STAT167 HW3 - Spring 2025

Ethan Choi

Contents

Homework #3 instructions

Review textbook Chapter 9 "Visualize > Layers", Chapter 10 "Visualize > Exploratory data analysis", and the lecture notes on `ggplot2` before answering the homework questions.

This homework contains 2 questions, each with multiple parts, 100 points in total.

Replace **INSERT_YOUR_ANSWER** with your own answers.

- First open this `rmd` file in RStudio and click `Knit -> Knit to PDF` to render it to PDF format. You need to have `LaTex` installed on the computer to render it to PDF format. If not, you can also render it to HTML format.
- It is best to read this `rmd` file and the rendered `pdf/html` file side-by-side, while you are working on this homework.
- If the question asks you to write some R code, remember to put your code into a **R code chunk**. Make sure both your R code chunk and its output are visible in the rendered `pdf/html` file.
- For this homework, use **ggplot2** to visualize your data. Do **NOT** use R base graphics functions.
- **Please comment your R code thoroughly, and follow the R coding style guideline (https://google.github.io/styleguide/Rguide.xml). Partial credit will be deducted for insufficient commenting or poor coding styles.**
- If you have any question about this homework assignment, we encourage you to post it on **Piazza**.

Homework submission guideline

- **This homework is DUE at *11:59 PM* on *Sunday April 27, 2025*.**
- Late submission penalties.

  - Submissions up to 24 hours late will incur a 10% deduction.
  - Submissions up to 48 hours late will incur a 30% deduction.
- **If you are using one or both of your free late days, please state here: INSERT_YOUR_ANSWER**
- After you complete all questions, save your `rmd` file to `FirstnameLastname-SID-HW3.rmd` and save the rendered pdf file to `FirstnameLastname-SID-HW3.pdf`. If you can not knit it to pdf, knit it to html first and then print/save it to pdf format.
- Submit **BOTH your source `rmd` file and the knitted `pdf` file** to **GradeScope**. Do NOT create a zip file. For the `pdf` submission, please tag specific pages that correspond with each question in the assignment.
- You can submit multiple times, you last submission will be graded.

Acknowledgments

Please list all the help you have received for completing this homework.

INSERT_YOUR_ANSWER Used some geeksforgeeks references

Load necessary packages

```
# install the tidyverse package first if you have not done it yet.
#install.packages("tidyverse") # you can comment out this line after you have installed `tidyverse`

library(tidyverse) # for the `ggplot2` package
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

Question 1 [40pt] Visualization of the `mpg` dataset

The `mpg` dataset contains fuel economy data 1999 - 2008 for 38 popular car models. https://ggplot2.tidyverse.org/reference/mpg.html

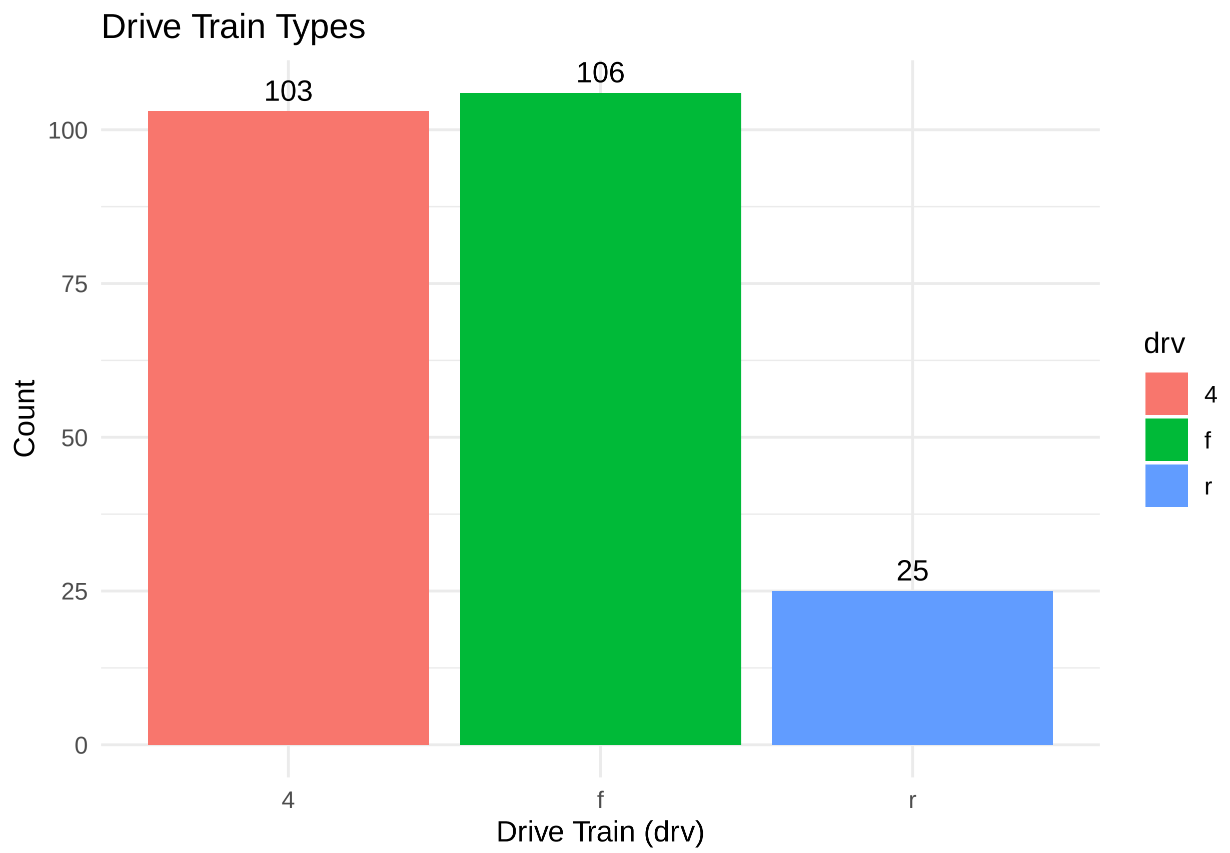
```
?mpg
## starting httpd help server ... done
glimpse(mpg) # get a glimpse of the mpg data
## Rows: 234
## Columns: 11
## $ manufacturer <chr> "audi", "audi", "audi", "audi", "audi", "audi", "audi", "~
## $ model        <chr> "a4", "a4", "a4", "a4", "a4", "a4", "a4", "a4 quattro", "~
## $ displ        <dbl> 1.8, 1.8, 2.0, 2.0, 2.8, 2.8, 3.1, 1.8, 1.8, 2.0, 2.0, 2.~
## $ year         <int> 1999, 1999, 2008, 2008, 1999, 1999, 2008, 1999, 1999, 200~
## $ cyl          <int> 4, 4, 4, 4, 6, 6, 6, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6, 8, 8, ~
## $ trans        <chr> "auto(l5)", "manual(m5)", "manual(m6)", "auto(av)", "auto~
## $ drv          <chr> "f", "f", "f", "f", "f", "f", "f", "4", "4", "4", "4", "4~
## $ cty          <int> 18, 21, 20, 21, 16, 18, 18, 18, 16, 20, 19, 15, 17, 17, 1~
## $ hwy          <int> 29, 29, 31, 30, 26, 26, 27, 26, 25, 28, 27, 25, 25, 25, 2~
## $ fl           <chr> "p", "p", "p", "p", "p", "p", "p", "p", "p", "p", "p", "p~
## $ class        <chr> "compact", "compact", "compact", "compact", "compact", "c~
```

(a) [20pt] Visualize the distribution of drive train types

(i) Draw a barplot (frequency histogram) to display the distribution of `drv`, the type of drive train. Use different colors to distinguish different drive train types. Explicitly label the number of cars for each drive train type on top of the bars.

INSERT_YOUR_ANSWER

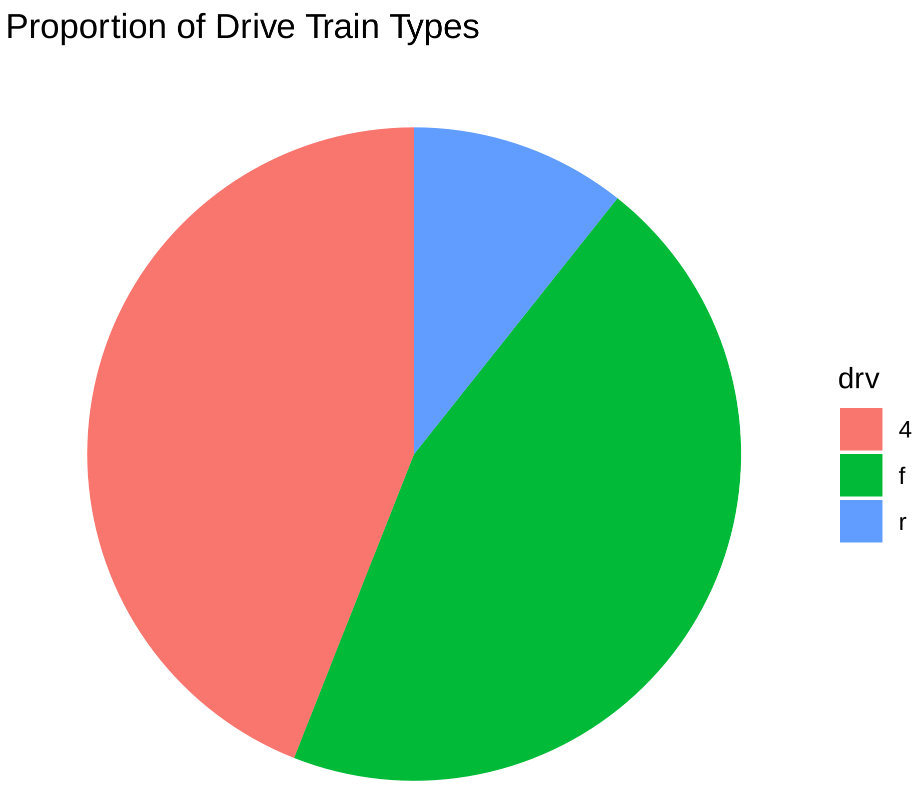
```
ggplot(mpg, aes(x = drv, fill = drv)) +
  geom_bar() +
  geom_text(stat = "count", aes(label = after_stat(count)), vjust = -0.5) +
  labs(title = "Drive Train Types", x = "Drive Train (drv)", y = "Count") +
  theme_minimal()
```



(ii) Draw a coxcomb or pie chart to display the proportions of each drive train types.

INSERT_YOUR_ANSWER

```
mpg %>%
  count(drv) %>%
  ggplot(aes(x = "", y = n, fill = drv)) +
  geom_col() +
  coord_polar(theta = "y") +
  labs(title = "Proportion of Drive Train Types") +
  theme_void()
```

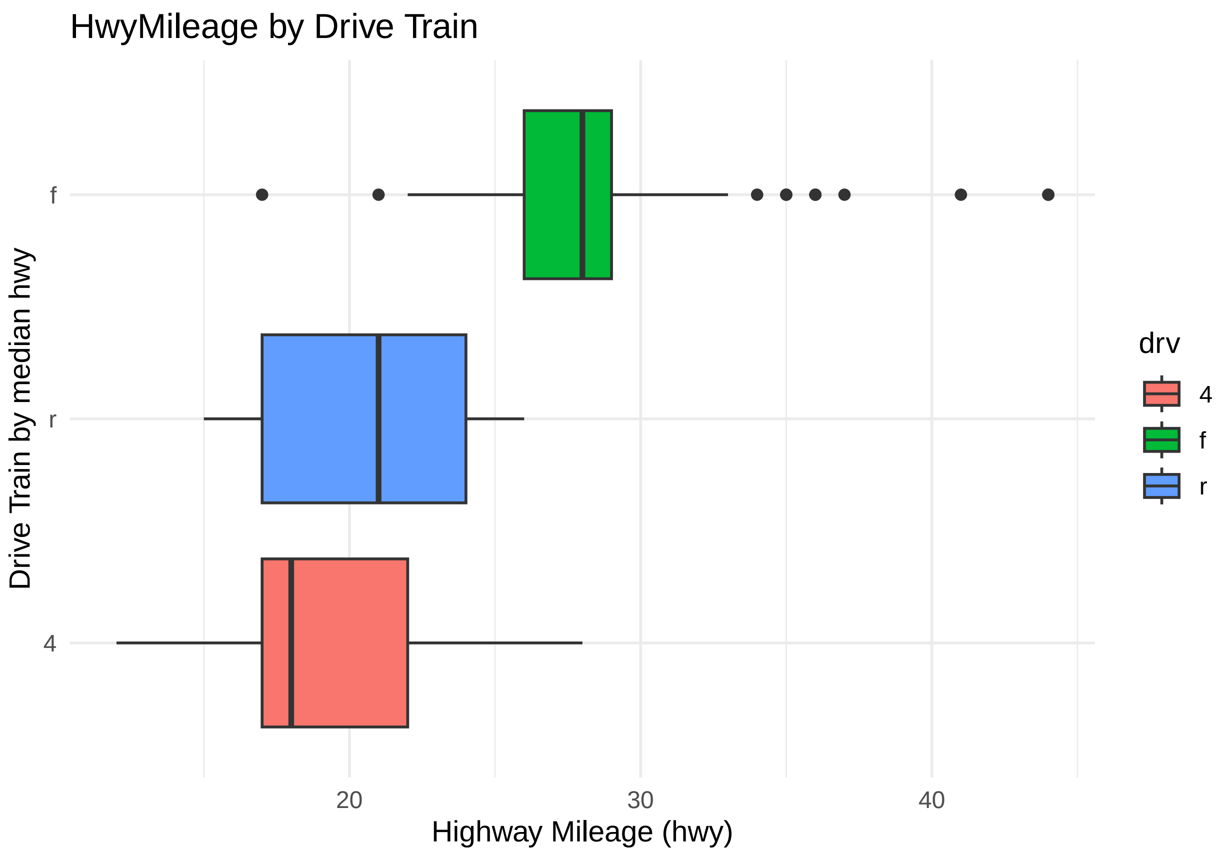


(b) [20pt] How highway mileage varies across drive train type?

Generate a horizontal boxplot to compare the distribution of highway mileage across three different drive train types. Reorder the boxes by the median mileage values.

INSERT_YOUR_ANSWER

```
ggplot(mpg, aes(x = reorder(drv, hwy, median), y = hwy, fill = drv)) +
  geom_boxplot() + coord_flip() + labs(title = "HwyMileage by Drive Train", x = "Drive Train by median
  theme_minimal()
```


HwyMileage by Drive Train
Drive Train by median hwy
f
r
4
20
30
40
Highway Mileage (hwy)
drv
4
f
r

Question 2 [60pt] Visualization the `diamonds` dataset

The `diamonds` dataset contains the prices and other attributes of almost 54,000 diamonds. https://ggplot2.tidyverse.org/reference/diamonds.html

