

---

## FORMATIVE ASSESSMENT 3

### DSC1105

February 28, 2025

Zion John Yousef T. Ramilo

---

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 what the default values for some of the aspects of the plot, examine*

`ls("package:ggplot2", pattern = "^Position").`

```
# Original Shorthand in GGplot
library(tidyverse)

ggplot(diamonds, aes(x=carat)) +
  geom_histogram() +
  labs(
    title = "Frequency per Carat"
  ) +
  ylab("Frequency")

# Rewrite the ggplot function into a layer version

ggplot(diamonds) +
  layer(
    geom = GeomBar, stat = StatBin, position = PositionStack,
    params = list(binwidth = 0.1, na.rm = FALSE),
    mapping = aes(x=carat)
  ) +
  labs(
    title = "Frequency per Carat",
    subtitle = "Histogram"
  ) +
  ylab("Frequency") +
  xlab("Carat") +
  theme_classic()
```

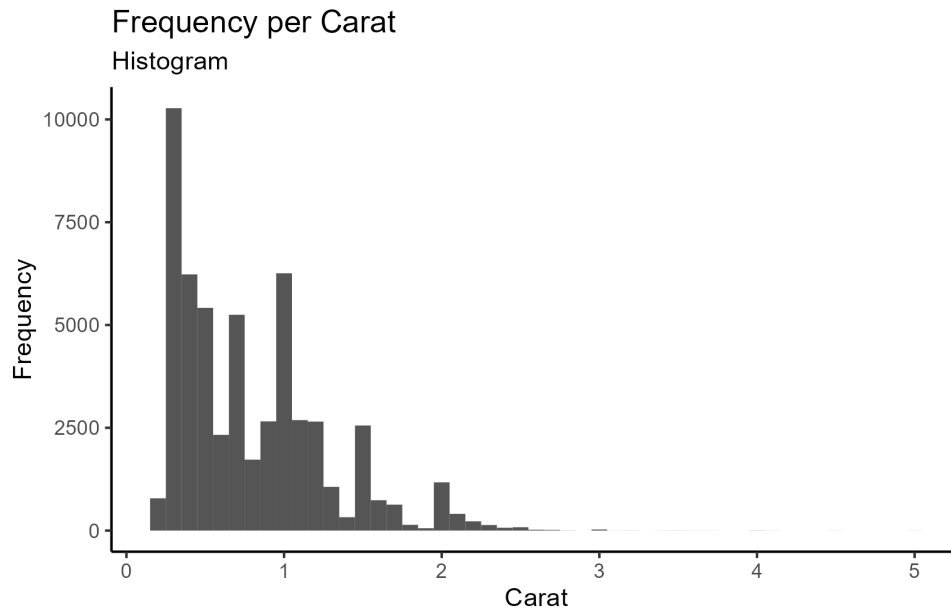
Remarks:

Using the following commands:

- `ls("package:ggplot2", pattern = "^Position")`
- `ls("package:ggplot2", pattern = "^Geom")`
- `ls("package:ggplot2", pattern = "^Stat")`

We can identify values that can be taken upon the arguments within the layer function of GGplot wherein to identify the given geom is by the last string of the shorthand command whilst for arguments of stat and position values for the given geom the command `?geom_*` was used.

Plot Output:



- 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.

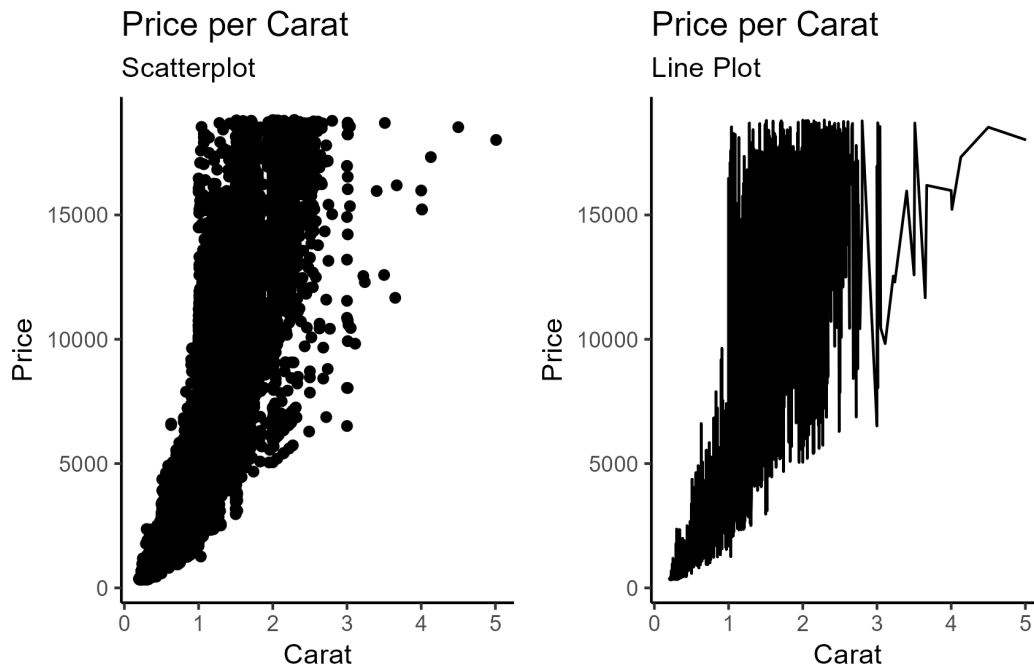
```
# Different plots using Layer
# Scatterplot
ggplot(diamonds)+
  layer(
    geom = GeomPoint, stat = "identity", position = "identity",
    params = list(na.rm = FALSE),
    mapping = aes(x=carat,y=price)
  )+
  labs(
    title = "Price per Carat",
    subtitle = "Scatterplot "
  )+
  ylab("Price")+
  xlab("Carat")+
  theme_classic()

# Line Plot
ggplot(diamonds)+
  layer(
    geom = GeomLine, stat = "identity", position = "identity",
    params = list(na.rm = FALSE),
    mapping = aes(x=carat,y=price)
  )+
  labs(
    title = "Price per Carat",
    subtitle = "Line Plot"
  )+
  ylab("Price")+
  xlab("Carat")+
  theme_classic()
```

