---

title: “Data Wrangling and Exploratory Data Analysis” author: “Ananya Sharma” date: “1/30/2024” output: pdf_document

This project focuses on data manipulation and exploratory data analysis (EDA). The tasks involve creating tibbles, performing operations on tuberculosis and WHO data, exploring the distribution of variables in the diamonds dataset, comparing distributions, making informative visualizations, and addressing questions related to data analysis and interpretation. Ananya demonstrates proficiency in utilizing functions from the tidyverse ecosystem to clean, transform, and visualize data, showcasing the practical application of these skills in real-world scenarios. The analysis also involves thoughtful considerations of data nuances, such as missing values and skewed distributions, and the creation of meaningful visualizations to extract insights from complex datasets.

Q. Can you provide a code snippet that creates a tibble named ‘data’ containing information about tuberculosis cases and population data for different countries in the years 1999 and 2000?

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tibble)
```

```
## Warning: package 'tibble' was built under R version 4.1.2
```

```
data <- tribble(
  ~country, ~type, ~"1999", ~"2000",
  "afghanistan", "cases", 745, 2666,
  "afghanistan", "population", 19987071, 20595360,
  "brazil", "cases", 37737, 80488,
  "brazil", "population", 172006362, 174504898,
  "china", "cases", 212258, 213766,
  "china", "population", 1272915272, 1280428583
)

data
```

```
## # A tibble: 6 x 4
##   country     type           `1999`     `2000`
##   <chr>       <chr>           <dbl>      <dbl>
## 1 afghanistan cases             745       2666
```

```
## 2 afghanistan population   19987071   20595360
## 3 brazil      cases           37737      80488
## 4 brazil      population  172006362  174504898
## 5 china       cases          212258     213766
## 6 china       population 1272915272 1280428583
```

Q. Perform four operations a. Extract the number of TB cases per country per year. b. Extract the matching population per country per year. c. Divide cases by population, and multiply by 10000. d. Convert the data back to the table3 format

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.1.2
```

```
#create table 3
table2 <- table1 |>
pivot_longer(cases:population, names_to = "type", values_to = "count")
table3 <- table2 |>
pivot_wider(names_from = year, values_from = count)
# create rate
table3_ans <- table3 |>
pivot_longer(cols = c("1999", "2000"), names_to = "year") |>
pivot_wider(names_from = "type") |>
mutate(rate = (cases * 1e4) / population) |>
pivot_longer(cols = c("cases", "population", "rate"), names_to = "type") |>
pivot_wider(names_from = "year")
table3_ans
```

```
## # A tibble: 9 x 4
##   country     type        `1999`        `2000`
##   <chr>       <chr>        <dbl>         <dbl>
## 1 Afghanistan cases      7.45e+2       2666
## 2 Afghanistan population 2.00e+7   20595360
## 3 Afghanistan rate       3.73e-1          1.29
## 4 Brazil      cases      3.77e+4      80488
## 5 Brazil      population 1.72e+8  174504898
## 6 Brazil      rate       2.19e+0          4.61
## 7 China       cases      2.12e+5     213766
## 8 China       population 1.27e+9 1280428583
## 9 China       rate       1.67e+0          1.67
```

Following questions are based on the case study - https://r4ds.had.co.nz/tidy-data.html#case-study

Q. What happens if you neglect the mutate() step? (mutate(key = stringr::str_replace(key, newrel, new_rel))) Answer - All variable names contain 3 parts of information - "new", the variable being measured and the age-sex, and we are looking to separate it using the "_". Since some names are newrel_agesex, when we perform separate, for these names, they will not separate into 3 parts and will separate only into 2 leading to problems as out of the 3 columns,"new" will contain "newrel", "var" will contain agesex info and "agesex" will contain NAs. T

```
who1 <- who %>%
pivot_longer(
```

```
cols = new_sp_m014:newrel_f65,
names_to = "key",
values_to = "cases",
values_drop_na = TRUE
) %>%
separate(key, c("new", "var", "sexage")) %>%
select(-new, -iso2, -iso3) %>%
separate(sexage, c("sex", "age"), sep = 1)
```

```
## Warning: Expected 3 pieces. Missing pieces filled with `NA` in 2580 rows [243, 244, 679,
## 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 903, 904, 905,
## 906, ...].
```

