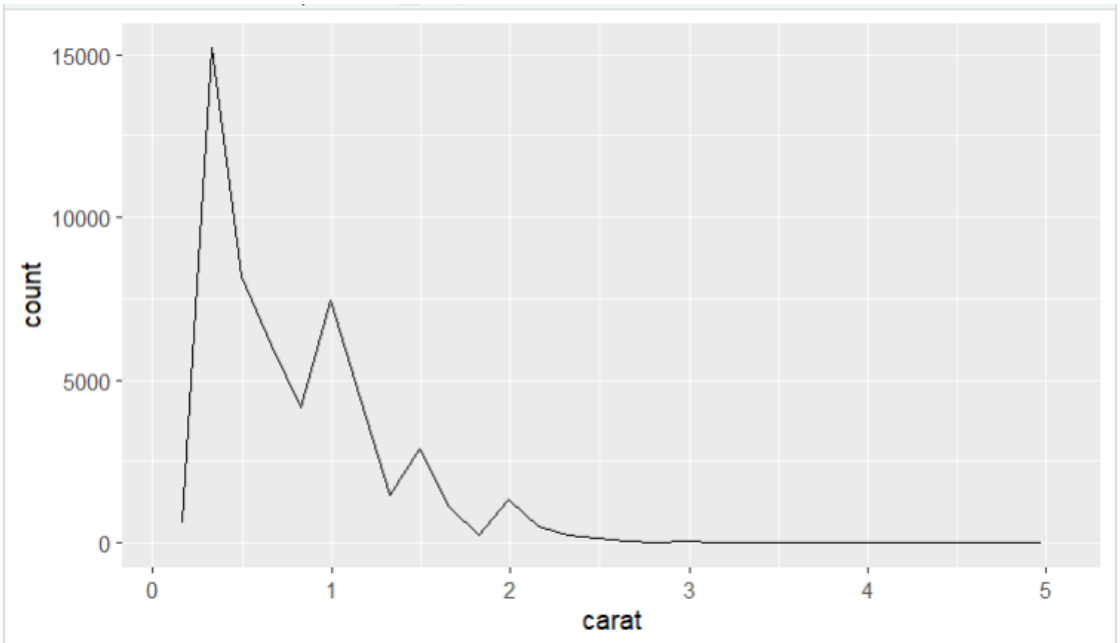
Re-write this using the layer function like we did in class. Hint: if you don't know the default values for some of the aspects of the plot, examine p$layers.

```
library(ggplot2)
# Base plot

p <- ggplot(data = diamonds)


# Add the histogram layer using geom_bar

p <- p + layer(

  data = diamonds,

  stat = "bin",

  geom = "bar",

  position = "identity",  # Specify position argument

  mapping = aes(x = carat)

)

print(p)
```



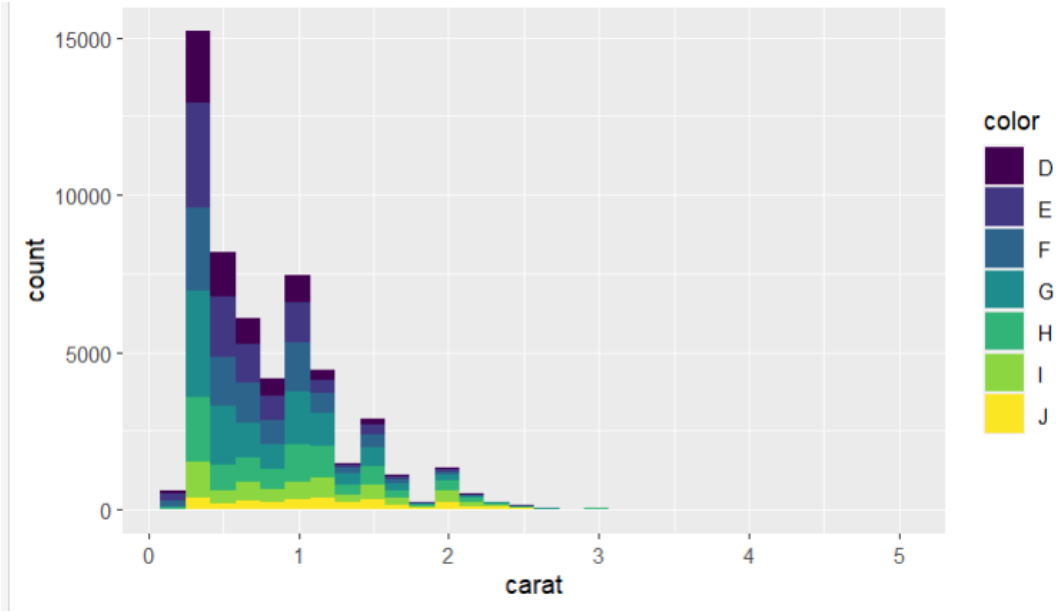
2. Remember that a histogram is a plot with stat_bin and geom_bar. Modify your histogram code so that it uses a different geom, for example geom_line or geom_point. This should be simple once you have the layer specification of a histogram.

```
# Create the base plot

p <- ggplot(data = diamonds)


# Add the histogram layer using geom_bar

p <- p + layer(

  data = diamonds,

  stat = "bin",

  geom = "line",

  position = "stack",  # Specify position argument

  mapping = aes(x = carat)

)

print(p)
```



3. In your histogram (the one plotted with bars that you created in question 1), add an aesthetic mapping from one of the factor variables (maybe color or clarity) to the fill or color aesthetic.

```
# Base plot

p <- ggplot(data = diamonds)


# Add the histogram layer using geom_bar

p <- p + layer(

  data = diamonds,

  stat = "bin",

  geom = "bar",

  position = "stack",  # Specify position argument

  mapping = aes(x = carat, fill = color)  # Add aesthetic mapping

)

print(p)
```



4. What is the default position adjustment for a histogram? Try changing the position adjustment in the histogram you created in question 3 to something different (hint: try dodge).

```
# Base plot

p <- ggplot(data = diamonds)


# Add the histogram layer using geom_bar

p <- p + layer(

  data = diamonds,

  stat = "bin",

  geom = "bar",

  position = "dodge",  # Change position adjustment to "dodge"

  mapping = aes(x = carat, fill = color)  # Add aesthetic mapping

)


print(p)
```