```
?diamonds
glimpse(diamonds) # get a glimpse of the data
## Rows: 53,940
## Columns: 10
## $ carat   <dbl> 0.23, 0.21, 0.23, 0.29, 0.31, 0.24, 0.24, 0.26, 0.22, 0.23, 0.~
## $ cut     <ord> Ideal, Premium, Good, Premium, Good, Very Good, Very Good, Ver~
## $ color   <ord> E, E, E, I, J, J, I, H, E, H, J, J, F, J, E, E, I, J, J, J, I,~
## $ clarity <ord> SI2, SI1, VS1, VS2, SI2, VVS2, VVS1, SI1, VS2, VS1, SI1, VS1, ~
## $ depth   <dbl> 61.5, 59.8, 56.9, 62.4, 63.3, 62.8, 62.3, 61.9, 65.1, 59.4, 64~
## $ table   <dbl> 55, 61, 65, 58, 58, 57, 57, 55, 61, 61, 55, 56, 61, 54, 62, 58~
## $ price   <int> 326, 326, 327, 334, 335, 336, 336, 337, 337, 338, 339, 340, 34~
## $ x       <dbl> 3.95, 3.89, 4.05, 4.20, 4.34, 3.94, 3.95, 4.07, 3.87, 4.00, 4.~
## $ y       <dbl> 3.98, 3.84, 4.07, 4.23, 4.35, 3.96, 3.98, 4.11, 3.78, 4.05, 4.~
## $ z       <dbl> 2.43, 2.31, 2.31, 2.63, 2.75, 2.48, 2.47, 2.53, 2.49, 2.39, 2.~
```

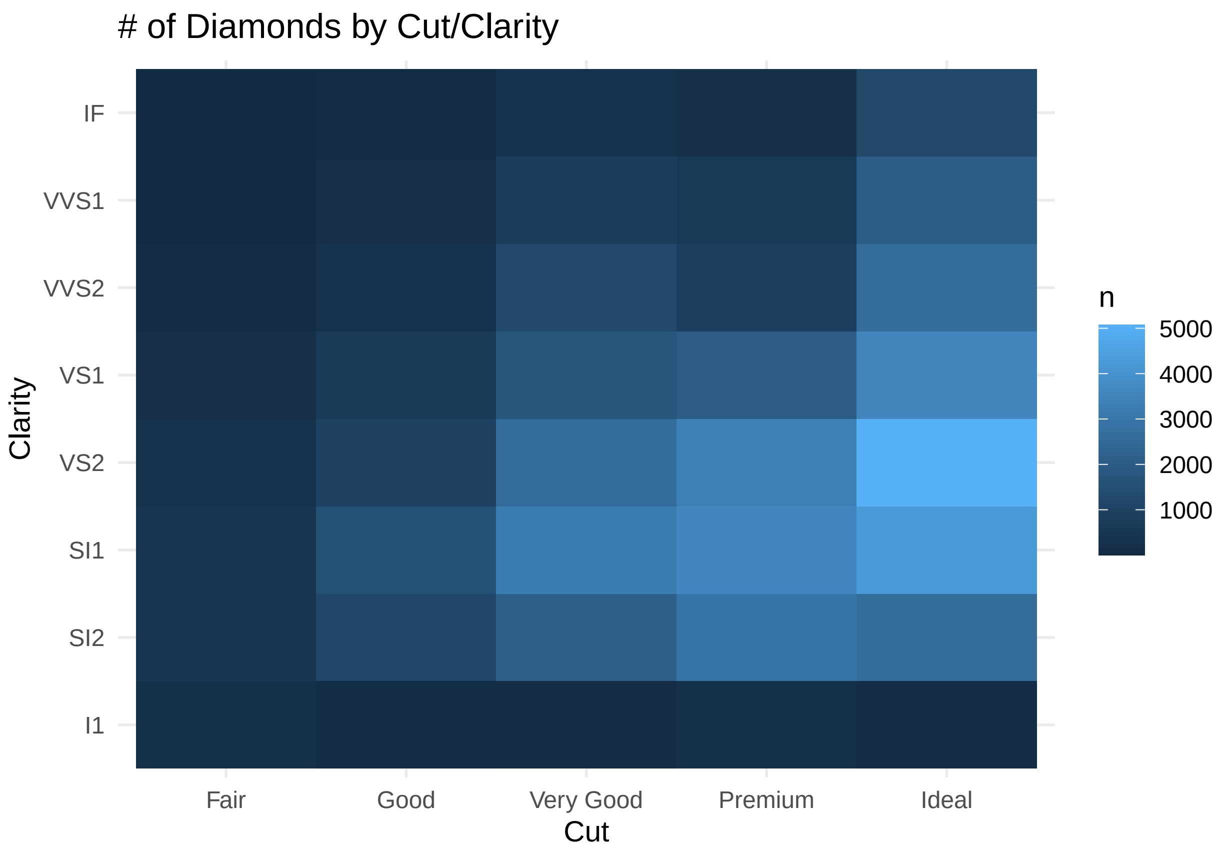
(a) [20pt] Heatmap of `cut` vs `clarity`

(i) Use the `geom_tile()` function to make a heatmap to visualize the number of diamonds in each `cut` and `clarity` combination.

INSERT_YOUR_ANSWER

```
diamonds_count <- diamonds %>%
  count(cut, clarity)