Q. Health outcomes are often sexed. As in certain maladies are more associated with males or females. Using the tidied WHO data, you will make an informative visualization to address the : "To what extent is Tuberculosis associated with a specific sex and has this changed from 1997 onward?" Begin with the following: a. For each country, year, and sex compute the total number of cases of TB. b. Using raw values is probably not going to provide clear evidence. Why not? c. For each country-year, compute the ratio of male to female patients. d. Producing these ratios by year (ignoring country) is probably a bad idea. Why? e. Make a data visualization that addresses the question - "To what extent is Tuberculosis associated with a specific sex and has this changed from 1997 onward?"

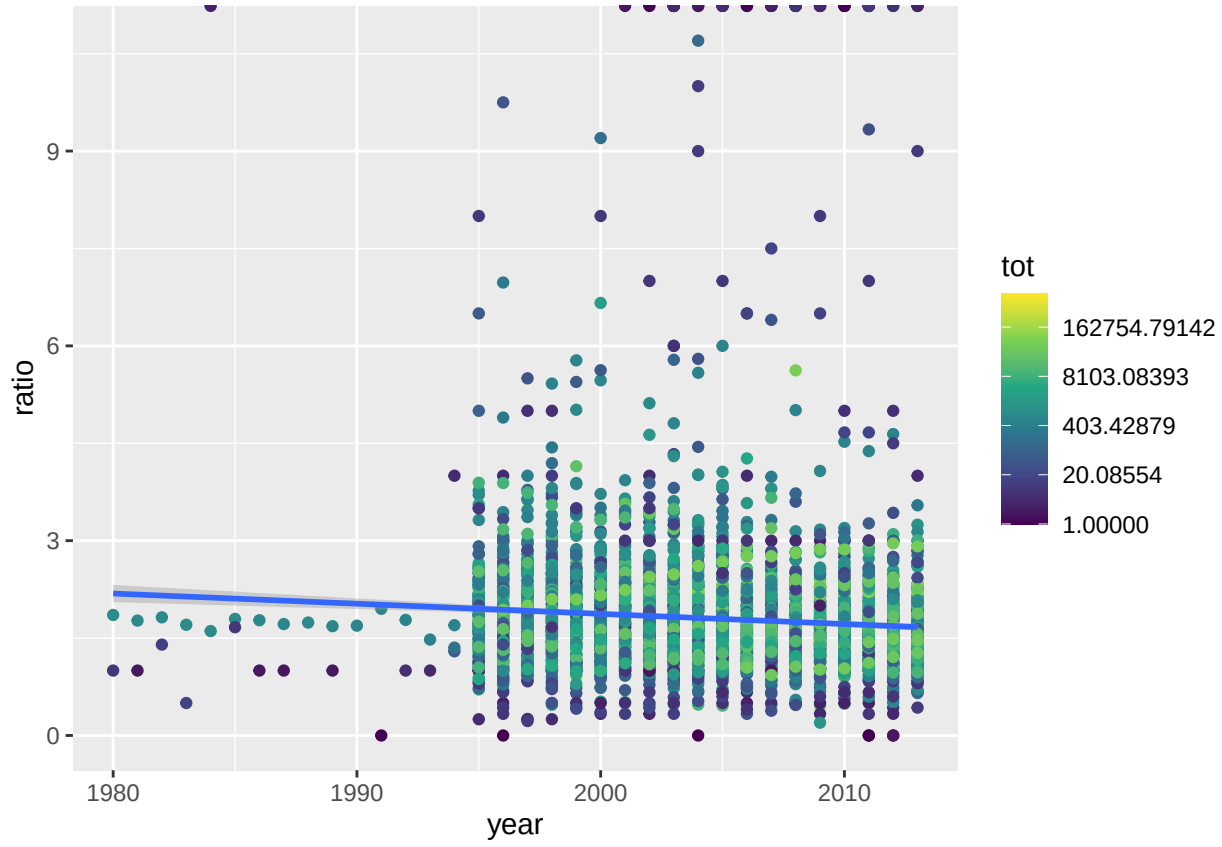
```
library(ggplot2)

who_clean <- who %>%
pivot_longer(
cols = new_sp_m014:newrel_f65,
names_to = "key",
values_to = "cases",
values_drop_na = TRUE
) %>%
mutate(
key = stringr::str_replace(key, "newrel", "new_rel")
) %>%
separate(key, c("new", "var", "sexage")) %>%
select(-new, -iso2, -iso3) %>%
separate(sexage, c("sex", "age"), sep = 1)
who_clean <- who_clean |>
group_by(country, year, sex) |>
summarise(tb = sum(cases)) |>
pivot_wider(names_from = "sex", values_from = "tb") |>
mutate(tot = m + f, ratio = m / f)
```

```
## `summarise()` has grouped output by 'country', 'year'. You can override using
## the `.groups` argument.
```

```
ggplot(data = who_clean, aes(x = year, y = ratio, color = tot)) +
geom_point() +
scale_color_viridis_c(trans = "log") +
geom_smooth(method = "lm")
```

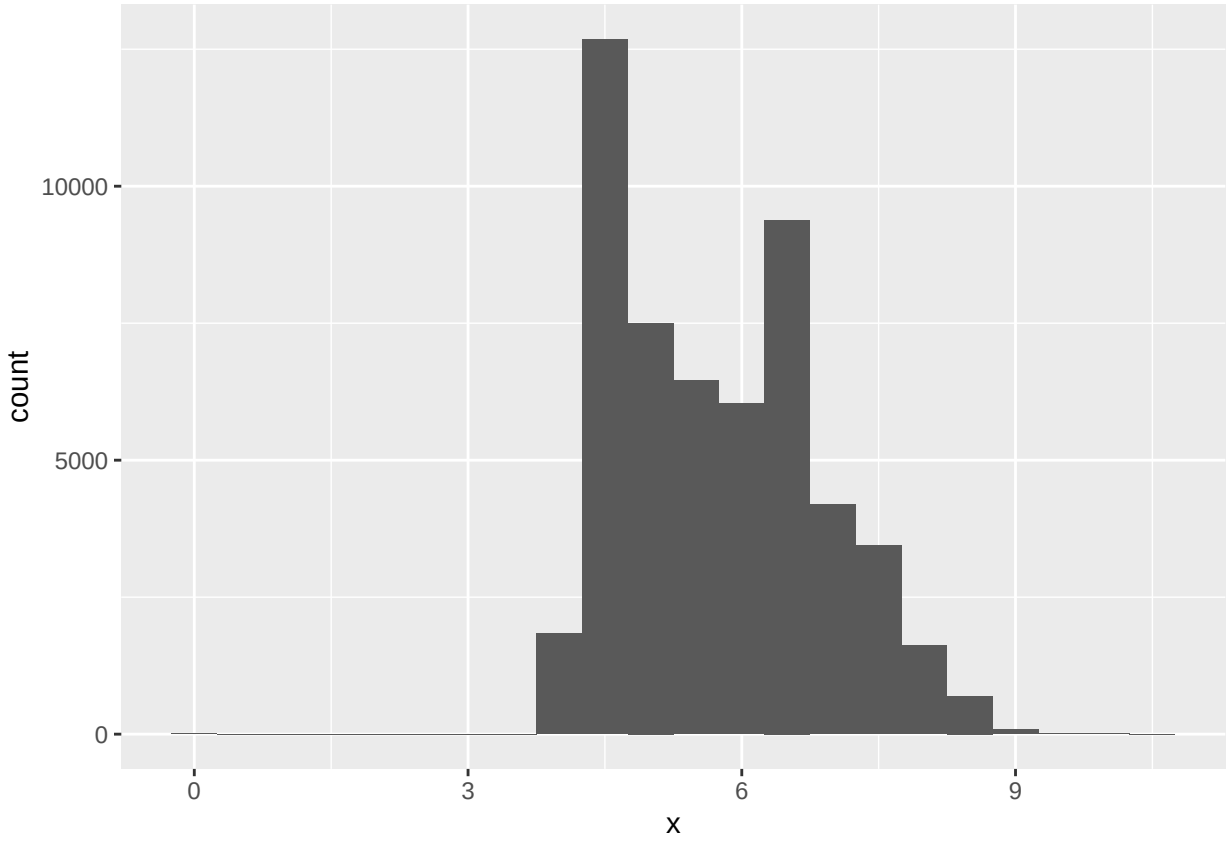
```
## Warning: Transformation introduced infinite values in discrete y-axis
## Transformation introduced infinite values in discrete y-axis
```

```
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 136 rows containing non-finite values (`stat_smooth()`).

## Warning: The following aesthetics were dropped during statistical transformation: colour
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?

## Warning: Removed 95 rows containing missing values (`geom_point()`).
```

