30535 Skills Problem Set 3

Weilu Jiang

04/22/2022

Front matter
This submission is my work alone and complies with the 30535 integrity policy.

Add your initials to indicate your agreement: **W.J.**

Late coins used this pset: 0. Late coins left: 3.

Clear Global Environment

Working Directory and Loading Packages

```
# Setting the Working Directory
setwd("~/Desktop/Spring Quarter 2022/DPPP R/Week 4/skills-problem-set-3-weiluj")
# Loading Packages
library(tidyverse)
library(dplyr)
library(tidyr)
library(ggplot2)
library(statar)
library(binsreg)
```

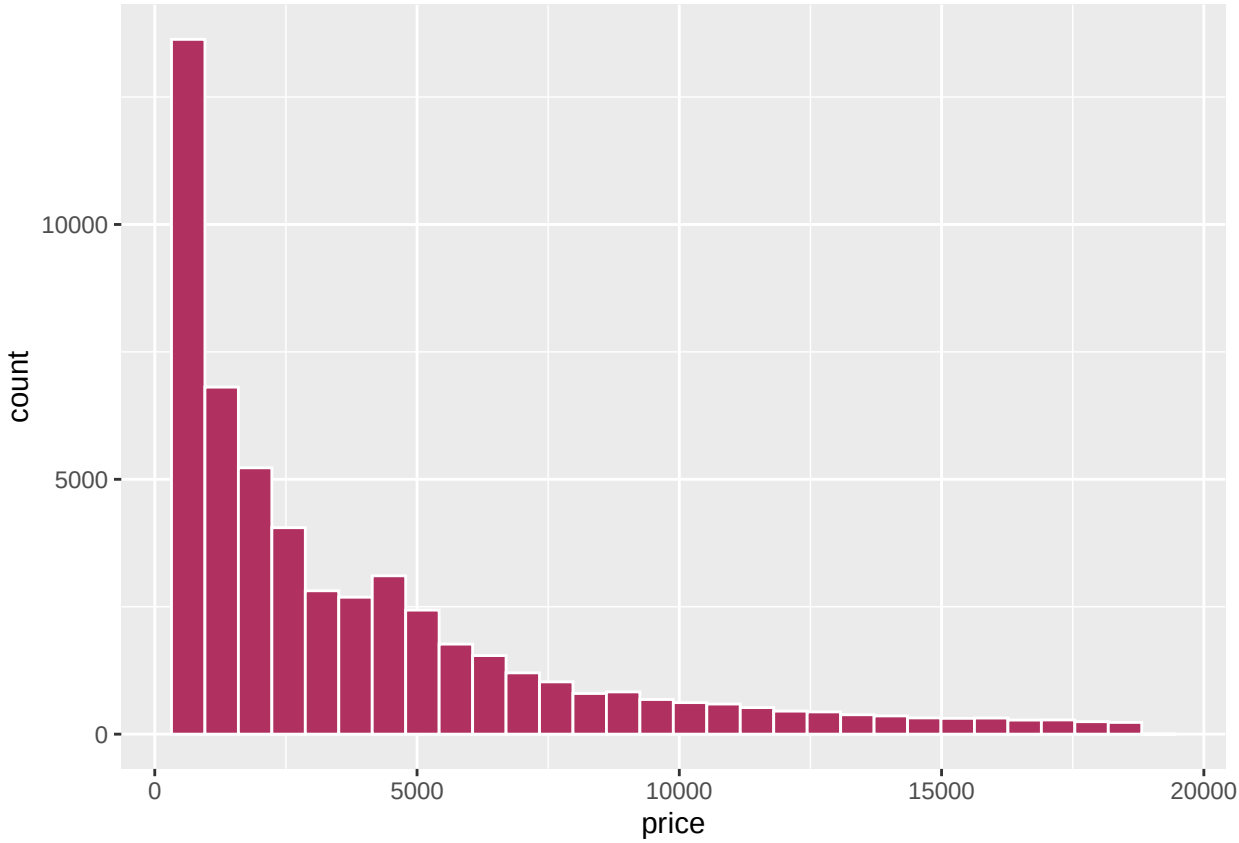
2 EDA : Exploring variation

2.1

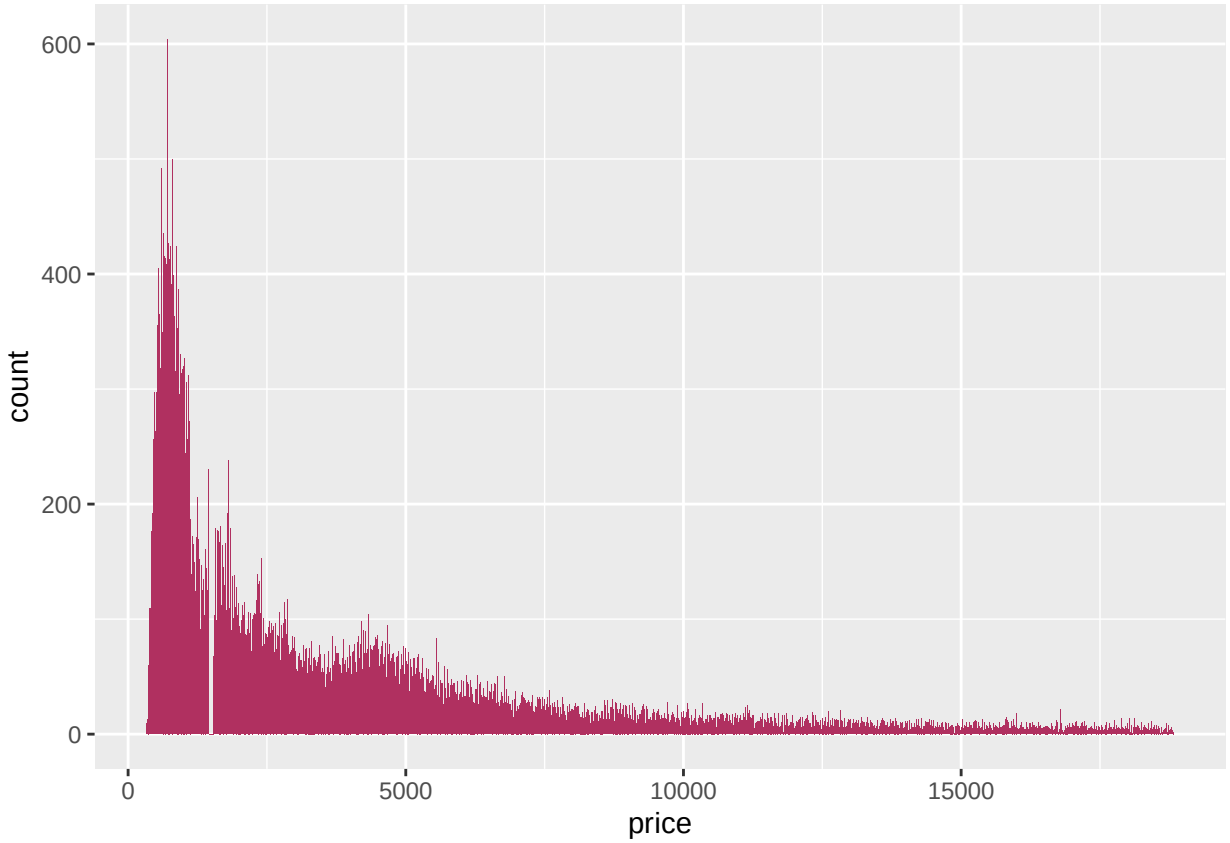
```
# Read dataset
diamonds <- diamonds
# a. distribution of price
summary(diamonds$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     326     950    2401    3933    5324   18823
```

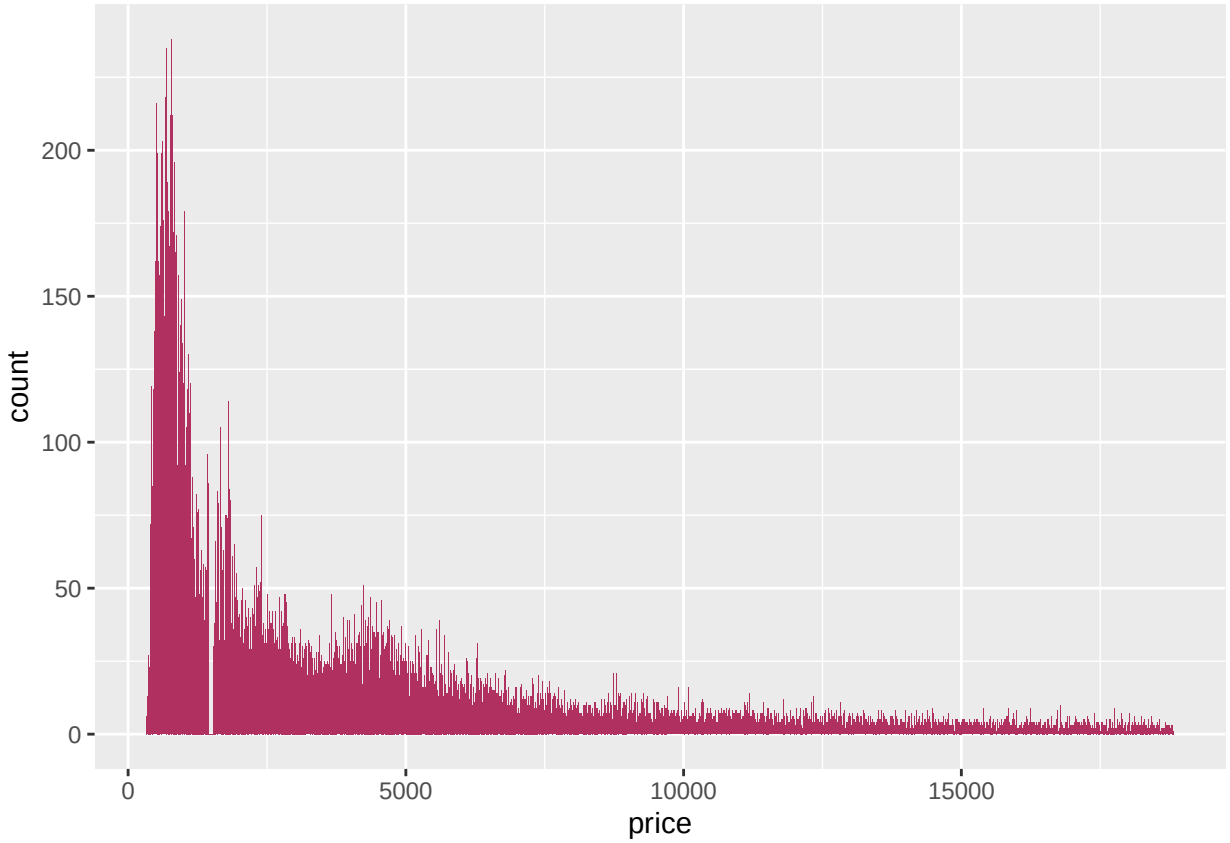
```
# b. plot distribution of price to find possible unusual observation
ggplot(data = diamonds) +
  geom_histogram(mapping = aes(x = price),
                 fill = "maroon",
                 color = "white")
```



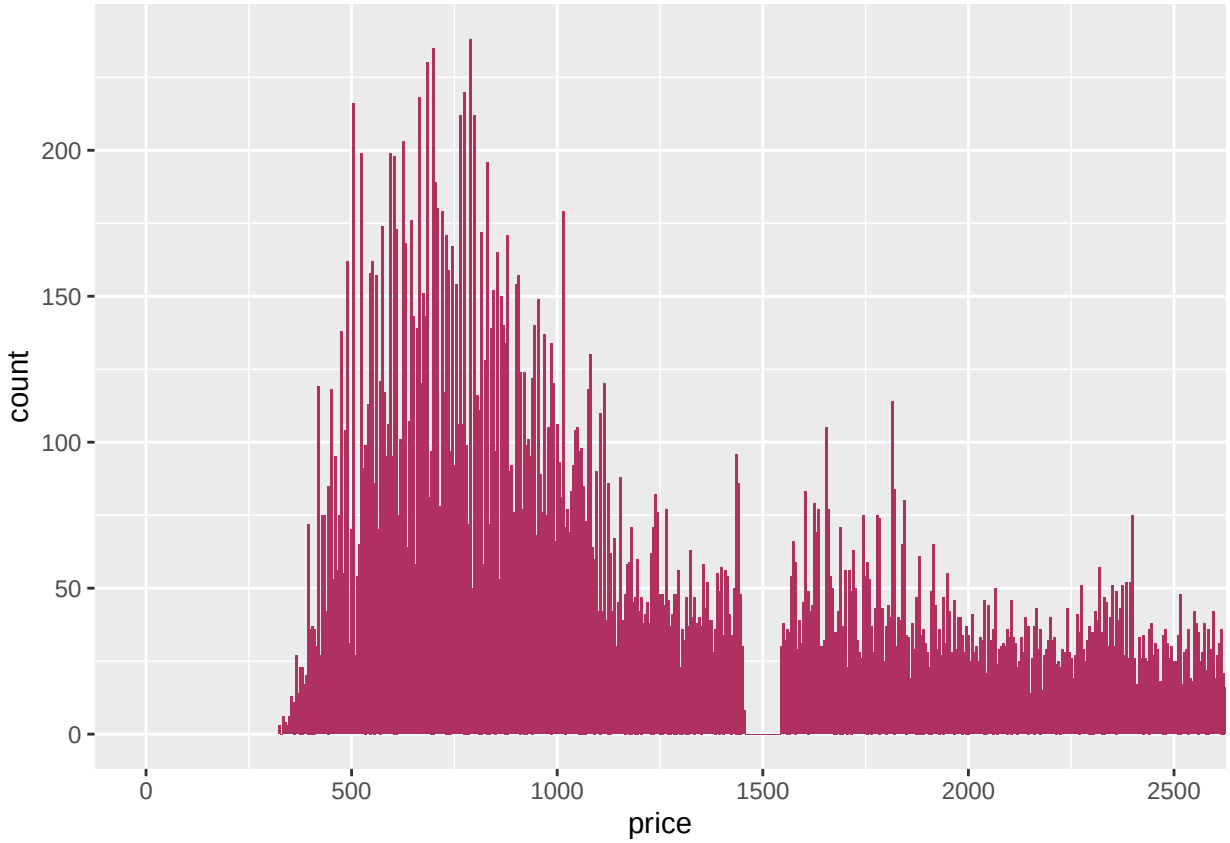

```
ggplot(data = diamonds) +
  geom_histogram(mapping = aes(x = price),
                 fill = "maroon",
                 binwidth = 15)
```




```
ggplot(data = diamonds) +
  geom_histogram(mapping = aes(x = price),
                 fill = "maroon",
                 binwidth = 5)
```




```
# Set x value limit
ggplot(diamonds) +
  geom_histogram(aes(x = price),
                 binwidth = 5,
                 fill = "maroon") +
  coord_cartesian(xlim = c(0, 2500))
```



Interpretation

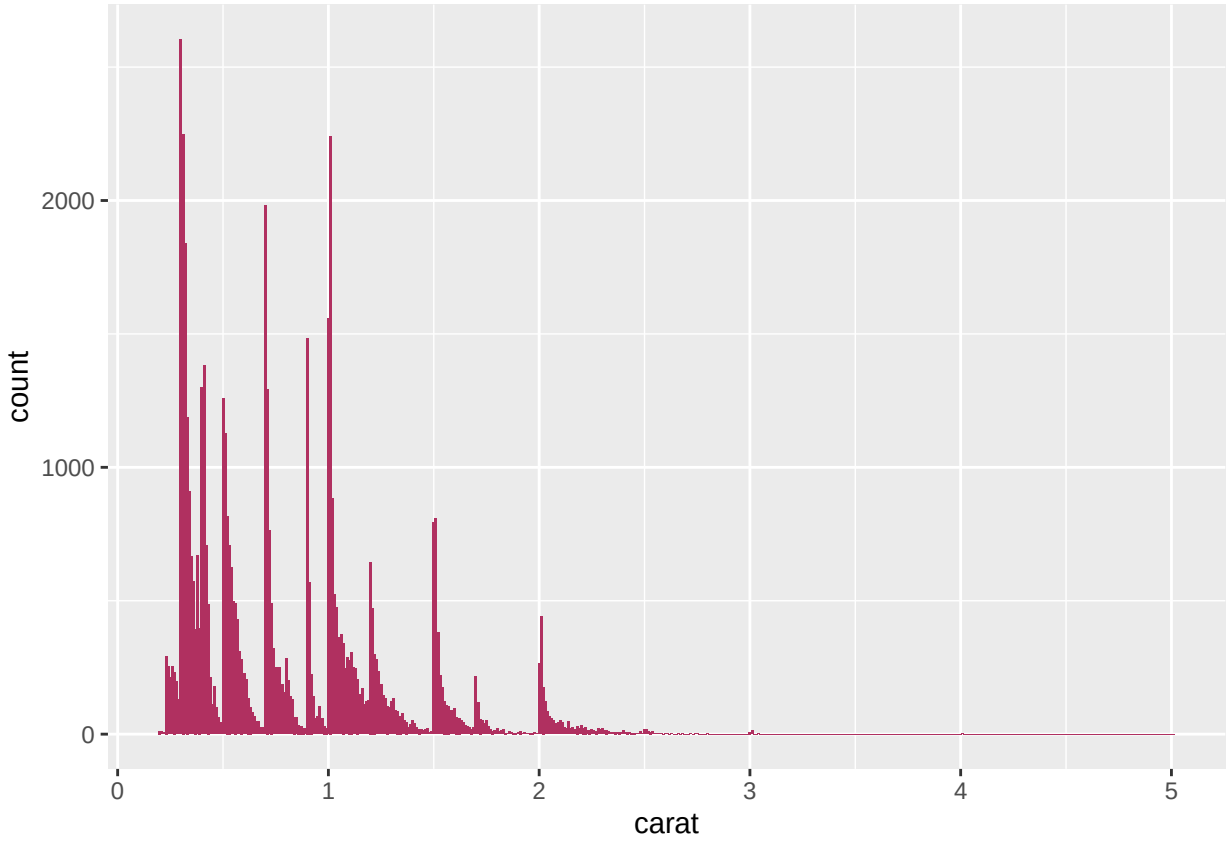
a. The overall distribution of diamond's price in the dataset is right-skewed, which means most data falls to the right side of the graph. With that, the mean of diamonds' prices are higher than the median. It's not surprising because diamonds are supposed to be expensive so most data will converged to the right side (higher prices). And some diamonds can be extremely expensive since people prefer high quality or heavier carat, which makes the data has a long right tail and bumped during the 75th quantile and the maximum value.

b. Histogram plots with different binwidth or variable range limit have uncovered unusual facts of the data:

- With the default histgram plot, we only observe the number of data is stably decreasing for most cases. However, when we change the binwidth, we surprisingly find out that there is a complete gap for prices around 1500 US Dollars. It's clearer when we set the x limit to 2,500.
- We also notice that there are great variation of cases for prices between 500 and 1000, which is unobservable with the default plot.