ggplot(diamonds_count, aes(x = cut, y = clarity, fill = n)) +
  geom_tile() +
  labs(title = "# of Diamonds by Cut/Clarity", x = "Cut", y = "Clarity") +
  theme_minimal()
```



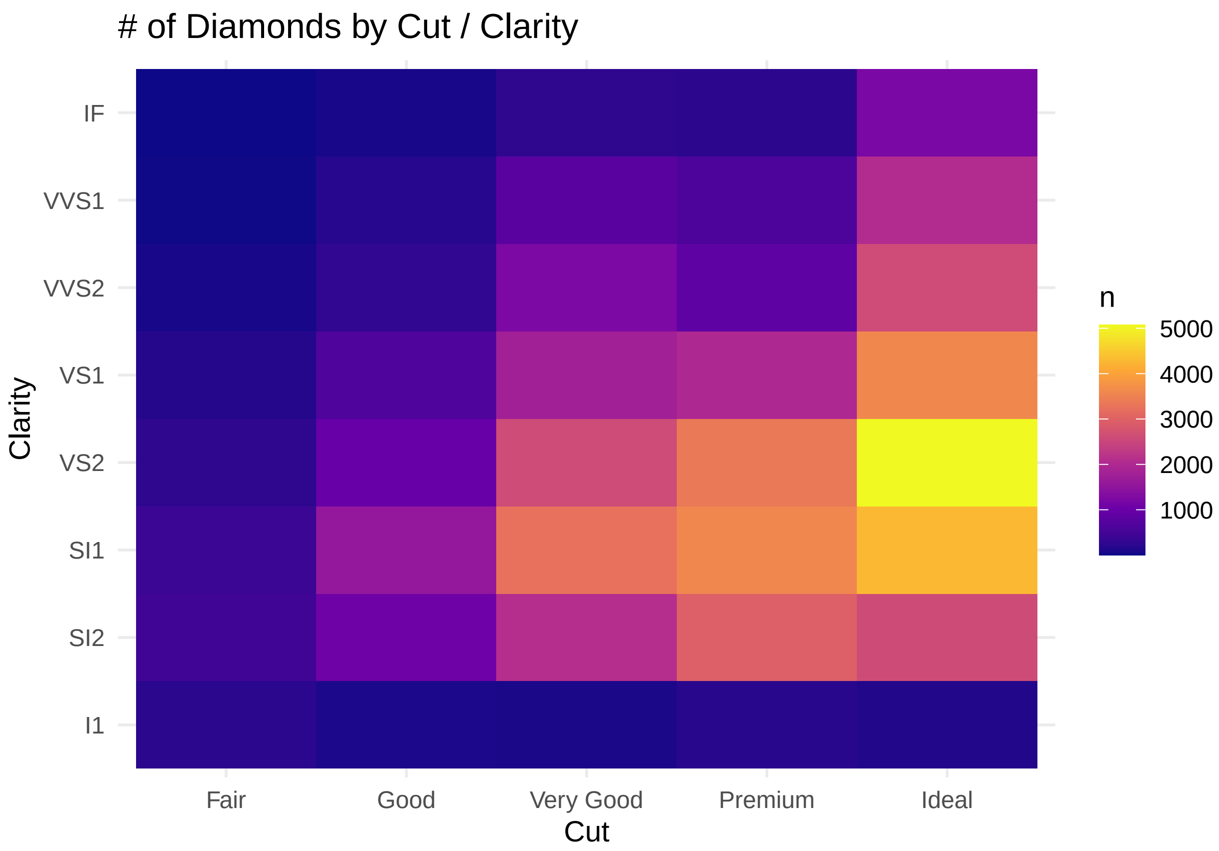
(ii) Change the color palette of your heatmap.

Hint: See these heatmap examples at the R Graph Gallery.

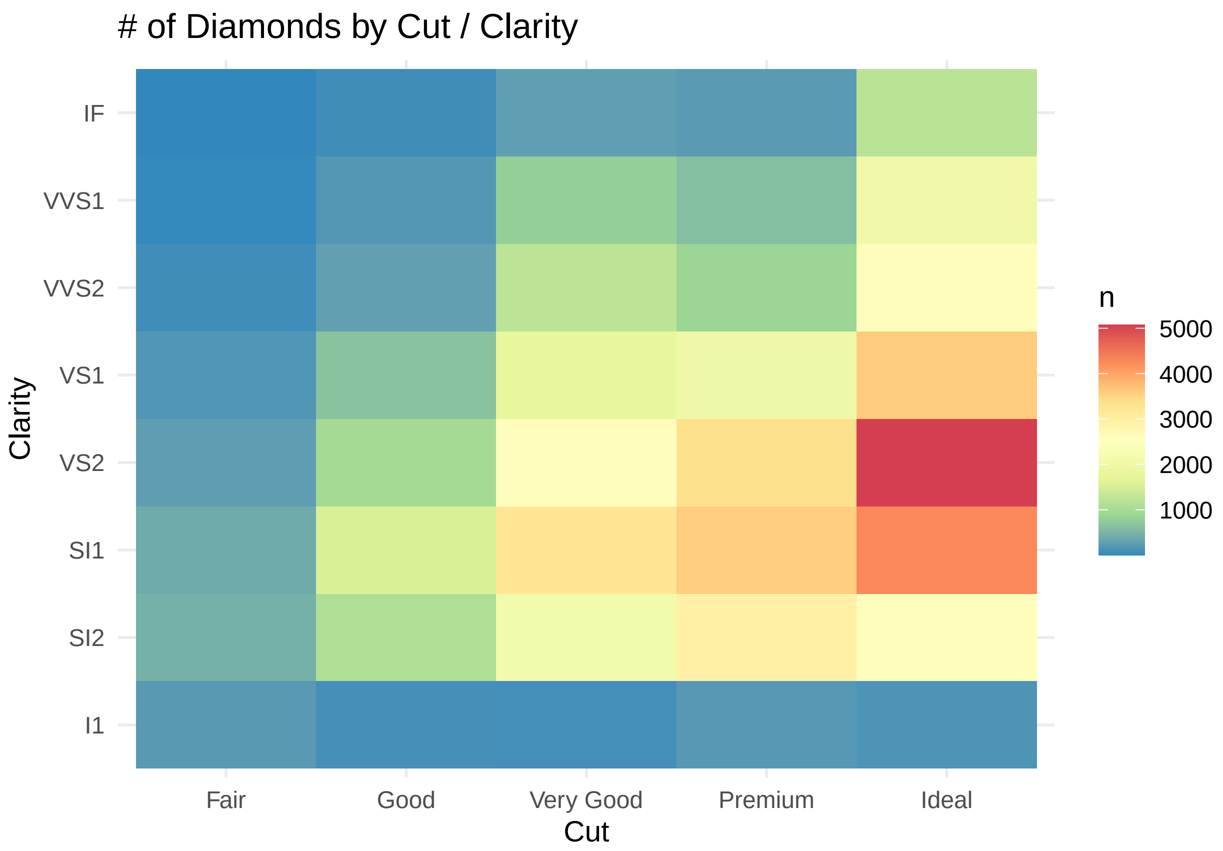
INSERT_YOUR_ANSWER

```
# I did two different color palettes

ggplot(diamonds_count, aes(x = cut, y = clarity, fill = n)) +
  geom_tile() +
  scale_fill_viridis_c(option = "C") +
  labs(title = "# of Diamonds by Cut / Clarity", x = "Cut", y = "Clarity") +
  theme_minimal()
```



```r
ggplot(diamonds_count, aes(x = cut, y = clarity, fill = n)) +
  geom_tile() +
  scale_fill_distiller(palette = "Spectral", direction = -1) +
  labs(title = "# of Diamonds by Cut / Clarity", x = "Cut", y = "Clarity") +
  theme_minimal()
```



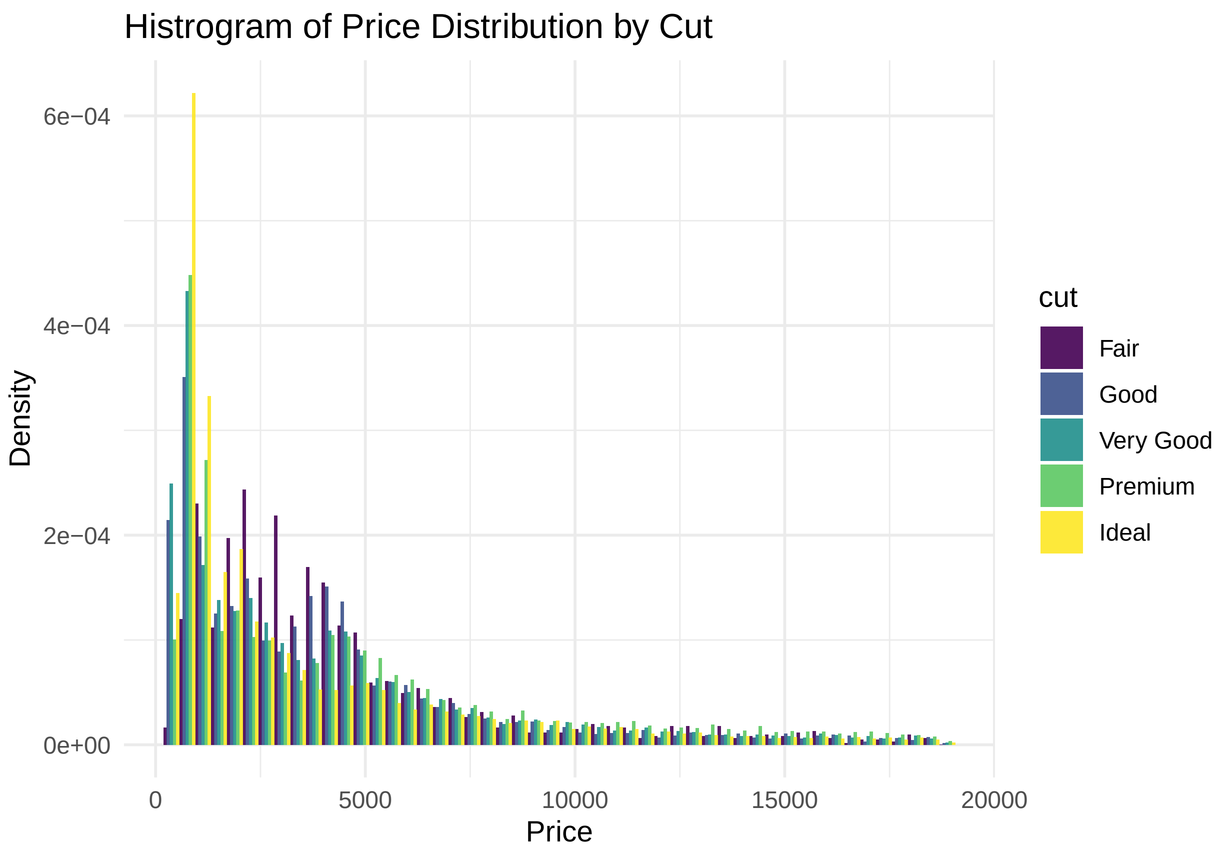
```r
# I like viridis better
```

(b) [40pt] Visualize the distribution of diamond price

(i) Use the `geom_histogram()` function to compare the distribution of `price` across different `cut`. Change the y-axis to density, and use the `dodge` position adjustment.