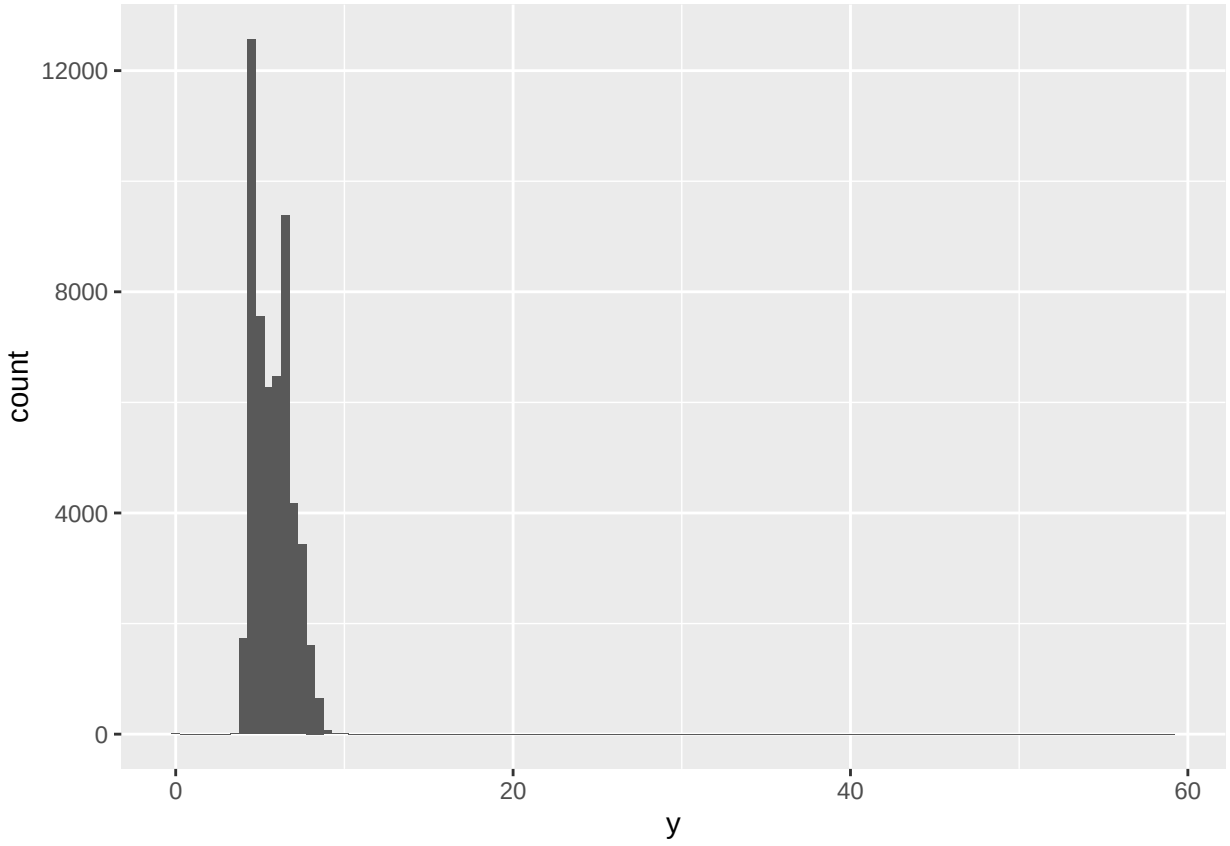


Q. Exploring variation: Explore the distribution of each of the x, y, and z variables in diamonds.

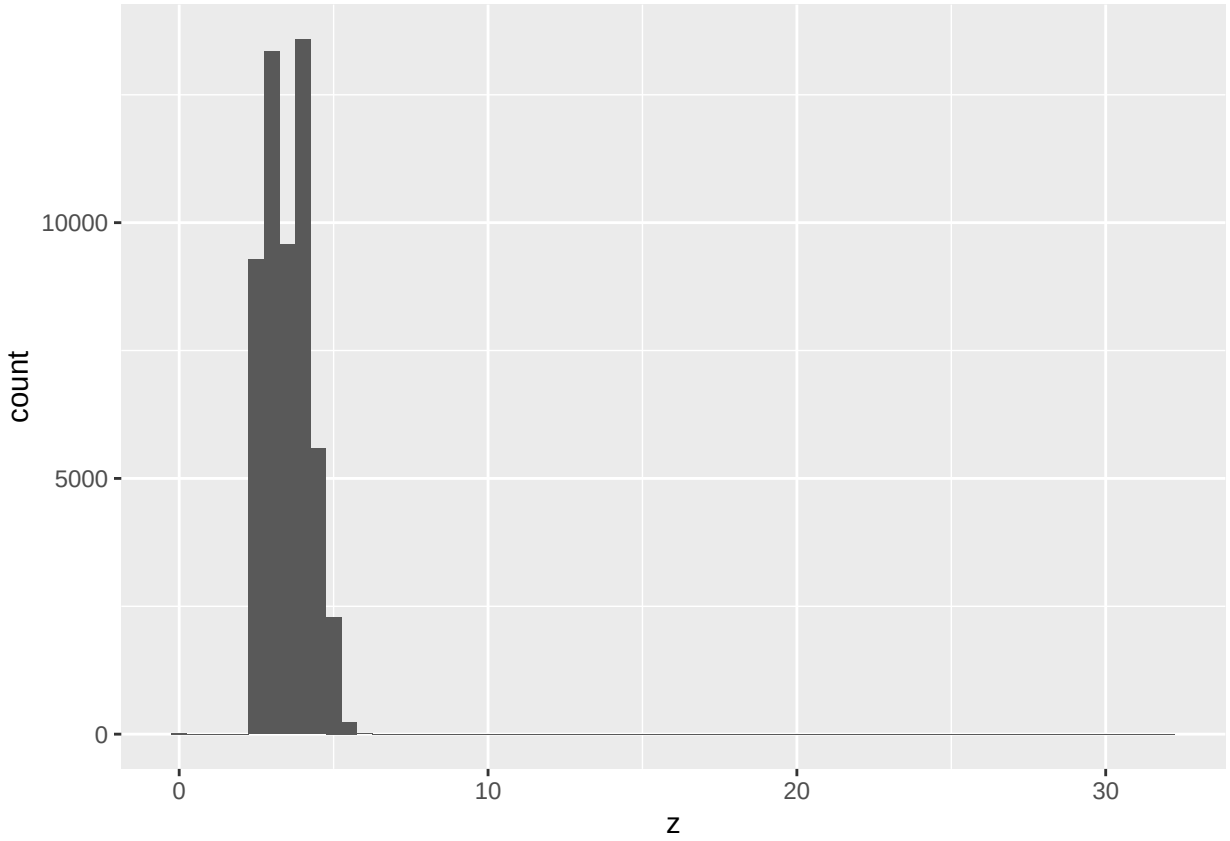
```
ggplot(data = diamonds) +
geom_histogram(aes(x), binwidth = 0.5)
```




```
ggplot(data = diamonds) +
geom_histogram(aes(y), binwidth = 0.5)
```




```
ggplot(data = diamonds) +
geom_histogram(aes(z), binwidth = 0.5)
```