2.2

```
ggplot(diamonds) +
  geom_histogram(aes(carat),
                 fill = "maroon",
                 binwidth = 0.01)
```



Interpretation

The overall distribution of diamond's carat is a bit different from our expectation. The pattern is also right-skewed but with less skewness. Also, it seems there will be a peak number of cases every 0.5 carats and also a significant decrease after the peak. It does not perfectly fit the expectation that diamonds with heavier carats will have cheaper prices. We may need to look at other columns in the dataset to find other variables that affect prices.

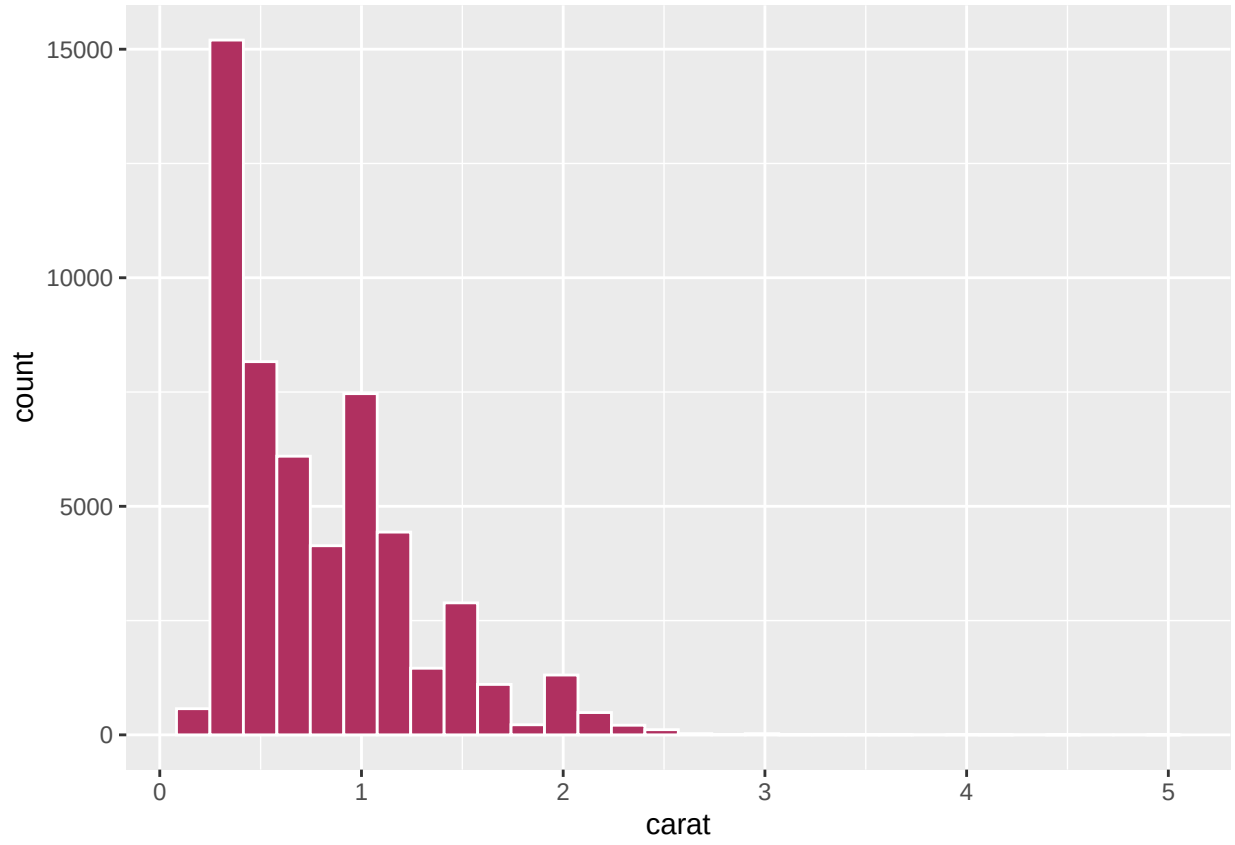
```
diamonds %>%
  filter(carat >= 0.99, carat <= 1) %>%
  count(carat)
# Support the argument
diamonds %>%
  filter(carat >= 0.9 & carat <= 1) %>%
  group_by(carat) %>%
  summarise(mean_price = mean(price),
            median_price = median(price),
            maxprice = max(price),
            min_price = min(price),
            count = n()) %>%
  arrange(desc(carat))
diamonds %>%
  filter(carat == 0.99) %>%
  ggplot() +
  geom_histogram(aes(price),
                 fill = "maroon",
                 color = "white")
```

```
diamonds %>%
  filter(carat == 1) %>%
  ggplot() +
  geom_histogram(aes(price),
                 fill = "maroon",
                 color = "white")
```

Interpretation There are 23 diamonds with 0.99 carat, while the number of diamonds with 1 carat is 1558. The prices of 1 carat diamonds are centered at higher values. The reason could be that people usually prefer to have integer numbers when purchasing goods, which causing 1 carat diamonds to be much more popular

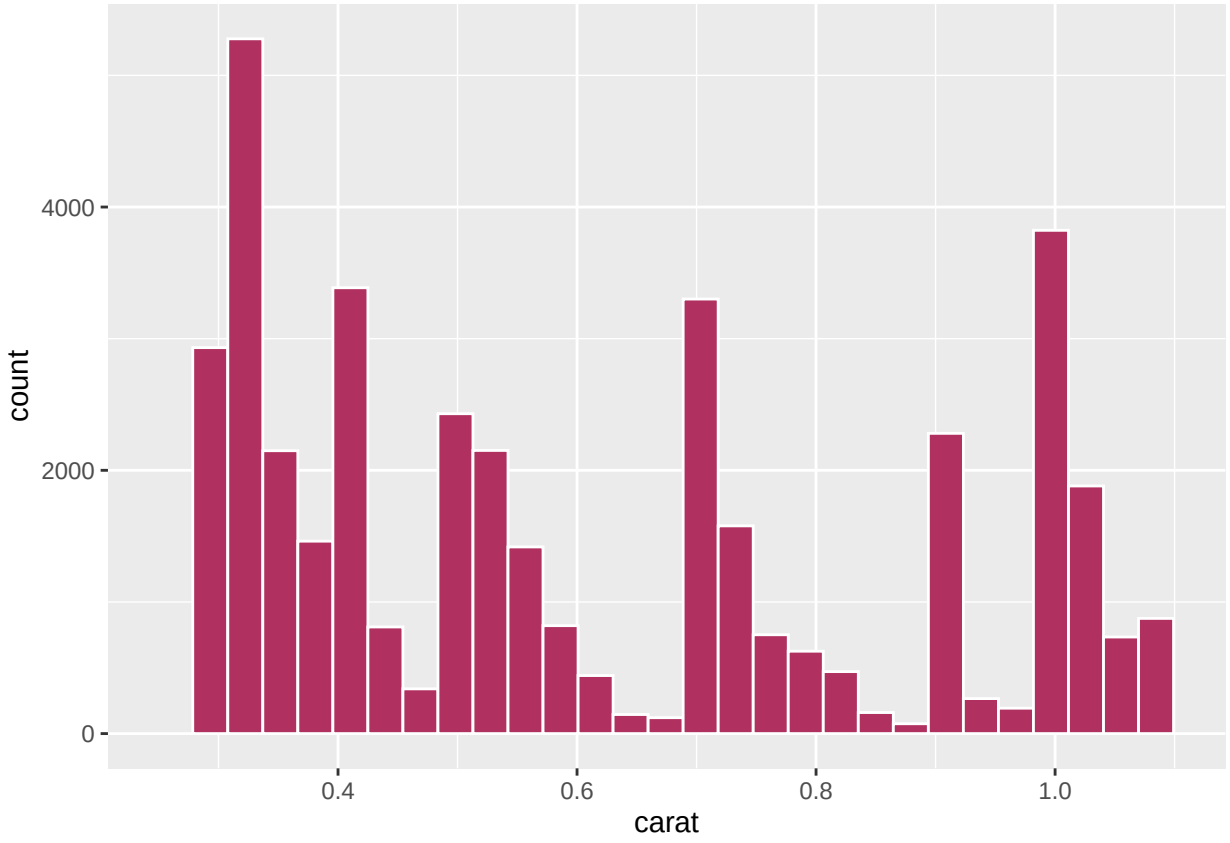
2.3

```
# No zoom in
ggplot(diamonds) +
  geom_histogram(aes(x = carat),
                 fill = "maroon",
                 color = "white")
```

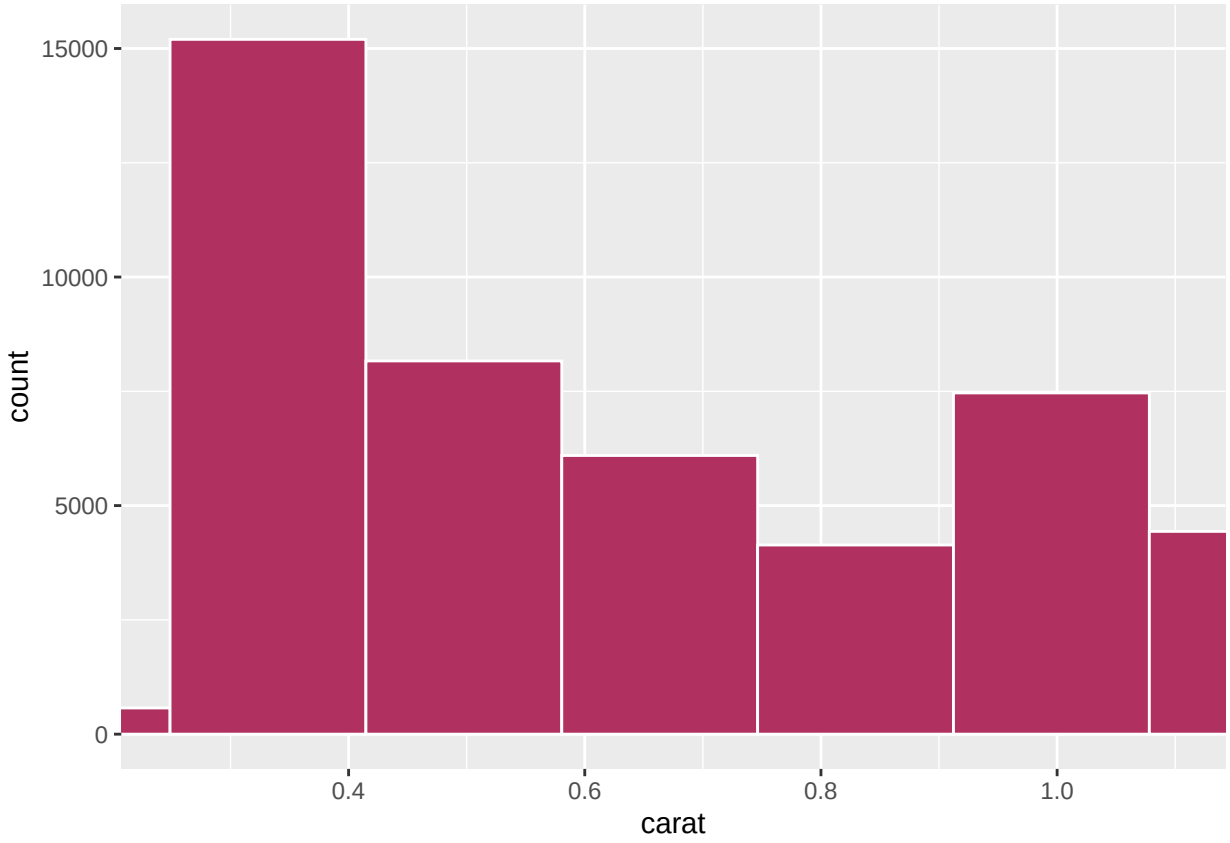



```
# Zooming in with xlim()
ggplot(diamonds) +
  geom_histogram(aes(x = carat),
                 fill = "maroon",
```

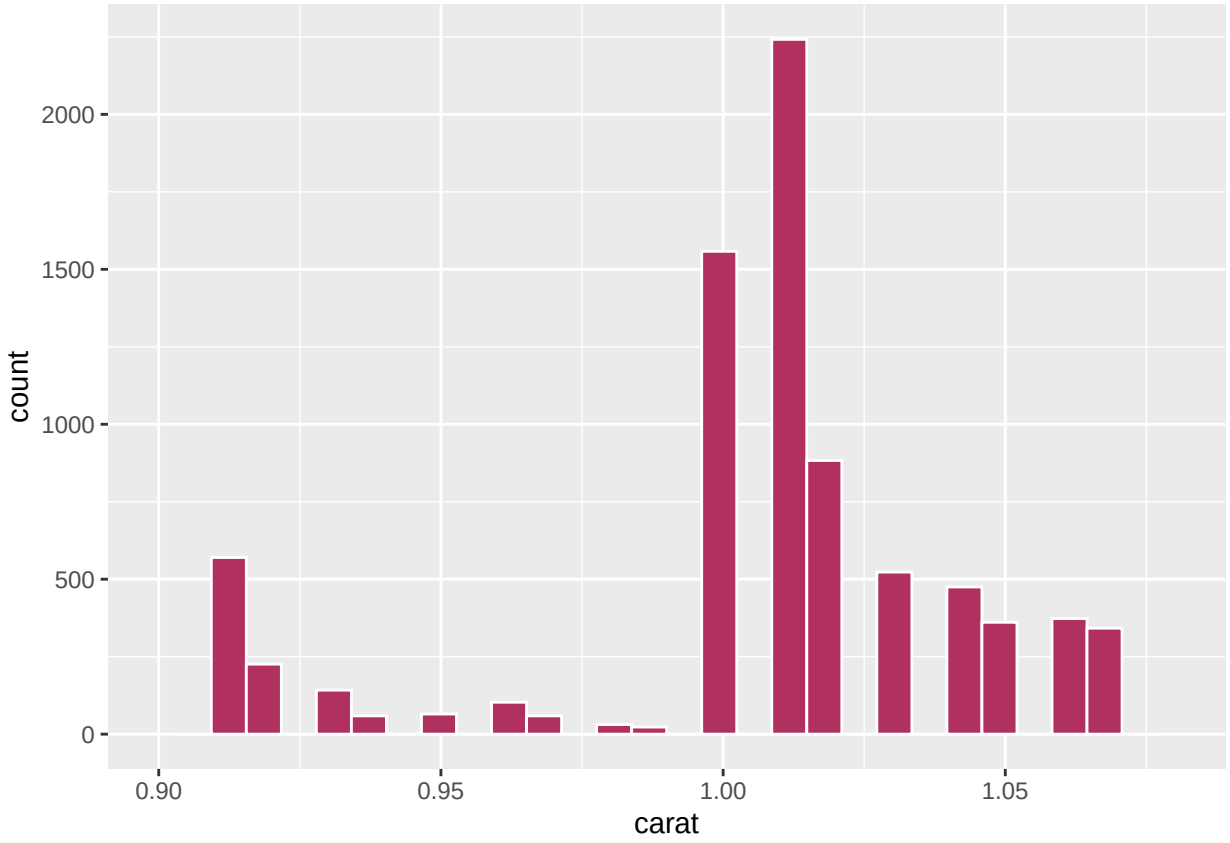
```
                 color = "white") +
  xlim(c(0.25, 1.1))
```