### Remarks:

Using the similar methods presented within the first number of the assessment we have been able to identify the given default values for the given plot behavior.

Plot Output:



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.

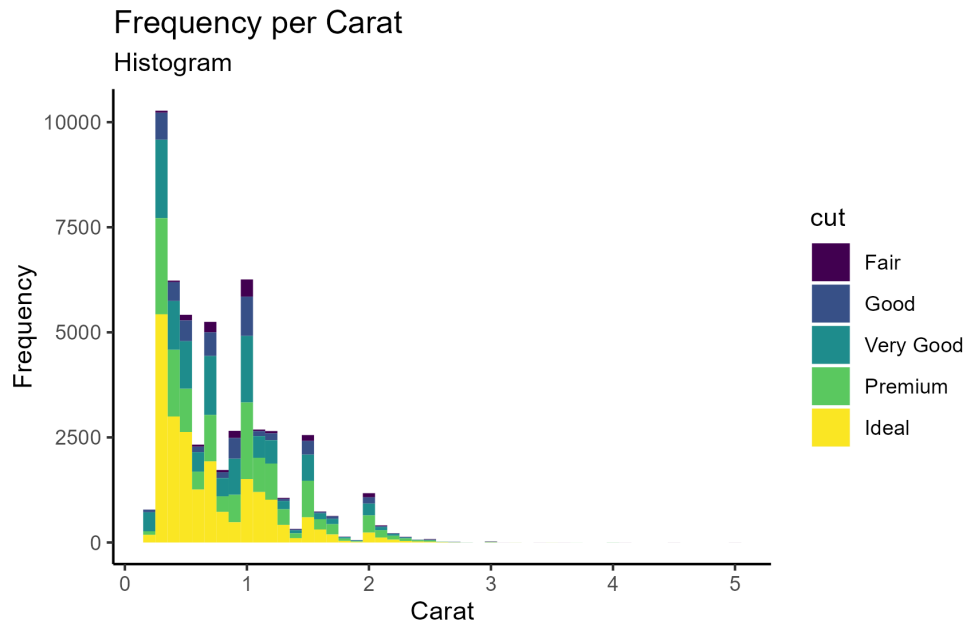
```
# Adding Fill to the layer function for histogram plot

ggplot(diamonds)+
  layer(
    geom = GeomBar, stat = StatBin, position = PositionStack,
    params = list(binwidth = 0.1, na.rm = FALSE),
    mapping = aes(x=carat,fill=cut)
  )+
  labs(
    title = "Frequency per Carat",
    subtitle = "Histogram"
  )+
  ylab("Frequency")+
  xlab("Carat")+
  theme_classic()
```

### Remarks:

To add fill for a variable within the data set we have adjusted the value of the mapping argument of the layer function to have an aesthetic of fill with a value of cut where it will color the values of each category within the cut column.

Plot Output:



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).

```
# Different position value

ggplot(diamonds)+
  layer(
    geom = GeomBar, stat = StatBin, position = PositionDodge,
    params = list(binwidth = 0.1, na.rm = FALSE),
    mapping = aes(x=carat,fill=cut)
  )+
  labs(
    title = "Frequency per Carat",
    subtitle = "Histogram"
  )+
  ylab("Frequency")+
  xlab("Carat")+
  theme_classic()
```

#### Remarks:

By the documentation of GGplot position is how the data is positioned where it is mainly to avoid overlapping of datapoints, with using a different position as to the default position value which in this case is `PositionDodge` the data points are shifted to separate each value within the cut variable of the dataset.

Plot Output:

