**Language: R (tidyverse, ggplot2, Jittered** scatter plot**)**

**Scope:** Analysis of **diamonds** data frame that comes with **tidyverse** package which contains 53,930 observations on 10 features related to diamonds.

1. **Jittered** scatter plot of the **price** of a diamond as a function of diamond **weight**, broken into five sub-panels across two rows, where each sub-panel represents a different cut quality. Depict the points as the color gray, with a transparency alpha value of 0.1. White background.

Code snippet:

#Install and read libraries to use

install.packages("tidyverse")

library(tidyverse)

library(ggplot2)

#load dataset and see type

data(diamonds)

class(diamonds)

#plot requested visual

ggplot(data = diamonds) +

geom\_jitter(mapping = aes(x = carat, y = price),

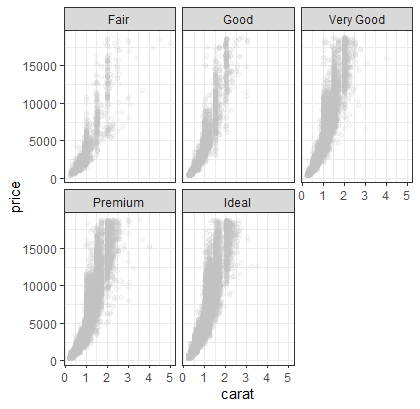
color = 'grey', alpha = 0.1) +

theme\_bw() +

facet\_wrap(~ cut, nrow = 2)

#save the plot

ggsave("plot1.png")



1. **Rug** **plot** to the figure from 1.

A rug plot is part of tidy verse and allows us to see a compact plot. Based on the ggplot website, the definition is a “Compact visualization designed to supplement a 2d display with the two 1d marginal distributions. Rug plots display individual cases so are best used with smaller datasets.” (Wickham et al.)

To develop a rug plot, you can use the geom\_rug() from ggplot2 library in R.

Code:

install.packages("tidyverse")

library(tidyverse)

library(ggplot2)

ggplot(data = diamonds) +

geom\_jitter(mapping = aes(x = carat, y = price),

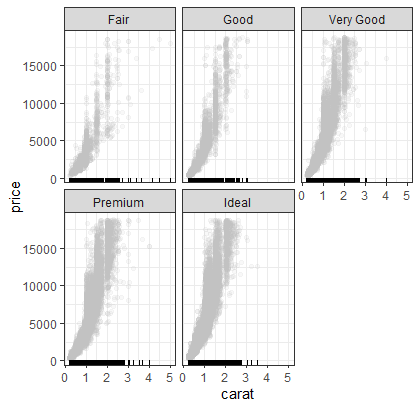
color = 'grey', alpha = 0.1) +

theme\_bw() +

facet\_wrap(~ cut, nrow = 2) +

geom\_rug(mapping = aes(x = carat))

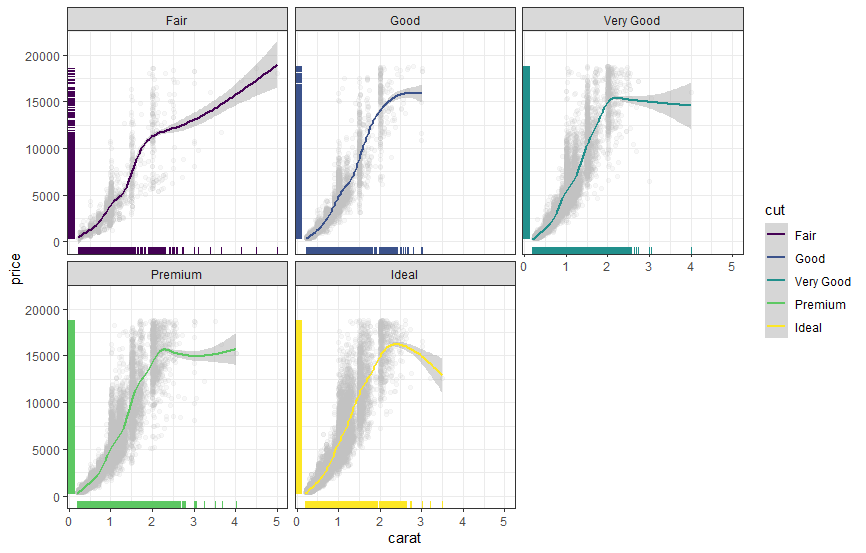
ggsave("plot2.png")



**Reference**

Wickham, H., Chang, W., Henry, L., Lin Pedersen, T., Takahashi, K., Wilke, C., Woo, K., Yutani, H., & Dunnington, D. (n.d.). *Rug plots in the margins - geom\_rug*. - geom\_rug • ggplot2. Retrieved March 20, 2023, from https://ggplot2.tidyverse.org/reference/geom\_rug.html