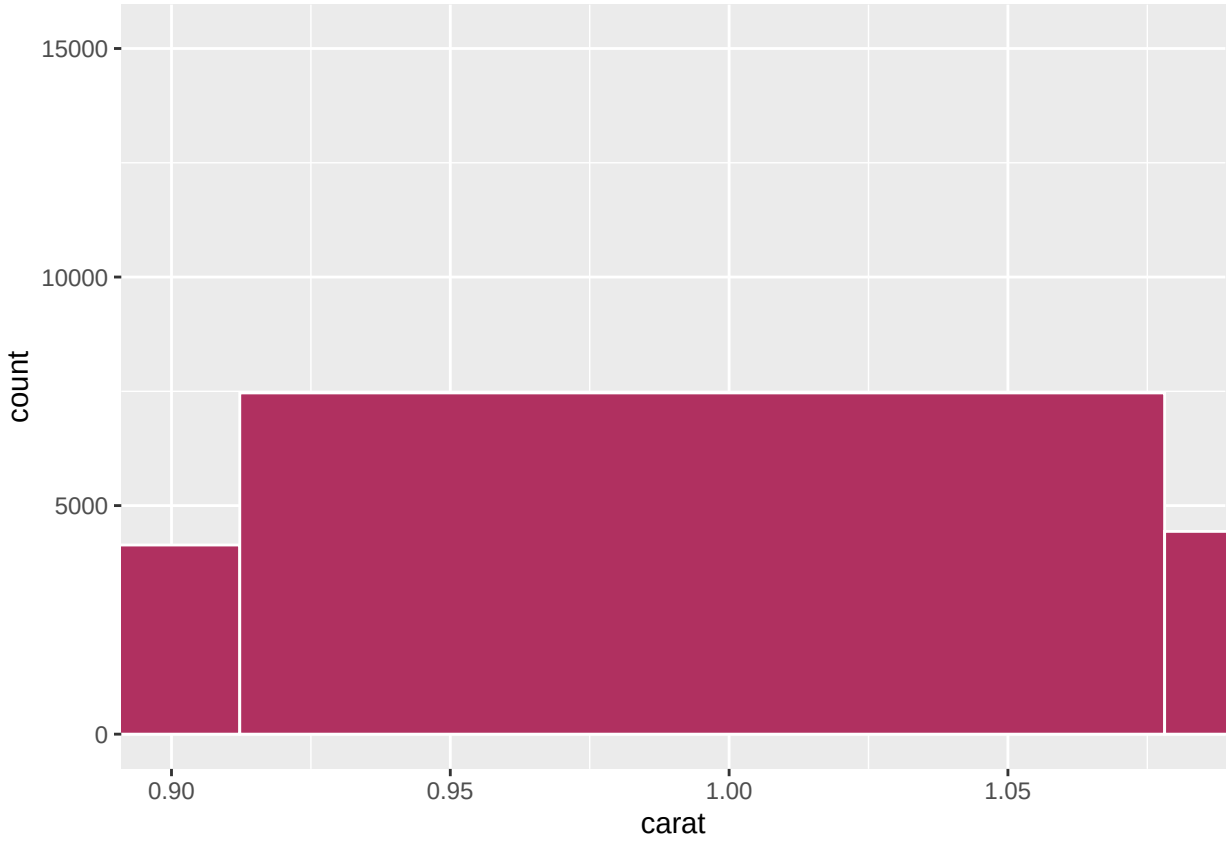

```
# Zooming in with coord_catesian()
ggplot(diamonds) +
  geom_histogram(aes(x = carat),
                 fill = "maroon",
                 color = "white") +
  coord_cartesian(xlim = c(0.25,1.1))
```




```
# Zooming in further
ggplot(diamonds) +
  geom_histogram(aes(x = carat),
                 fill = "maroon",
                 color = "white") +
  xlim(c(0.9, 1.08))
```




```
# Zooming in with coord_catesian()
ggplot(diamonds) +
  geom_histogram(aes(x = carat),
                 fill = "maroon",
                 color = "white") +
  coord_cartesian(xlim = c(0.9, 1.08))
```



Interpretation

- They both zoom on the plot by setting limits to axis values, but xlim() or ylim() will remove values outside of the limits and there is nothing special with the plot other than that.
- However, coord_cartesian will directly cut off the plot, which means it does not change the data but will 'screenshot' part of it. The default bin width will be 30 and looks like trapezoid.

3 EDA : Navigating NAs

3.1

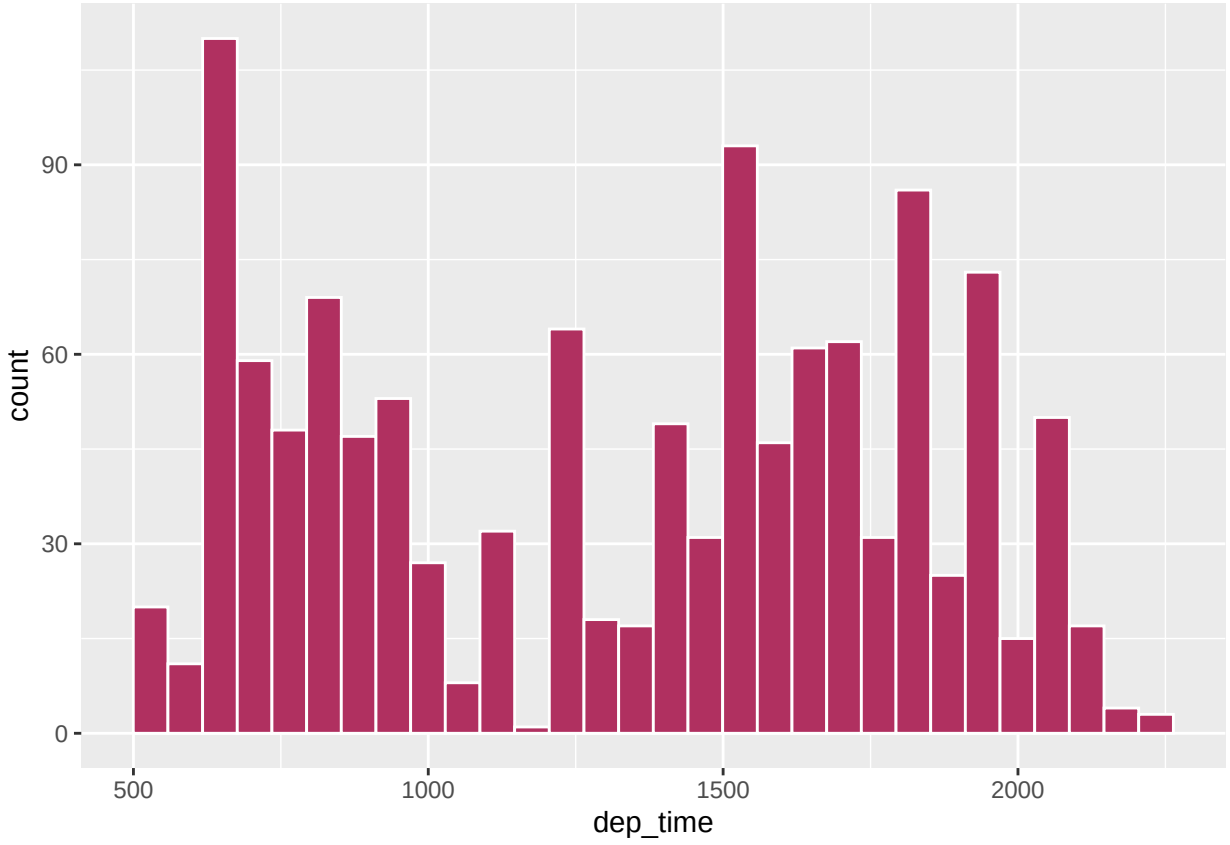
```
flights <- nycflights13::flights
flights <-
  flights %>%
  filter(month == 1,
         dest == "ORD") %>%
  mutate(carrier_na = ifelse(carrier == "MQ",
                             print(NA),
                             carrier),
           )
```

```
## [1] NA
```

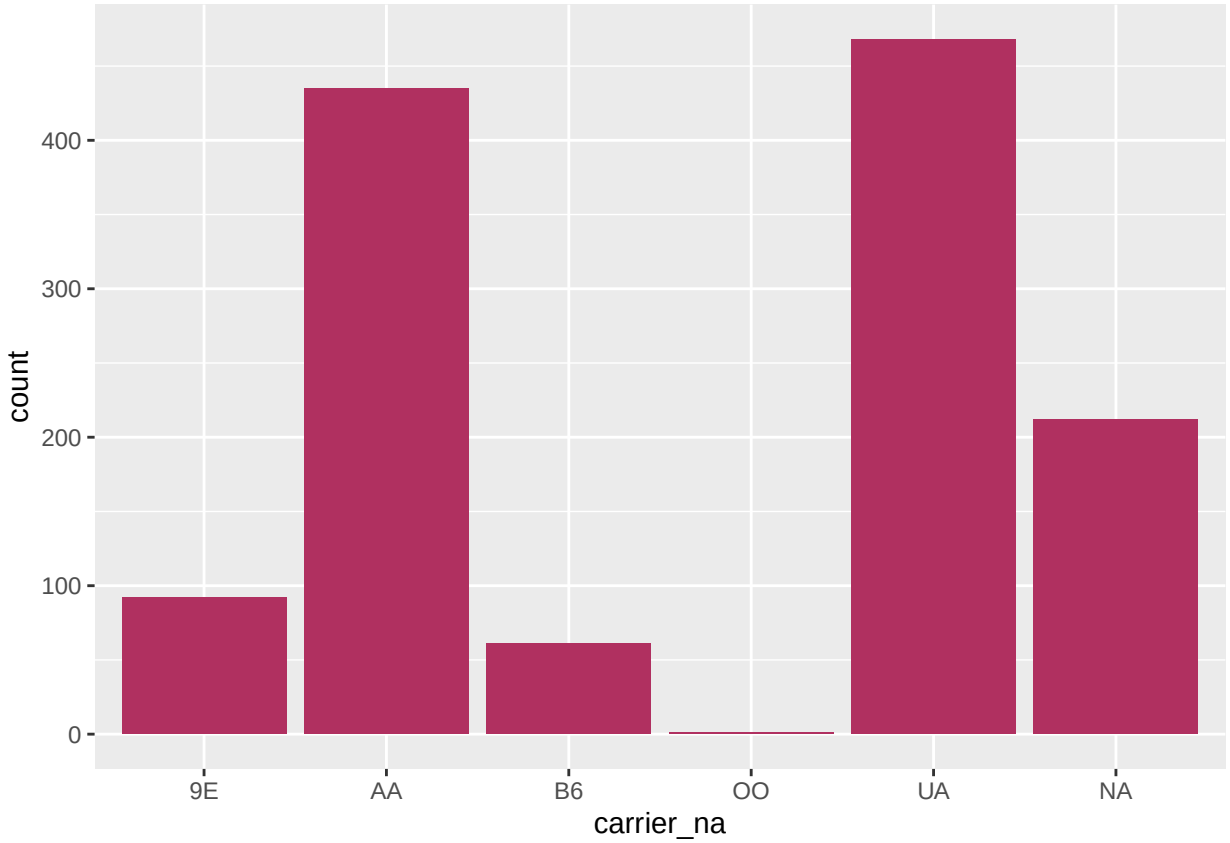
```
summary(flights$carrier_na)
```

```
##    Length     Class      Mode
##      1269 character character
```

```
# Plot
ggplot(flights) +
  geom_histogram(aes(x = dep_time),
                 fill = "maroon",
                 color = "white")
```




```
ggplot(flights) +
  geom_bar(aes(x = carrier_na),
                 fill = "maroon")
```



Interpretation

- Histogram: NA values will be removed and ignored when plotting. R will tell us this with a warning message
- Bar chart: NA values will be regarded as a single group
- The reason is histogram is for continuous variables and could not plot NAs, while bar chart could be used for categorical vairalbes so it could include NAs as a group

3.2

```
# Example
sample <- c(seq(1, 5, 1), NA)
# Mean
mean(sample)
```

```
## [1] NA
```

```
mean(sample, na.rm = TRUE)
```

```
## [1] 3
```

```r
# Sum
sum(sample)
```

```
## [1] NA
```

```r
sum(sample, na.rm = TRUE)
```

```
## [1] 15
```

Interpretation $na.rm = TRUE$ will remove NA values from the data/vector before calculating the mean and sum. If we are calculating the mean or sum of a specific column in a dataset, using $na.rm()$ will only remove NA values of this column instead of removing all NA values which occur in the dataset. We'll only get NA as a result if not removing NA values.

4 Diamonds

4.1

```r
# Define the most important variable for predicting diamonds' prices
diamonds_reg <-
  diamonds %>%
  mutate(color = as.numeric(fct_rev(color)),
         clarity = as.numeric(clarity),
         cut = as.numeric(cut)
         )

reg_diamonds_price <- lm(price ~ . , data = diamonds_reg)
summary(reg_diamonds_price)
```

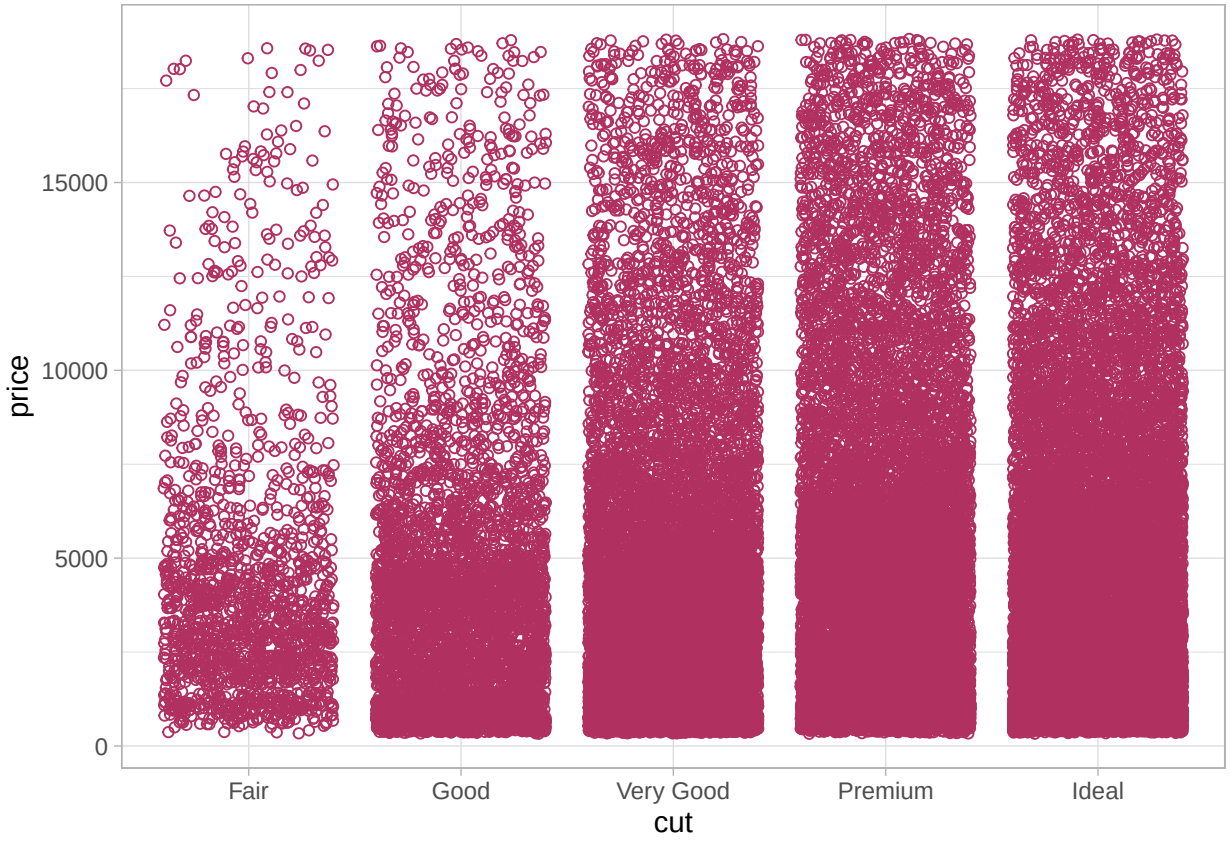
```
##
## Call:
## lm(formula = price ~ ., data = diamonds_reg)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23560.7   -629.7   -127.9    494.9   9903.1
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2781.147    428.809   6.486 8.91e-11 ***
## carat       10743.908     51.837 207.263  < 2e-16 ***
## cut           120.750      5.715  21.130  < 2e-16 ***
## color         322.696      3.259  99.003  < 2e-16 ***
## clarity       501.856      3.523 142.450  < 2e-16 ***
## depth         -79.793      4.794 -16.644  < 2e-16 ***
## table         -26.760      2.948  -9.078  < 2e-16 ***
## x            -877.631     35.226 -24.914  < 2e-16 ***
## y              43.735     20.751   2.108   0.0351 *
## z             -29.335     36.017  -0.814   0.4154
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1217 on 53930 degrees of freedom
## Multiple R-squared:  0.907,  Adjusted R-squared:  0.907
## F-statistic: 5.845e+04 on 9 and 53930 DF,  p-value: < 2.2e-16
```

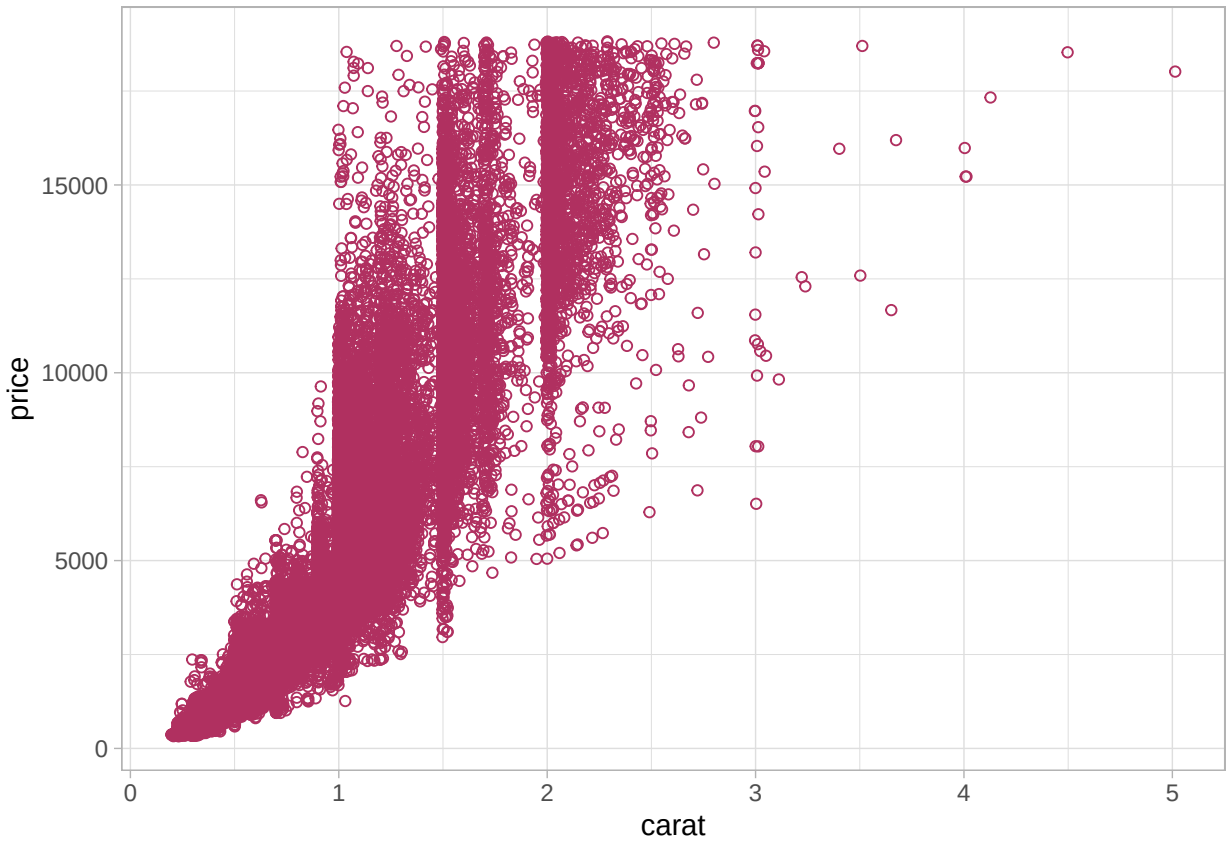
```
quantile(diamonds_reg$cut)
```

```
##   0%  25%  50%  75% 100%
##    1    3    4    5    5
```

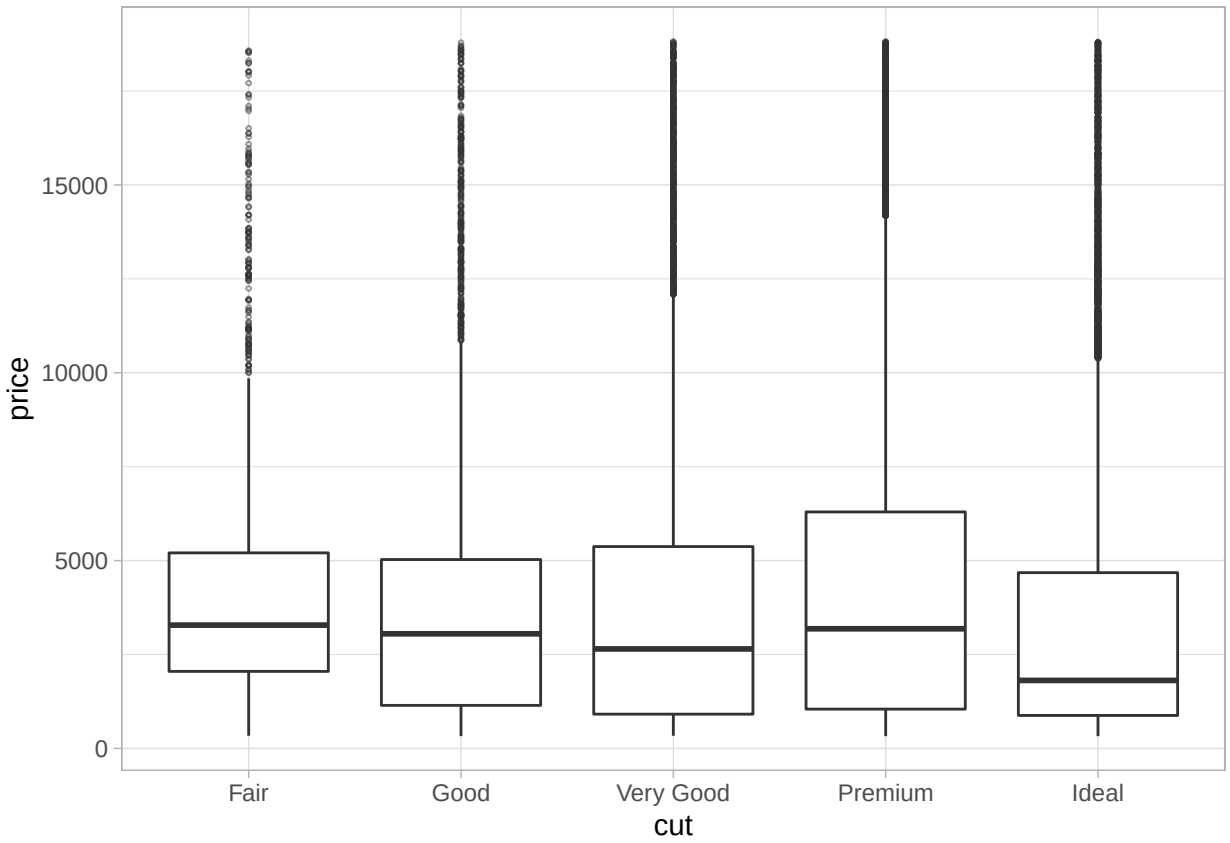
```
# Scatter plot
ggplot(diamonds,
       aes(x = cut, y = price)) +
  geom_jitter(shape = 1,
              color = "maroon") +
  theme_light()
```



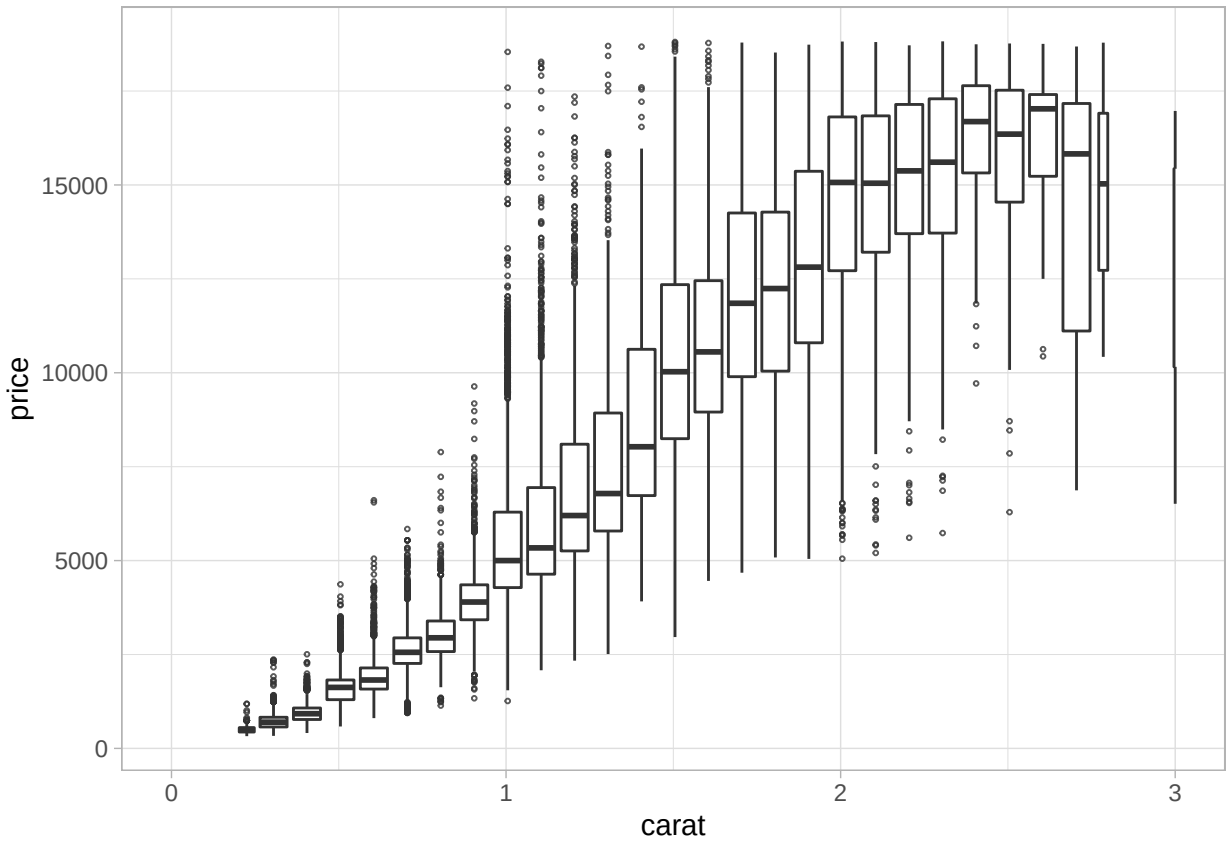

```
ggplot(diamonds,
       aes(x = carat, y = price)) +
  geom_jitter(shape = 1,
              color = "maroon") +
  theme_light()
```



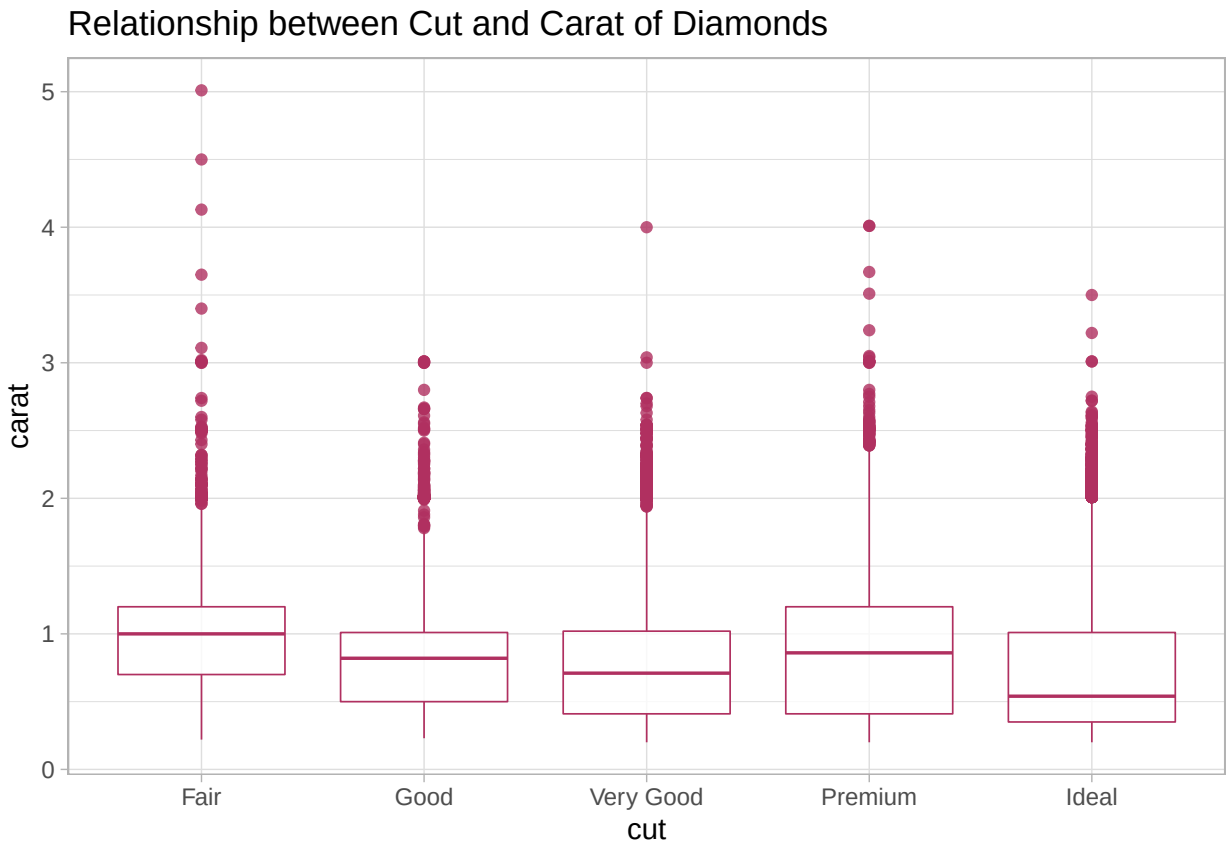

```
# Box plot
ggplot(diamonds, aes(x = cut, y = price)) +
  geom_boxplot(outlier.shape = 1,
               outlier.size = 0.5,
               outlier.alpha = 0.5) +
  theme_light()
```




```
ggplot(diamonds, aes(x = carat, y = price)) +
  geom_boxplot(aes(group = cut_width(carat, 0.1)),
               orientation = "x",
               outlier.shape = 1,
               outlier.size = 0.5,
               outlier.alpha = 0.8) +
  xlim(0,3) +
  theme_light()
```




```
# Relationship between carat and cut
ggplot(diamonds,
       aes(x = cut, y = carat)) +
  geom_boxplot(size = 0.3,
               alpha = 0.8,
               color = "maroon") +
  labs(title = "Relationship between Cut and Carat of Diamonds") +
  theme_light()
```

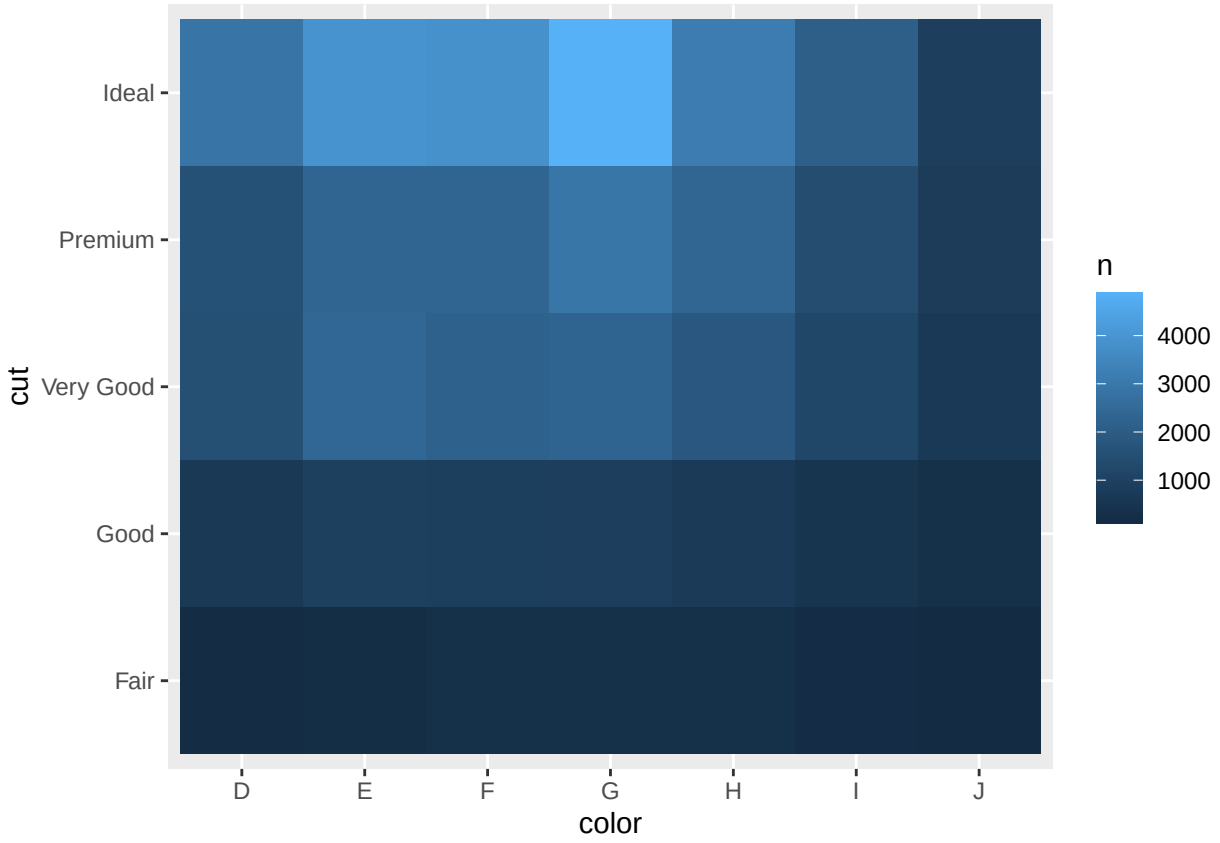


Interpretation

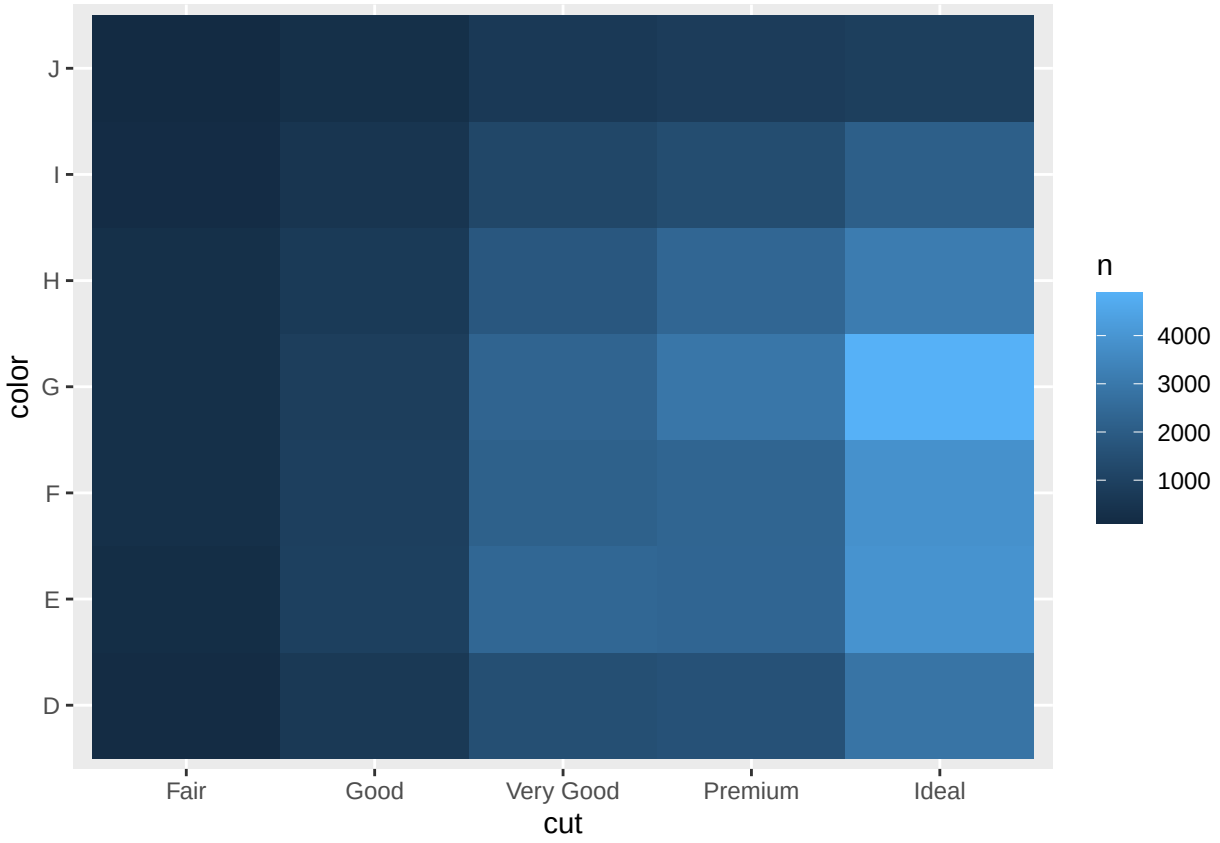
Form the above plots, we can see that carat size is the most important for predicting prices of a diamond since it correlates the most with price. The regression results reveal that 1 unit hcange in carat will lead to the highesst change of price, and we can also notice the change in median price with carat changing is the most in boxplots.

4.2

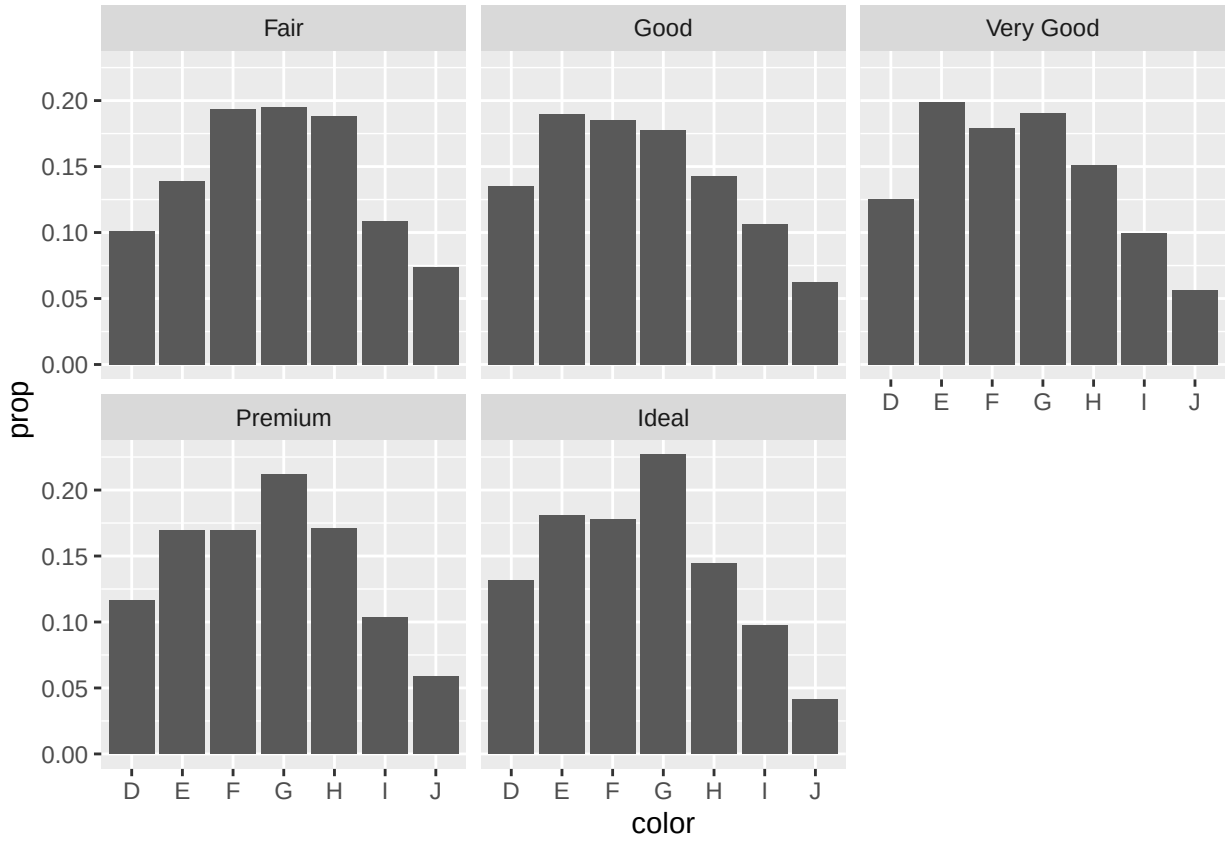
```
diamonds %>%
  count(color, cut) %>%
  ggplot(mapping = aes(x = color, y = cut)) +
    geom_tile(mapping = aes(fill = n))
```




```
diamonds %>%
  count(color, cut) %>%
  ggplot(mapping = aes(x = cut, y = color)) +
    geom_tile(mapping = aes(fill = n))
```

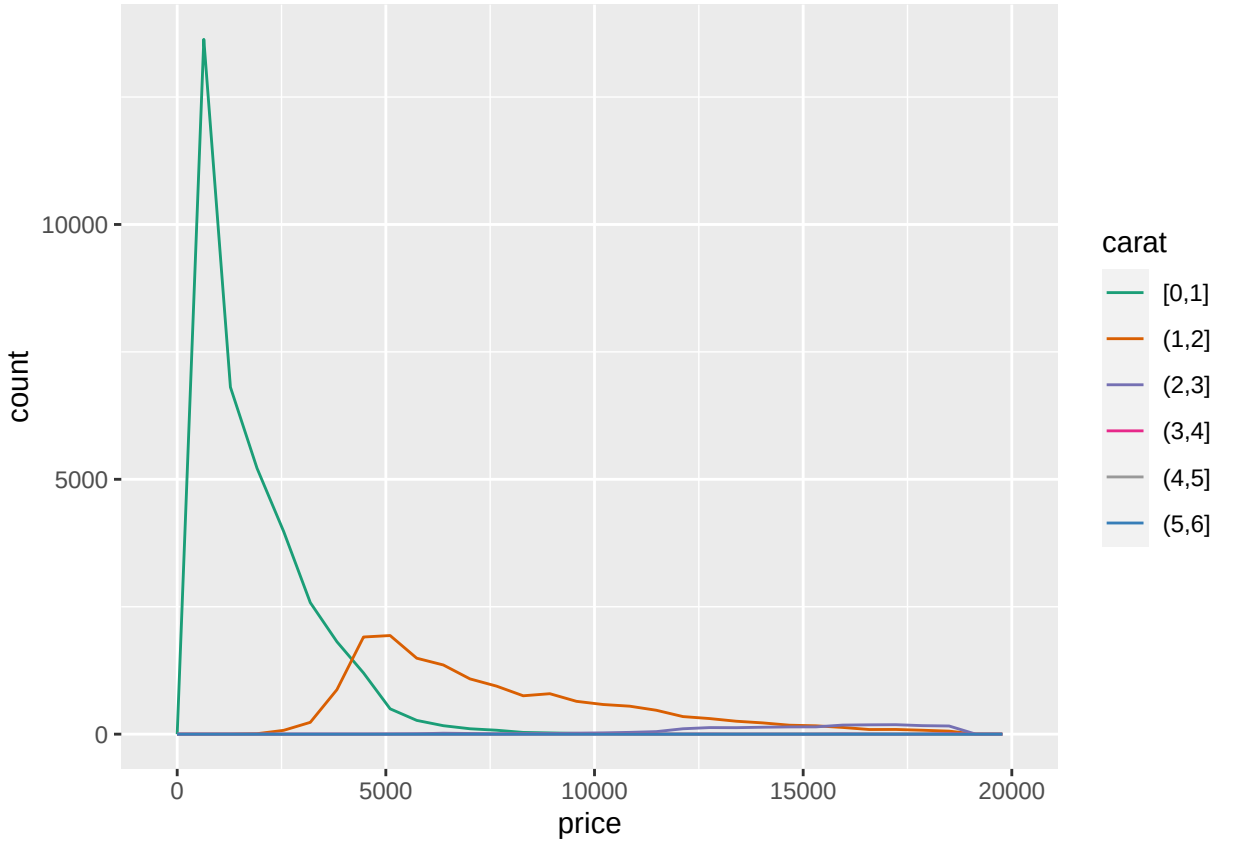



```
# Reproduce plots
diamonds %>%
  group_by(cut, color) %>%
  summarise(count = n()) %>%
  ungroup(color) %>%
  mutate(sum = sum(count)) %>%
  ggplot((aes(x = color,
              y = count/sum))) +
  geom_bar(stat = "identity") +
  labs(y = "prop") +
  facet_wrap(~cut,
             ncol = 3)
```

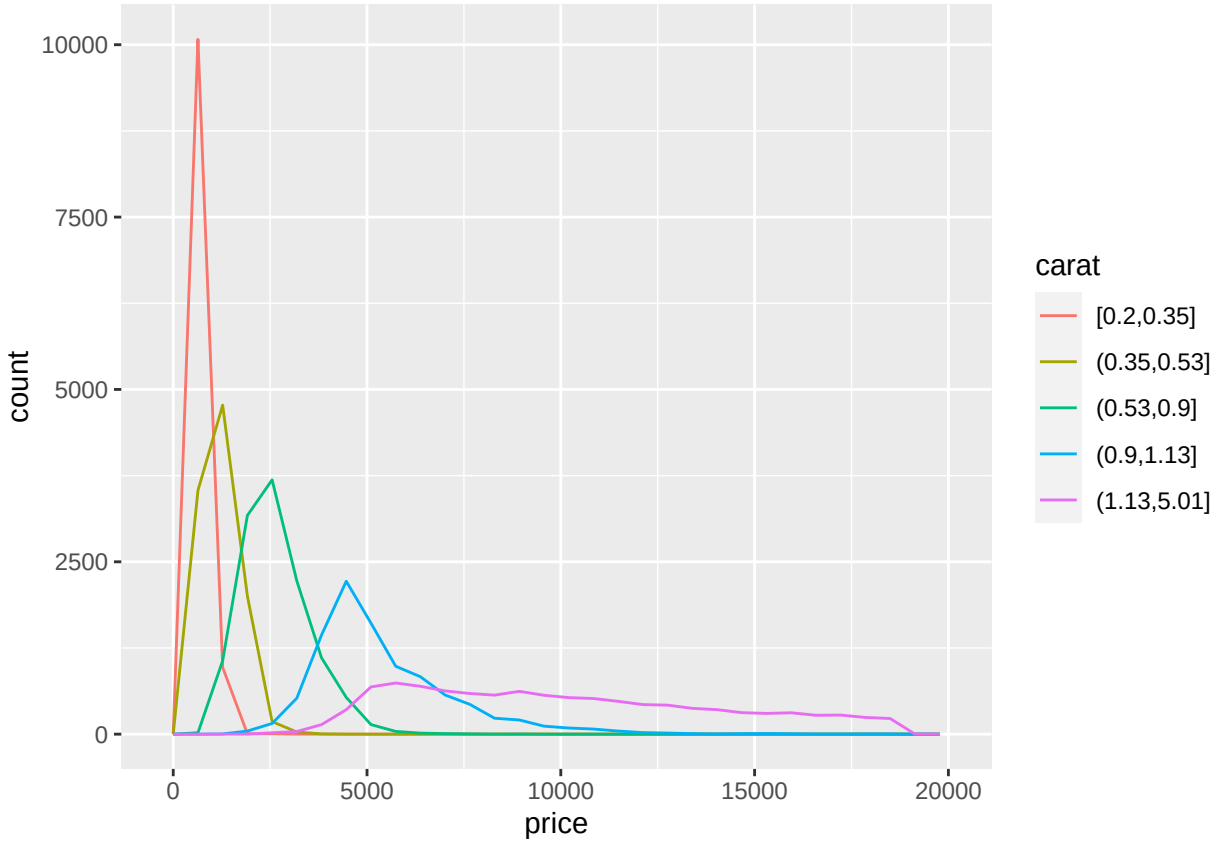


3.3

```
# Using cut_width()
ggplot(data = diamonds) +
  geom_freqpoly(aes(x = price,
                    color = cut_width(carat, 1, boundary = 0))
                ) +
  scale_color_manual(values = c("#1B9E77", "#D95F02", "#7570B3", "#E7298A",
                                "#999999", "#377EB8")) +
  labs(color = "carat")
```




```
# Using cut_number()
ggplot(data = diamonds) +
  geom_freqpoly(aes(x = price,
                    color = cut_number(carat, 5)
                    )
                ) +
  labs(color = "carat")
```