INSERT_YOUR_ANSWER

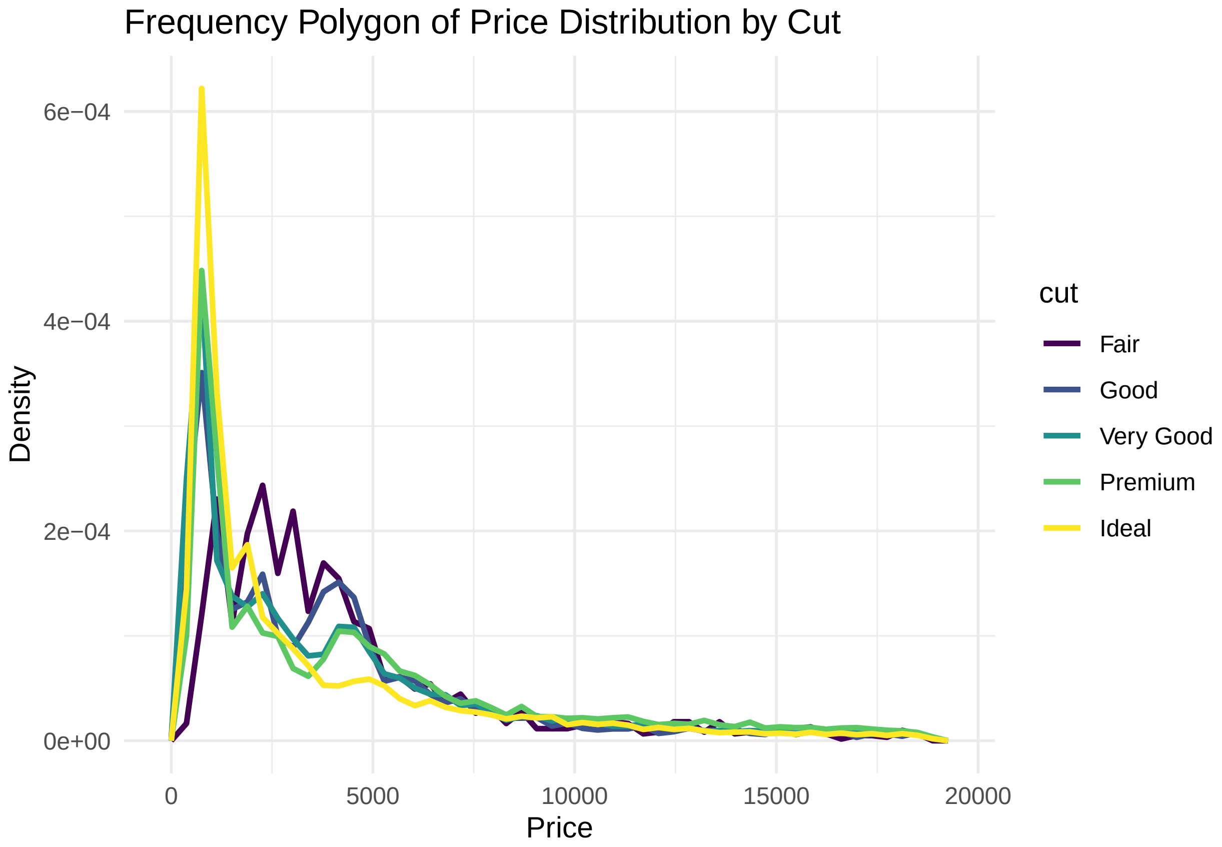
```r
ggplot(diamonds, aes(x = price, fill = cut)) +
  geom_histogram(aes(y = after_stat(density)), position = "dodge", bins = 50, alpha = 0.9) +
  labs(title = "Histrogram of Price Distribution by Cut", x = "Price", y = "Density") +
  theme_minimal()
```



(ii) Use the `geom_freqpoly()` function to compare the distribution of `price` across different `cut`. Change the y-axis to density.

INSERT_YOUR_ANSWER

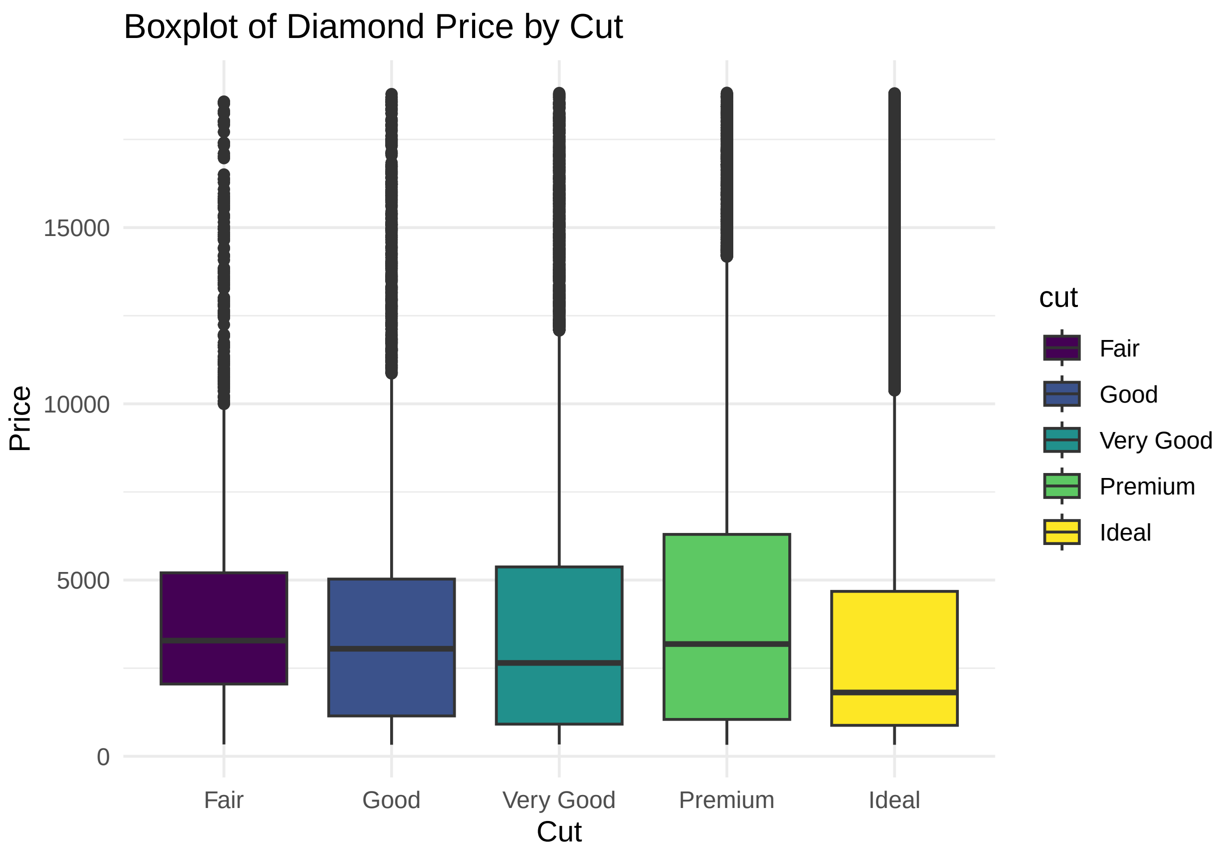
```
ggplot(diamonds, aes(x = price, color = cut)) +
  geom_freqpoly(aes(y = after_stat(density)), bins = 50, size = 1) +
  labs(title = "Frequency Polygon of Price Distribution by Cut", x = "Price", y = "Density") +
  theme_minimal()
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