**3.** Plot where the bands around the fitted colored lines are 95% confidence intervals.



Code snippet:

ggplot(data = diamonds) +

geom\_jitter(mapping = aes(x = carat, y = price),

color = 'grey', alpha = 0.1) +

theme\_bw() +

geom\_rug(mapping = aes(x = carat, color = cut)) +

facet\_wrap(~ cut, nrow = 2) +

geom\_smooth(mapping = aes(x = carat, y = price,

color = cut))

**Analysis:**

* **Carat versus Price Relationship:** From the plots of carat versus price, when seeing different cuts, we can infer there’s a direct relationship between diamond weight and price. When the diamond weight increases, the price increases too, being the price increase higher for the better diamond’s cut.
* **Why are the confidence intervals much narrower for diamonds weighing less than three carats than for diamonds weighing greater than three carats?:** Confidence levels are narrower in diamonds weighing less than three carats because in the dataset there are more rows with these weights of diamonds; which results in a more confident plot (narrower confidence level) than in diamonds with more weight, giving there are not so many data points in the dataset, so to plot the 95% confidence level the visualization is wider to ensure will capture all variation of data (but considering fewer data points given the current dataset)

**4.** Box plot removing feature of the inclusion of outliers (only the box and whiskers are plotted. Made by assigning the argument **outlier.shape** to **NA**)

Code snippet:

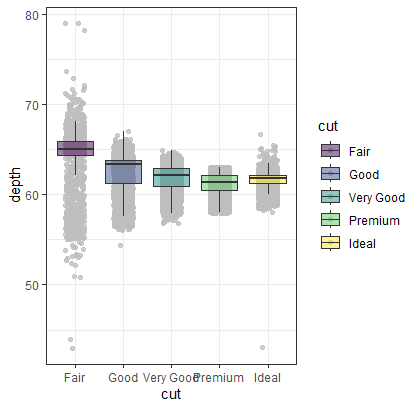
ggplot(data = diamonds,

mapping = aes(x = cut, y = depth, fill = cut)) +

geom\_jitter(width = 0.2, color = "gray", alpha = 0.75) +

geom\_boxplot(alpha = 0.5, outlier.shape = NA) +

theme\_bw()



Plot Notes:

Plot update: below code was added to the scatter plot with a width of datapoints of 0.2 in color gray and a little transparency to represent the datapoints the whisker plot is showing. This helps the visualization in giving more insights about how data is spread based on different categories in the x-axis related to the cut of the diamond.

geom\_jitter(width = 0.2, color = "gray", alpha = 0.75)

**5.** Violin plot with the same level of transparency as the box plots.

Code snippet:

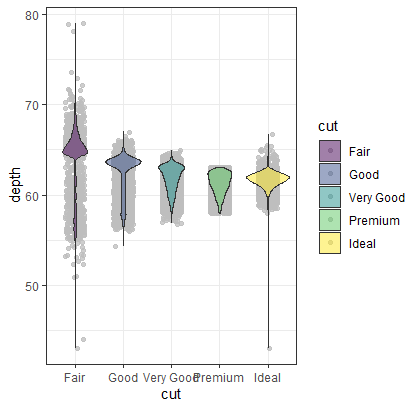
ggplot(data = diamonds,

mapping = aes(x = cut, y = depth, fill = cut)) +

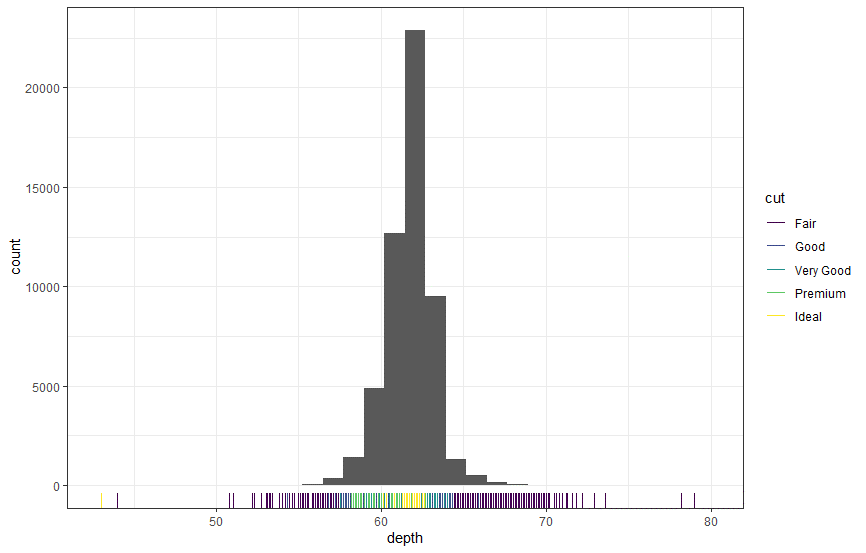
geom\_jitter(width = 0.2, color = "gray", alpha = 0.75) +

geom\_violin(alpha = 0.5, outlier.shape = NA) +

theme\_bw()



**6. Histogram**



Code snippet:

ggplot(data = diamonds) +

geom\_histogram(mapping = aes(x = depth)) +

geom\_rug(mapping = aes(x = depth, color = cut))