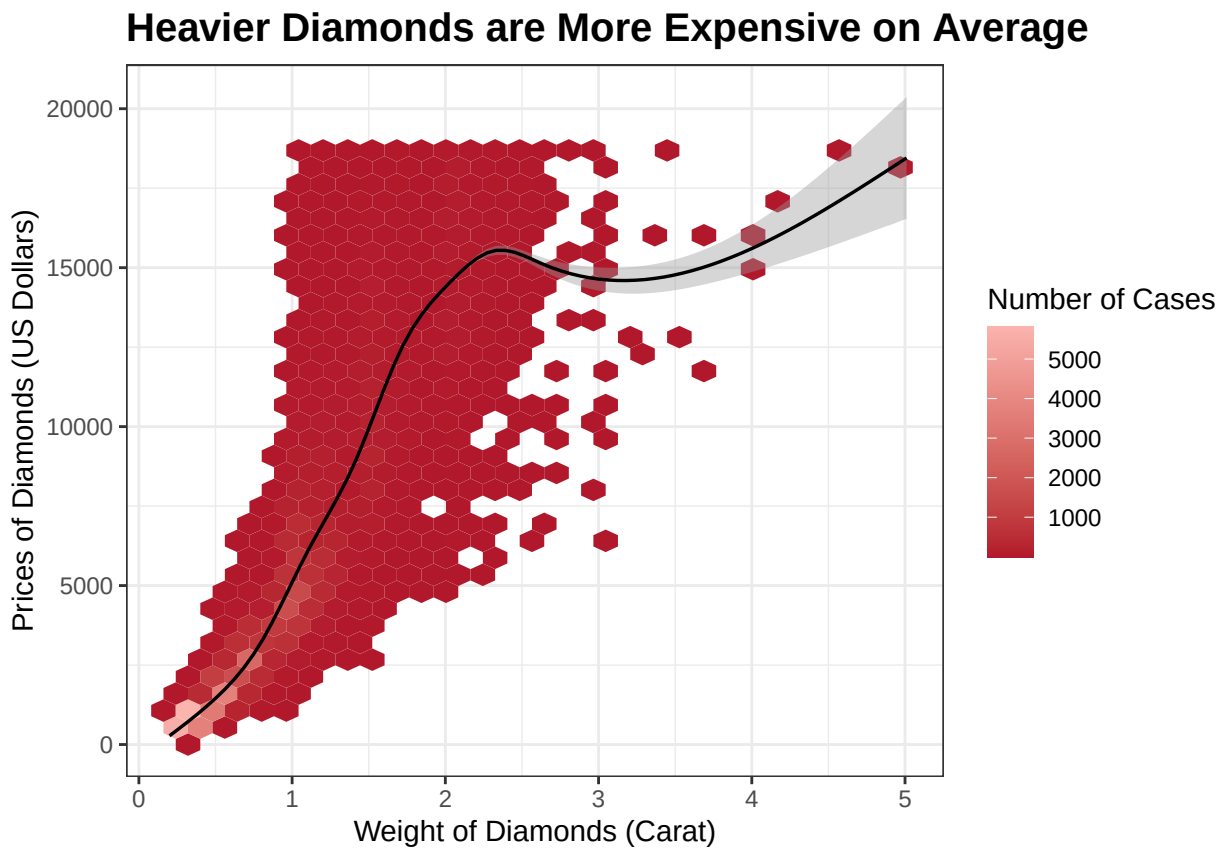
Interpretation

I used for cut_width() because

-
-

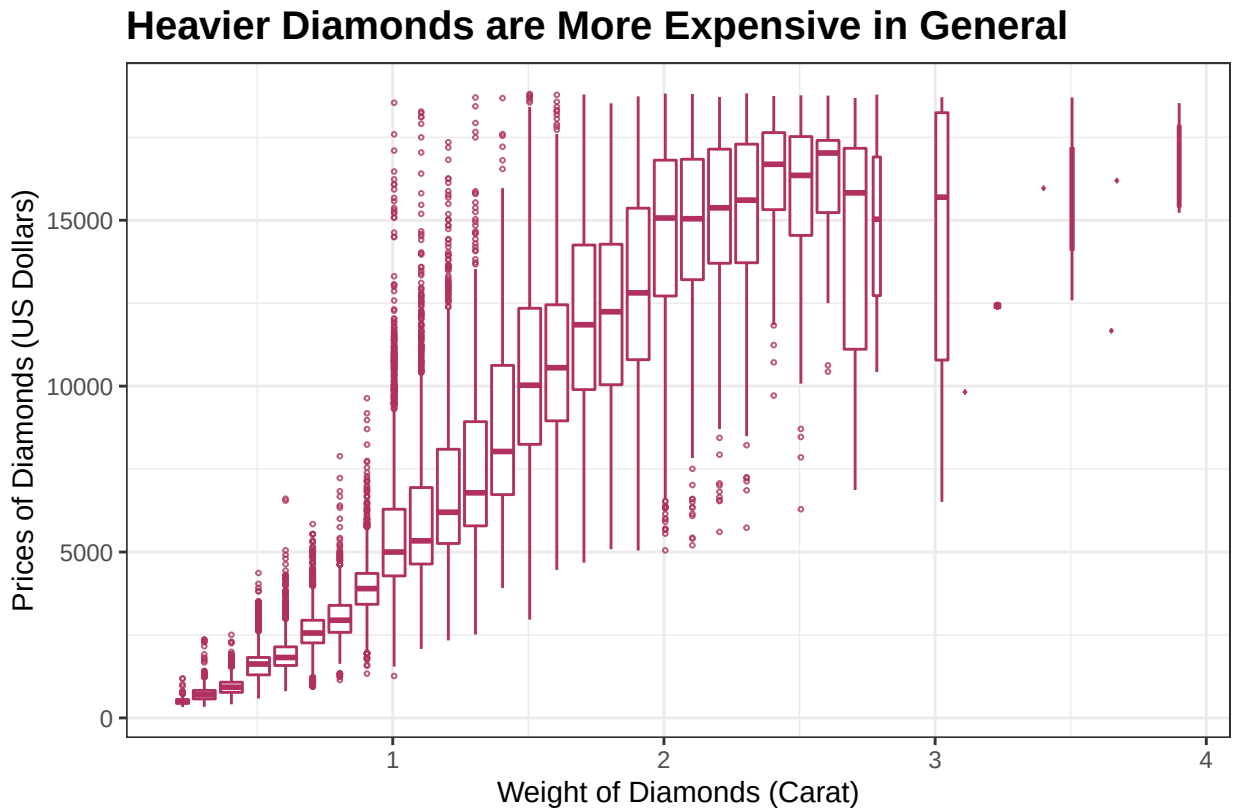
4.4

```
# Plot
ggplot(data = diamonds,aes(x = carat, y = price)) +
  geom_hex() +
  geom_smooth(method = "gam",
              color = "black",
              size = 0.6) +
  scale_fill_continuous(low = "#B2182B",
                        high = "#FBB4AE") +
  labs(title = "Heavier Diamonds are More Expensive on Average",
       x = "Weight of Diamonds (Carat)",
       y = "Prices of Diamonds (US Dollars)",
       fill = "Number of Cases") +
  theme_bw() +
  theme(plot.title = element_text(size = 15, face = "bold"))
```




```
# Winsorize carat
diamonds$carat_winsor <- winsorise(diamonds$carat)

ggplot(data = diamonds) +
  geom_boxplot(aes(x = carat_winsor, y = price,
                   group = cut_width(carat_winsor, 0.1)),
               color = "maroon",
               outlier.shape = 1,
               outlier.size  = 0.5,
               outlier.alpha = 0.8) +
  labs(title = "Heavier Diamonds are More Expensive in General",
       caption = "Data Source: R data set Diamonds, with carat values winsorized",
       x = "Weight of Diamonds (Carat)",
       y = "Prices of Diamonds (US Dollars)") +
  theme_bw() +
  theme(plot.title = element_text(size = 15, face = "bold"),
        plot.caption = element_text(hjust = 0, face = "italic"))
```



Data Source: R data set Diamonds, with carat values winsorized

Interpretation

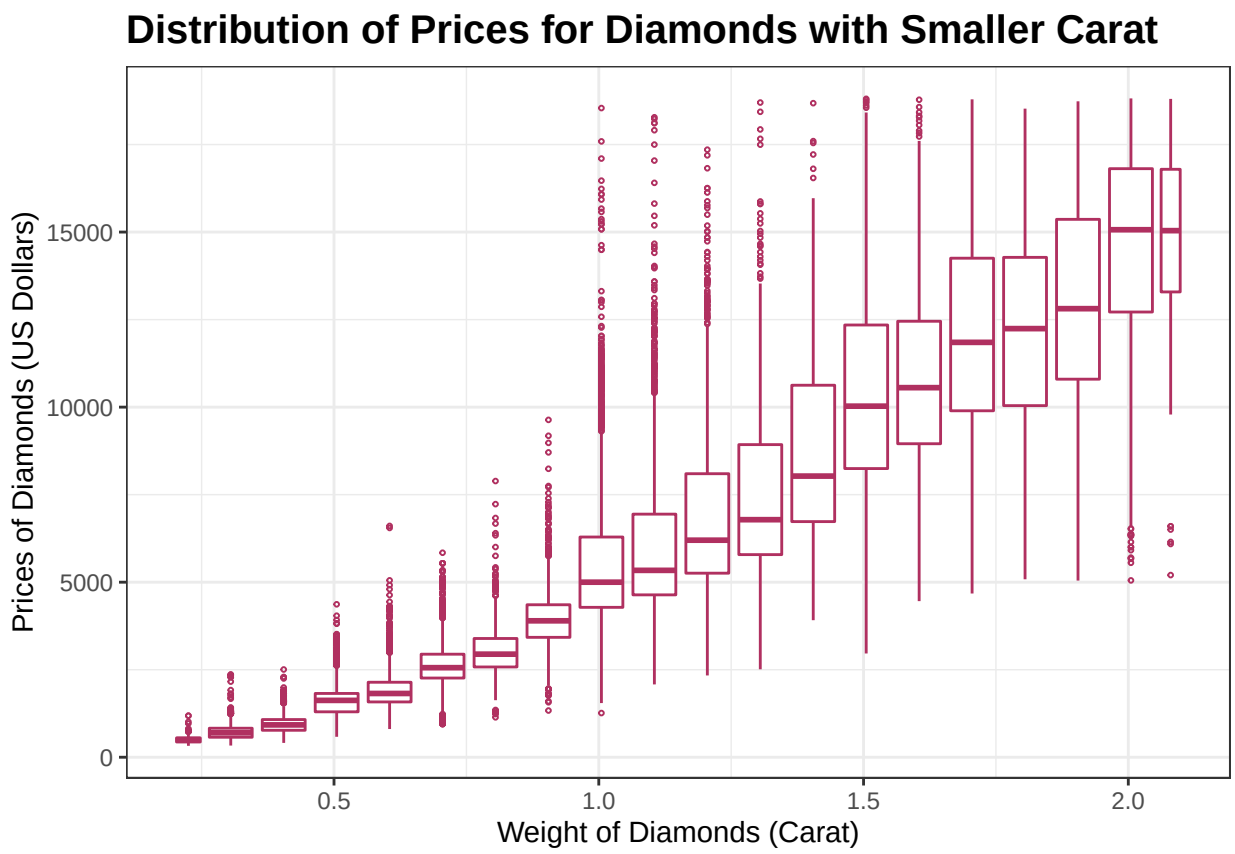
4.5

```
# Find the distribution of carat
quantile(diamonds$carat)
```

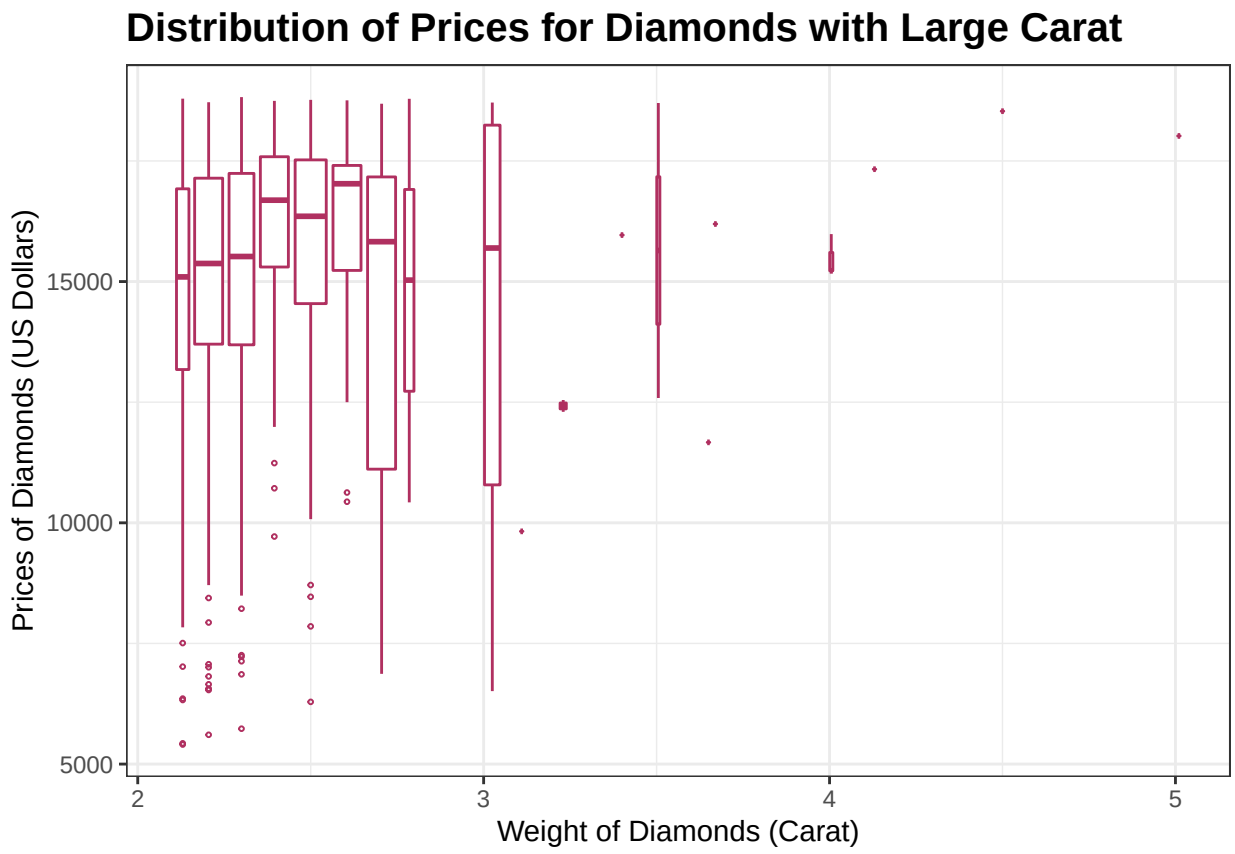
```
##   0%  25%  50%  75% 100%
## 0.20 0.40 0.70 1.04 5.01
```

```
# Subset data based on carat values
diamonds_small <-
  diamonds %>%
  filter(carat <= 2.1)
diamonds_large <-
  diamonds %>%
  filter(carat > 2.1)
# Plot distribution
ggplot(data = diamonds_small) +
  geom_boxplot(aes(x = carat, y = price,
                   group = cut_width(carat, 0.1)),
               color = "maroon",
               outlier.shape = 1,
               outlier.size  = 0.5) +
```

```
  labs(title = "Distribution of Prices for Diamonds with Smaller Carat",
       x = "Weight of Diamonds (Carat)",
       y = "Prices of Diamonds (US Dollars)") +
  theme_bw() +
  theme(plot.title = element_text(size = 15, face = "bold"))
```



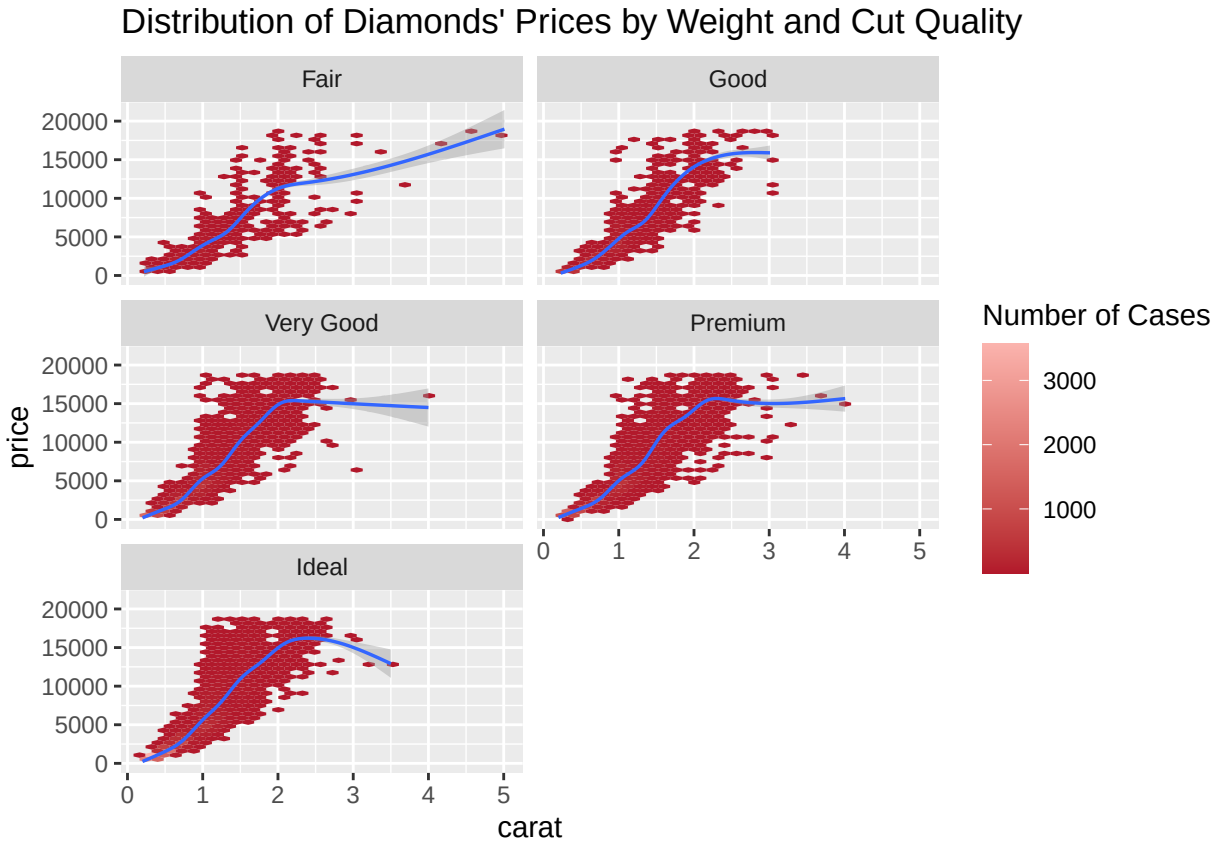

```
ggplot(data = diamonds_large) +
  geom_boxplot(aes(x = carat, y = price,
                   group = cut_width(carat, 0.1)),
               color = "maroon",
               outlier.shape = 1,
               outlier.size  = 0.5) +
  labs(title = "Distribution of Prices for Diamonds with Large Carat",
       x = "Weight of Diamonds (Carat)",
       y = "Prices of Diamonds (US Dollars)") +
  theme_bw() +
  theme(plot.title = element_text(size = 15, face = "bold"))
```