(iii) Use the `geom_boxplot()` function to compare the distribution of `price` across different `cut`.

INSERT_YOUR_ANSWER

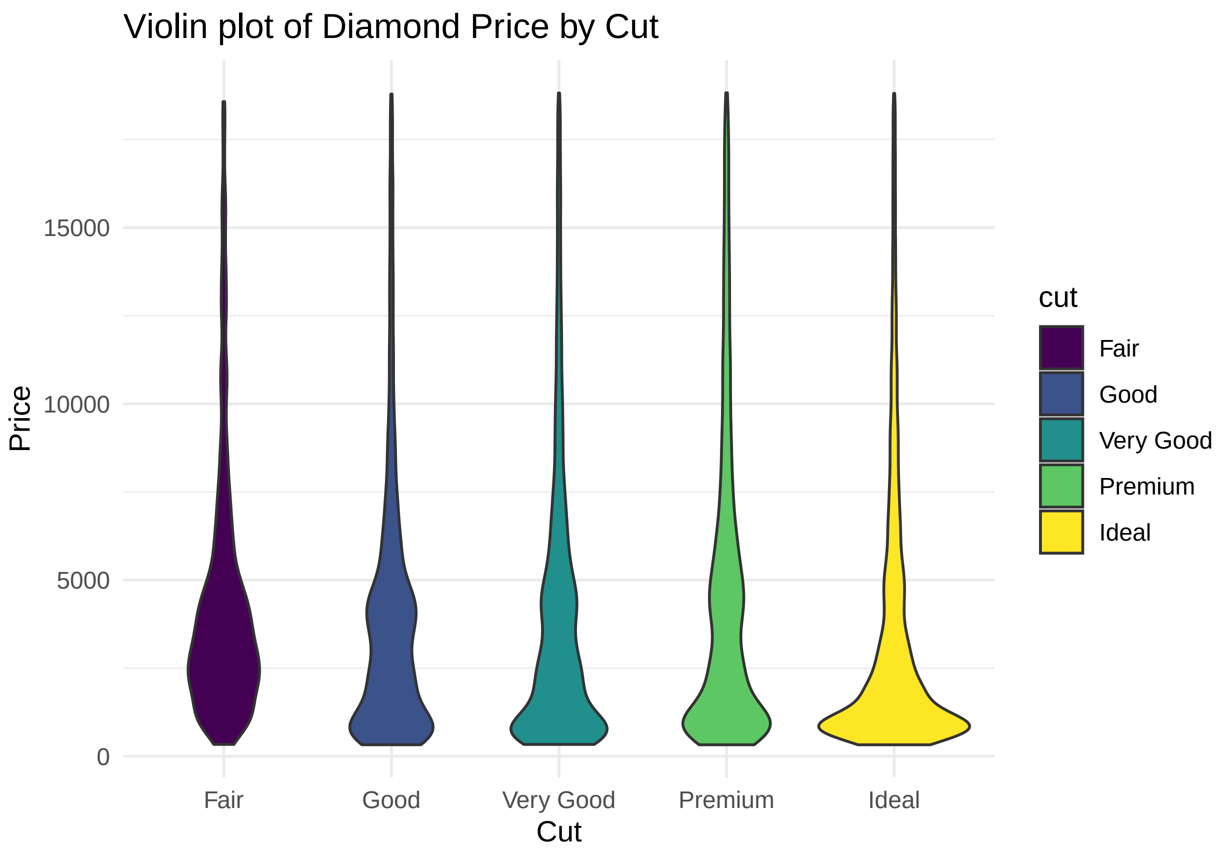
```r
ggplot(diamonds, aes(x = cut, y = price, fill = cut)) +
  geom_boxplot() +
  labs(title = "Boxplot of Diamond Price by Cut", x = "Cut", y = "Price") +
  theme_minimal()
```



(iv) Use the `geom_violin()` function to compare the distribution of `price` across different `cut`.

INSERT_YOUR_ANSWER

```
ggplot(diamonds, aes(x = cut, y = price, fill = cut)) +
  geom_violin() +
  labs(title = "Violin plot of Diamond Price by Cut", x = "Cut", y = "Price") +
  theme_minimal()
```



(v) What observations can you make from the above plots? Which visualization function is your favorite? Explain your choice.

INSERT__YOUR__ANSWER Diamonds with ideal and premium cuts tend to have lower median prices than fair cuts, and fair cut diamonds seem to have a wider range of variety with expensive outliers. Also, I can see clustering / skewness in the histogram/frequency plots, but not in the boxplots and violin plots. My favorite plot is probably the histogram plot or frequency plot just because they visualize the clustering, and I also like how they look.