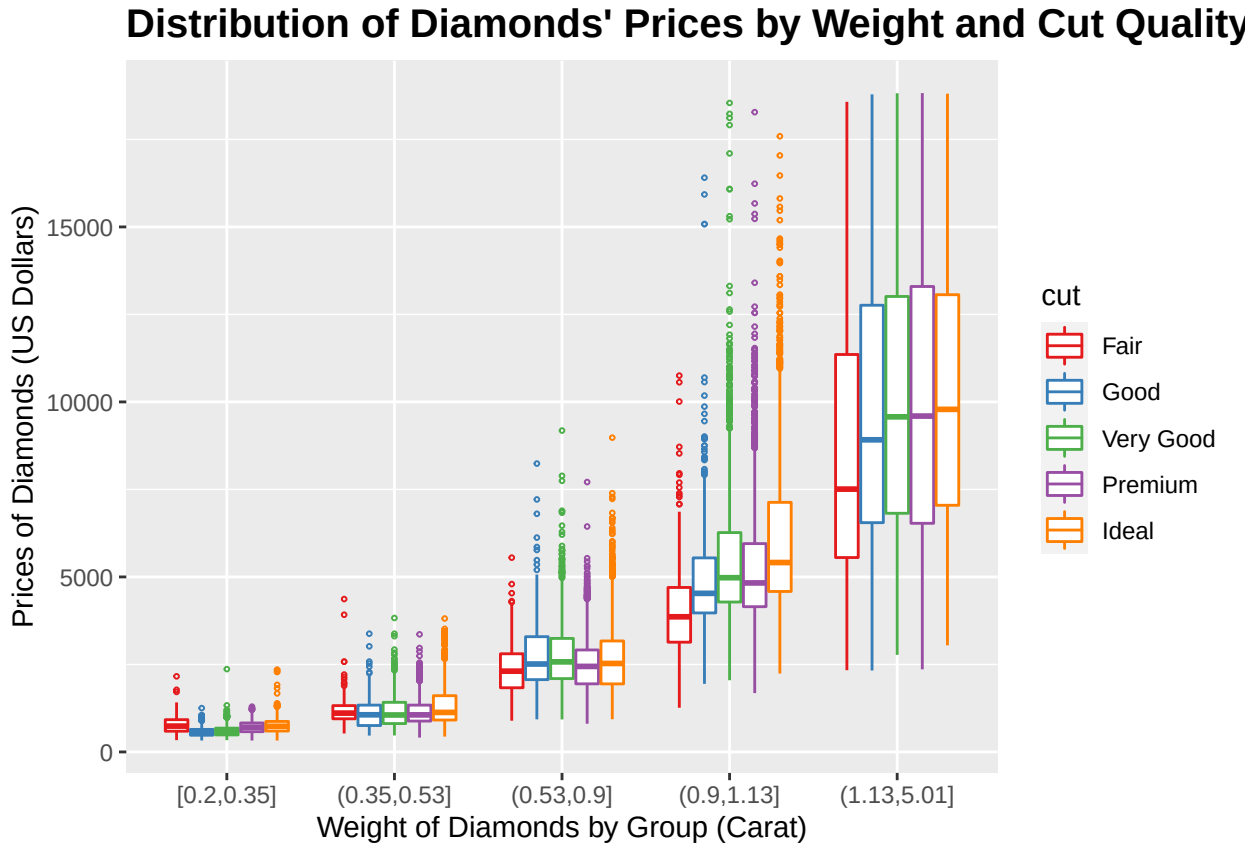
Interpretation

- The distribution of very large diamonds is more variable. ### 4.6

```
# Facet
ggplot(data = diamonds,
       aes(x = carat, y = price)) +
  geom_hex() +
  geom_smooth(size = 0.6) +
  scale_fill_continuous(low = "#B2182B",
                        high = "#FBB4AE") +
  labs(title = "Distribution of Diamonds' Prices by Weight and Cut Quality",
       fill = "Number of Cases") +
  facet_wrap(~cut,
             ncol = 2)
```

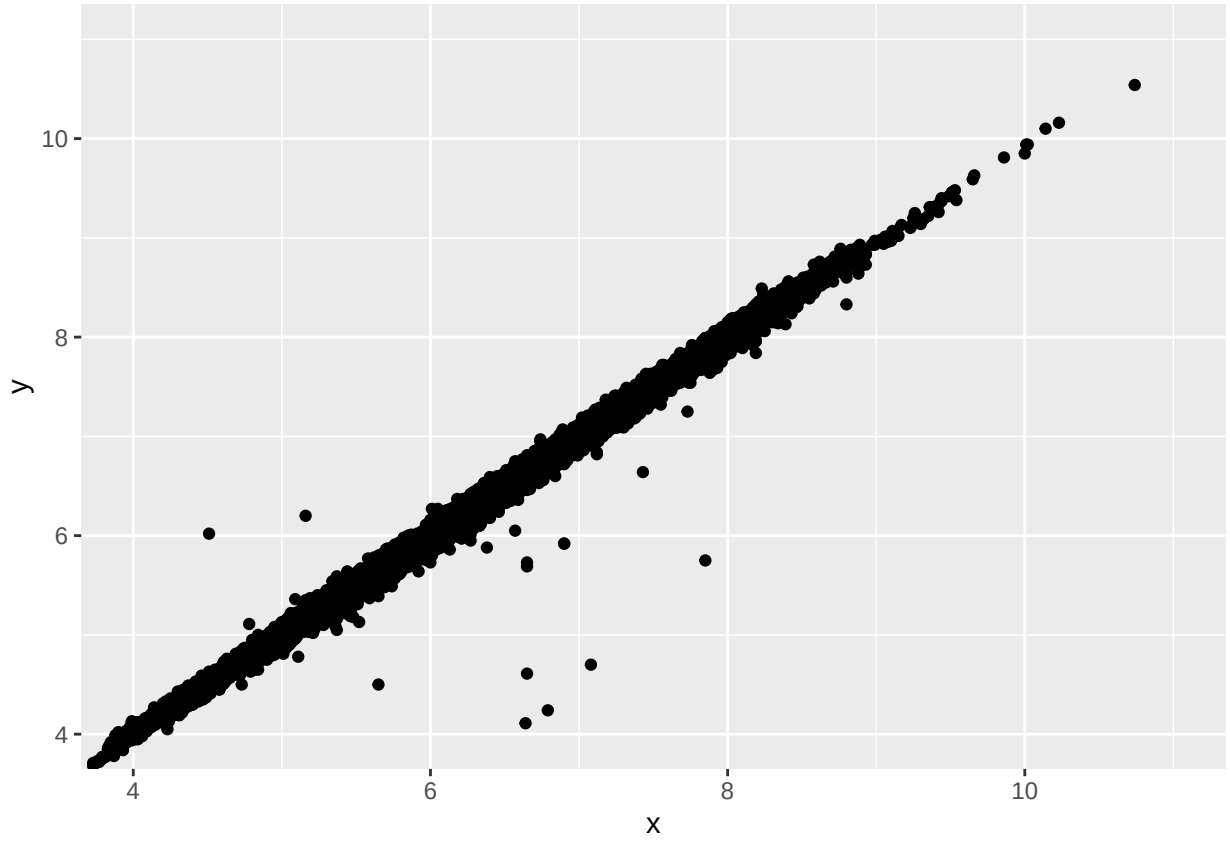



```
# Add legend dimension
ggplot(data = diamonds,
       aes(x = cut_number(carat, 5), y = price, colour = cut)) +
  geom_boxplot(outlier.shape = 1,
               outlier.size = 0.5) +
  scale_color_brewer(palette = "Set1") +
  labs(title = "Distribution of Diamonds' Prices by Weight and Cut Quality",
       x = "Weight of Diamonds by Group (Carat)",
       y = "Prices of Diamonds (US Dollars)") +
  theme(plot.title = element_text(size = 15, face = "bold"))
```

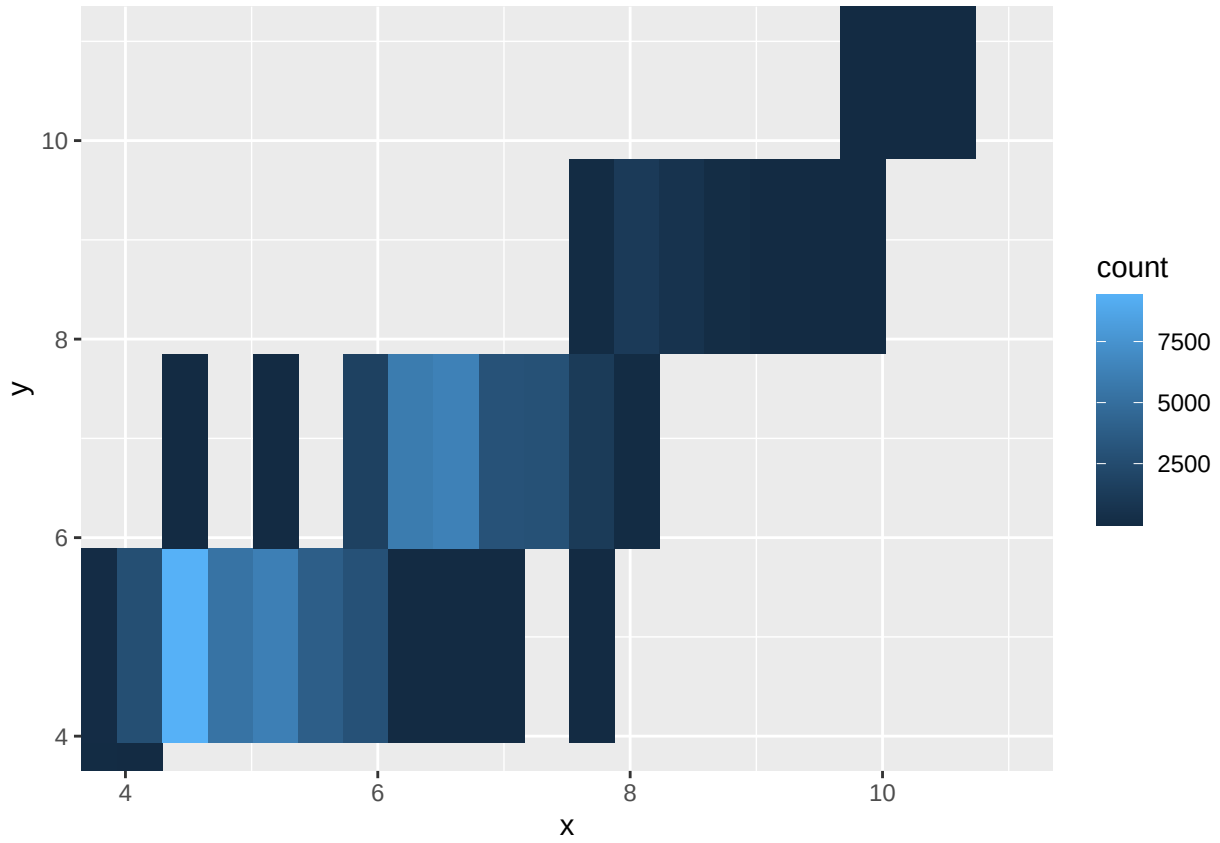


4.7

```
# Plot 1
 ggplot(data = diamonds) +
  geom_point(mapping = aes(x = x, y = y)) +
  coord_cartesian(xlim = c(4, 11), ylim = c(4, 11))
```



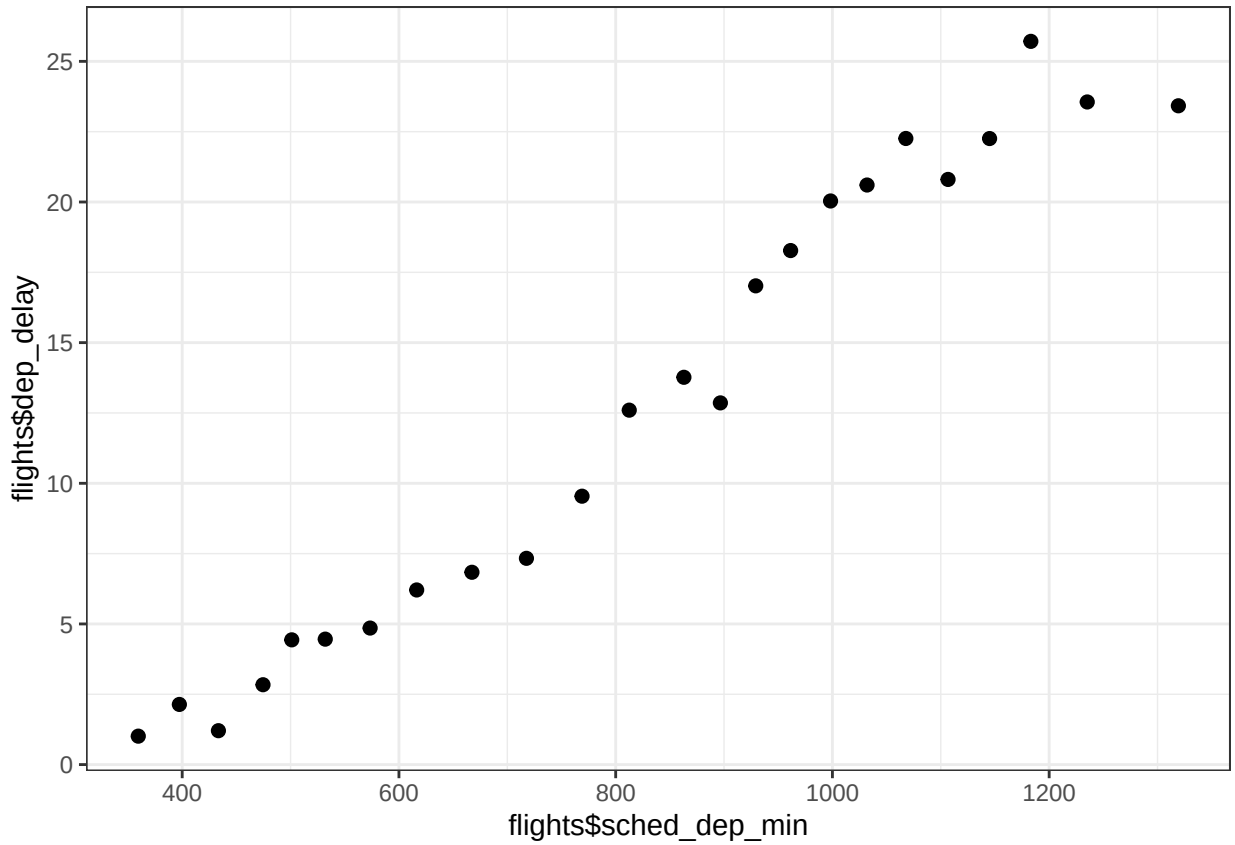

```
# Plot 2
ggplot(data = diamonds) +
  geom_bin2d(mapping = aes(x = x, y = y)) +
  coord_cartesian(xlim = c(4, 11), ylim = c(4, 11))
```



Interpretation

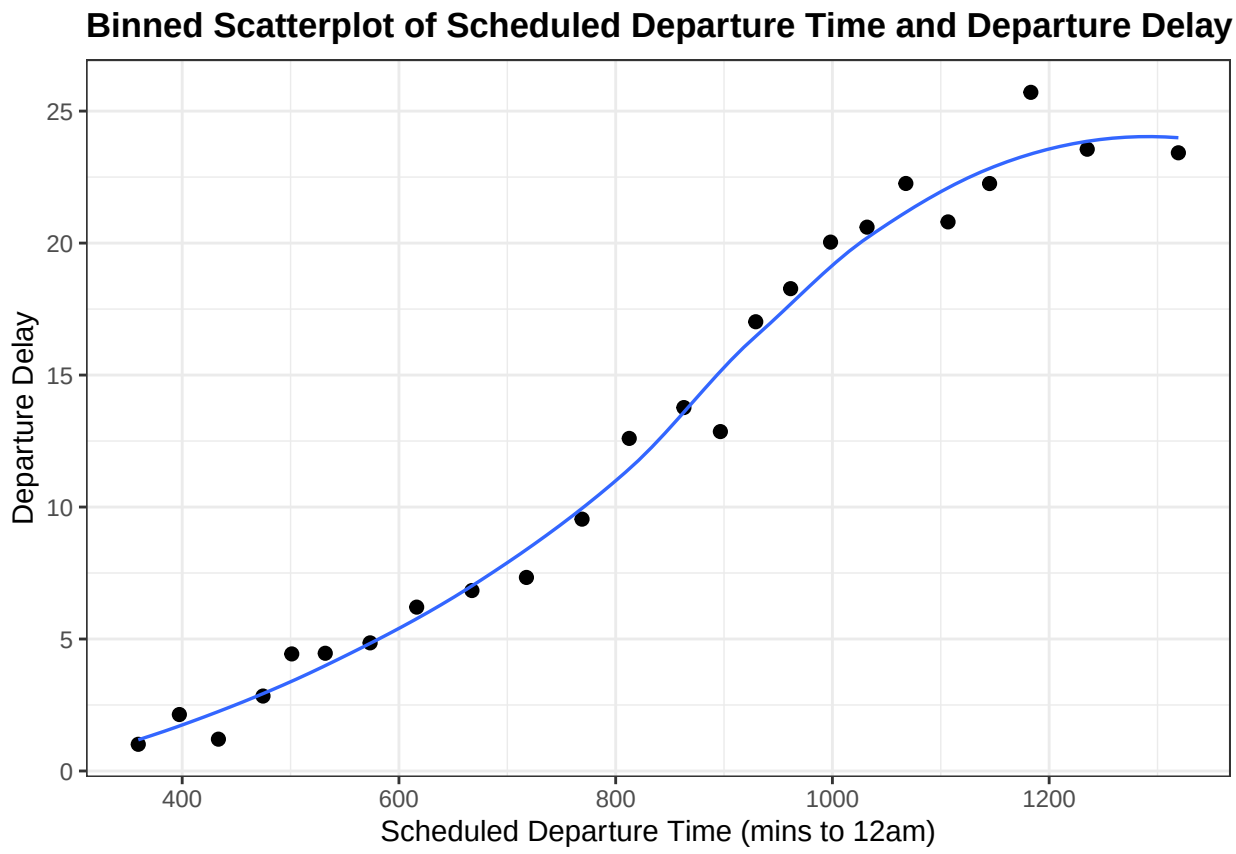
5

```
# Load and clean dataset
flights <- nycflights13::flights
flights <- flights %>%
  mutate(sched_dep_min = (sched_dep_time %/% 100 * 60 + sched_dep_time %% 100) %% 1440) %>%
  select(sched_dep_min,dep_delay,everything())
# Binsreg
binscatter <- binsreg(flights$dep_delay,flights$sched_dep_min,
                      nbins = 24,
                      bycolors = "black")
```

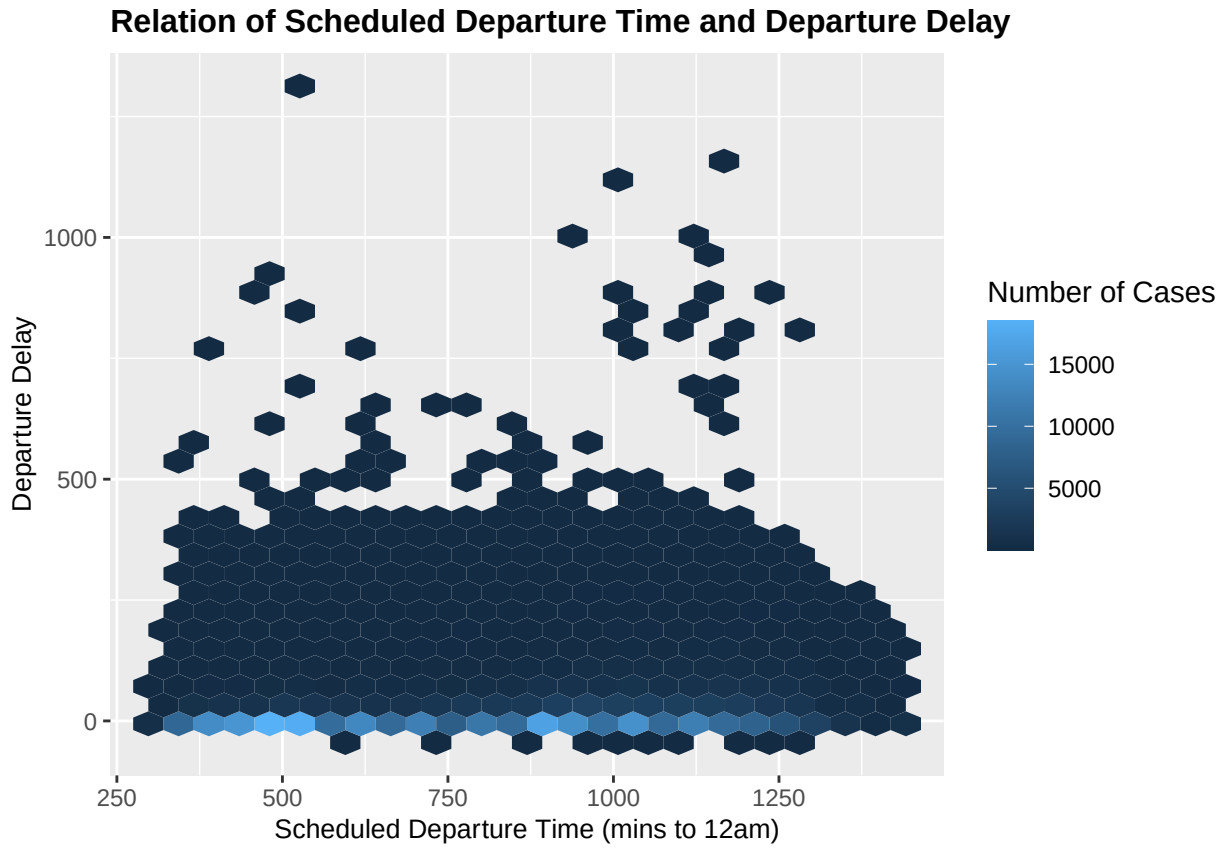



```
bins_data <- as.data.frame(binscatter$data.plot)
colnames(bins_data)[2] <- "sched_dep_time"
colnames(bins_data)[6] <- "dep_delay"
```

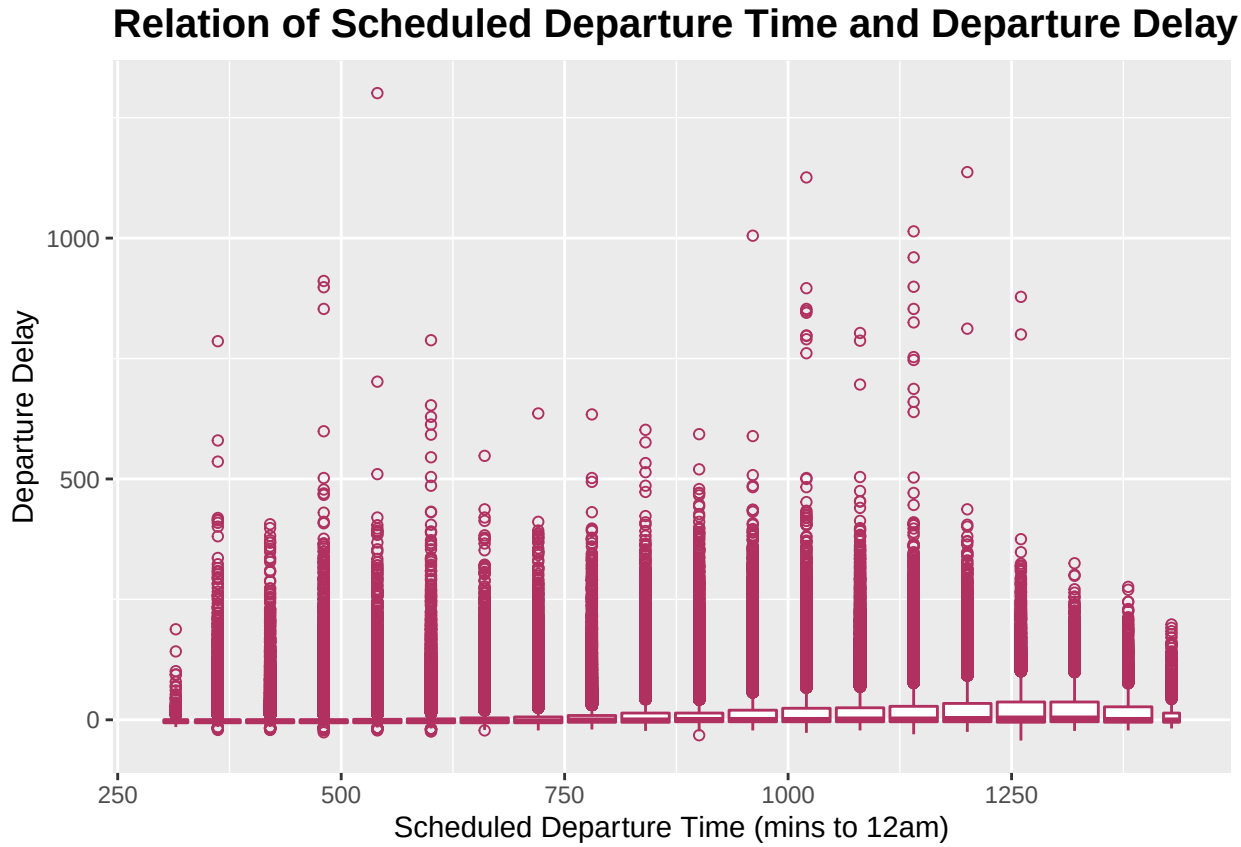
```
# Binsreg
binscatter$bins_plot +
  geom_smooth(data = bins_data,
              aes(x = sched_dep_time, y = dep_delay),
              se = FALSE,
              size = 0.6) +
  labs(title = "Binned Scatterplot of Scheduled Departure Time and Departure Delay",
       x = "Scheduled Departure Time (mins to 12am)",
       y = "Departure Delay") +
  theme(plot.title=element_text(face='bold'))
```



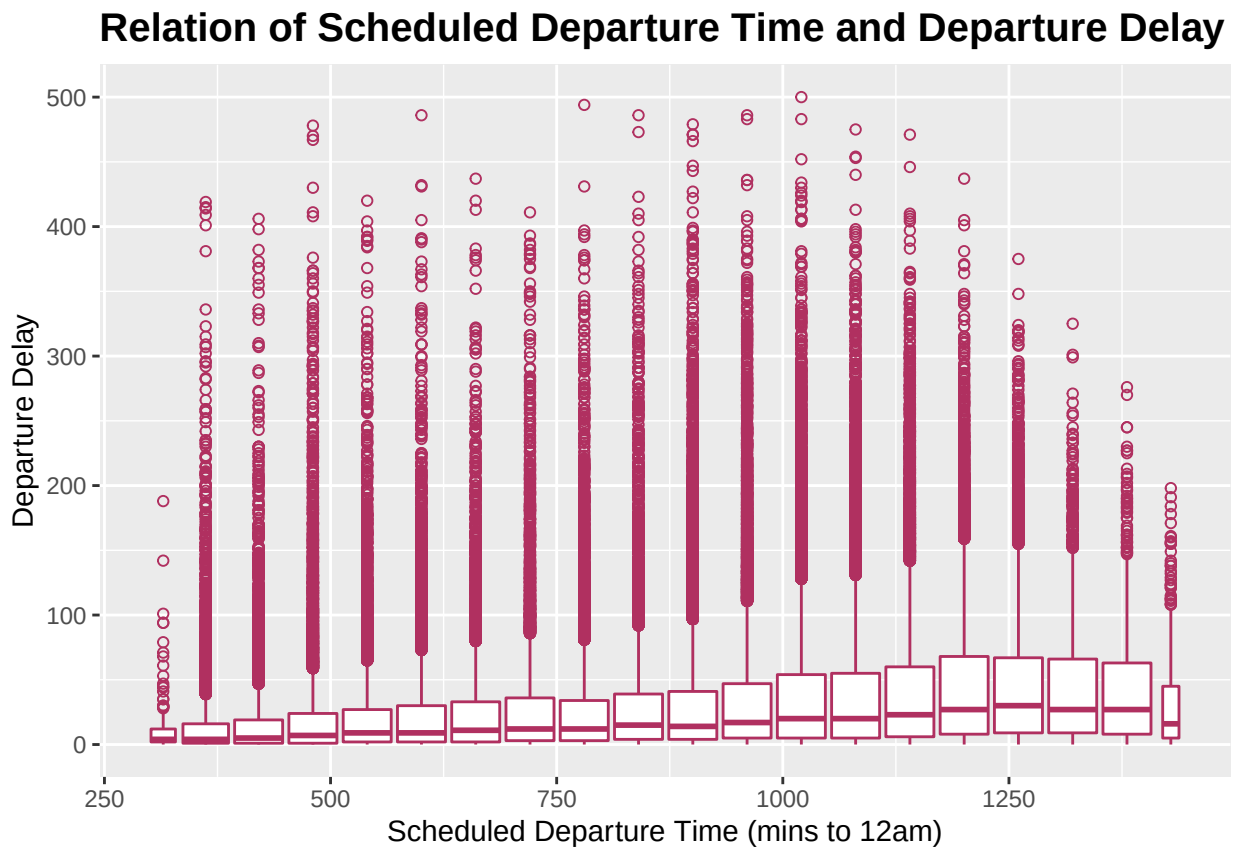

```
# Geom_hex
ggplot(data = flights) +
  geom_hex(aes(x = sched_dep_min, y = dep_delay)) +
  labs(title = "Relation of Scheduled Departure Time and Departure Delay",
       x = "Scheduled Departure Time (mins to 12am)",
       y = "Departure Delay",
       fill = "Number of Cases") +
  theme(plot.title = element_text(size = 12, face='bold'),
        axis.title = element_text(size = 10))
```




```
# Geom_boxplot
boxplot <- ggplot(data = flights) +
  geom_boxplot(aes(x = sched_dep_min, y = dep_delay,
                   group = cut_width(sched_dep_min, 60),
                   orientation = "x"),
               color = "maroon",
               outlier.shape = 1) +
  labs(title = "Relation of Scheduled Departure Time and Departure Delay",
       x = "Scheduled Departure Time (mins to 12am)",
       y = "Departure Delay") +
  theme(plot.title = element_text(size = 15, face = "bold"))
boxplot
```




```
boxplot +
  ylim(0,500)
```



Interpretation

6 R4DS Chapter 10 and 11

6.1

```
# Dataframe
d_f <- data.frame(abc = 1, xyz = "a")
# Tibble
d_f_tibble <-
  d_f %>%
  as_tibble()
# First pair
d_f$x
```

```
## [1] "a"
```

```
d_f_tibble$x
```

```
## NULL
```

```
# Second pair
d_f[, "xyz"]
```

```
## [1] "a"
```

```
d_f_tibble[, "xyz"]
```

```
## # A tibble: 1 x 1
##   xyz
##   <chr>
## 1 a
```

```
# Third pair
d_f[, c("abc", "xyz")]
```

```
##   abc xyz
## 1   1   a
```

```
d_f_tibble[, c("abc", "xyz")]
```

```
## # A tibble: 1 x 2
##     abc xyz
##   <dbl> <chr>
## 1     1 a
```

6.2

```
read_delim("x|y\n1|2\n3|4", delim = "|",
                   col_types = cols(
                     x = col_double(),
                     y = col_double()
                   ))
```

```
## # A tibble: 2 x 2
##       x     y
##   <dbl> <dbl>
## 1     1     2
## 2     3     4
```

6.3

```
read_csv("a,b\n1,2,3\n4,5,6")
```

```
## # A tibble: 2 x 2
##       a     b
##   <dbl> <dbl>
## 1     1    23
## 2     4    56
```

```
read_csv("a,b,c\n1,2\n1,2,3,4")
```

```
## # A tibble: 2 x 3
##       a     b     c
##   <dbl> <dbl> <dbl>
## 1     1     2    NA
## 2     1     2    34
```

```
read_csv("a,b\n\"1")
```

```
## # A tibble: 0 x 2
## # ... with 2 variables: a <chr>, b <chr>
```

```
read_csv("a,b\n1,2\na,b")
```

```
## # A tibble: 2 x 2
##   a     b
##   <chr> <chr>
## 1 1     2
## 2 a     b
```

```
read_csv("a;b\n1;3")
```

```
## # A tibble: 1 x 1
##   `a;b`
##   <chr>
## 1 1;3
```

```
read_csv("x,y\n1,'a,b'")
```

```
## # A tibble: 1 x 2
##       x y
##   <dbl> <chr>
## 1     1 'a,b'
```

6.4

6.5

```
d1 <- "January 1, 2010"
parse_date(d1, "%B %d, %Y")
```

```
## [1] "2010-01-01"
```

```
d2 <- "2015-Mar-07"
parse_date(d2, "%Y-%b-%d")
```

```
## [1] "2015-03-07"
```

```
d3 <- "06-Jun-2016"
parse_date(d3, "%d-%b-%Y")
```

```
## [1] "2016-06-06"
```

```
d4 <- c("August 19 (2015)", "July 1 (2015)")
parse_date(d4, "%B %d (%Y)")
```

```
## [1] "2015-08-19" "2015-07-01"
```

```
d5 <- "12/30/14" # Dec 30, 2014
parse_date(d5, "%m/%d/%y")
```

```
## [1] "2014-12-30"
```

```
t1 <- "1805" # 6:05 pm
parse_time(t1, "%H%M")
```

```
## 18:05:00
```

```
t2 <- "11:25:10.12 PM"
parse_time(t2, "%H:%M:%OS %p")
```

```
## 23:25:10.12
```

6.6

```
massey <- readr_example("massey-rating.txt")
writeLines(read_lines(massey))
```

```
## UCC PAY LAZ KPK  RT   COF BIH DII ENG ACU Rank Team           Conf
##   1   1   1   1   1     1   1   1   1   1    1 Ohio St         B10
##   2   2   2   2   2     2   2   2   4   2    2 Oregon          P12
##   3   4   3   4   3     4   3   4   2   3    3 Alabama         SEC
##   4   3   4   3   4     3   5   3   3   4    4 TCU             B12
##   6   6   6   5   5     7   6   5   6  11    5 Michigan St     B10
##   7   7   7   6   7     6  11   8   7   8    6 Georgia         SEC
##   5   5   5   7   6     8   4   6   5   5    7 Florida St      ACC
##   8   8   9   9  10     5   7   7  10   7    8 Baylor          B12
##   9  11   8  13  11    11  12   9  14   9    9 Georgia Tech    ACC
##  13  10  13  11   8     9  10  11   9  10   10 Mississippi     SEC
```

```
fwf_empty(massey)
```

```
## $begin
##  [1]  0  4  8 12 17 22 26 30 34 38 42 47 63
##
```

```
## $end
##  [1]  3  7 11 15 19 25 29 33 37 41 46 59 NA
##
## $col_names
##  [1] "X1"  "X2"  "X3"  "X4"  "X5"  "X6"  "X7"  "X8"  "X9"  "X10" "X11" "X12"
## [13] "X13"
```