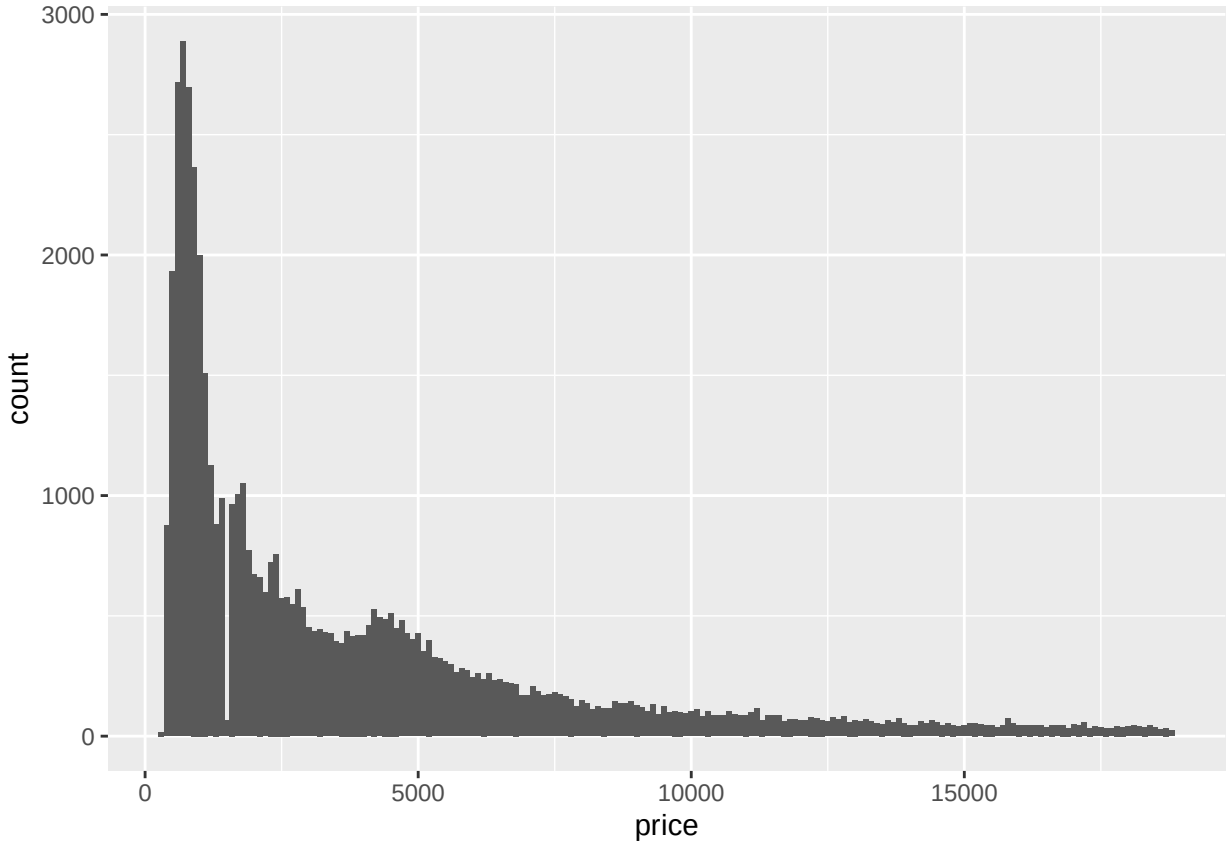



```
summary(diamonds[, c("x", "y", "z")])
```

```
##        x                y                z        
##  Min.   : 0.000   Min.   : 0.000   Min.   : 0.000  
##  1st Qu.: 4.710   1st Qu.: 4.720   1st Qu.: 2.910  
##  Median : 5.700   Median : 5.710   Median : 3.530  
##  Mean   : 5.731   Mean   : 5.735   Mean   : 3.539  
##  3rd Qu.: 6.540   3rd Qu.: 6.540   3rd Qu.: 4.040  
##  Max.   :10.740   Max.   :58.900   Max.   :31.800
```

Explore the distribution of price. Do you discover anything unusual or surprising?

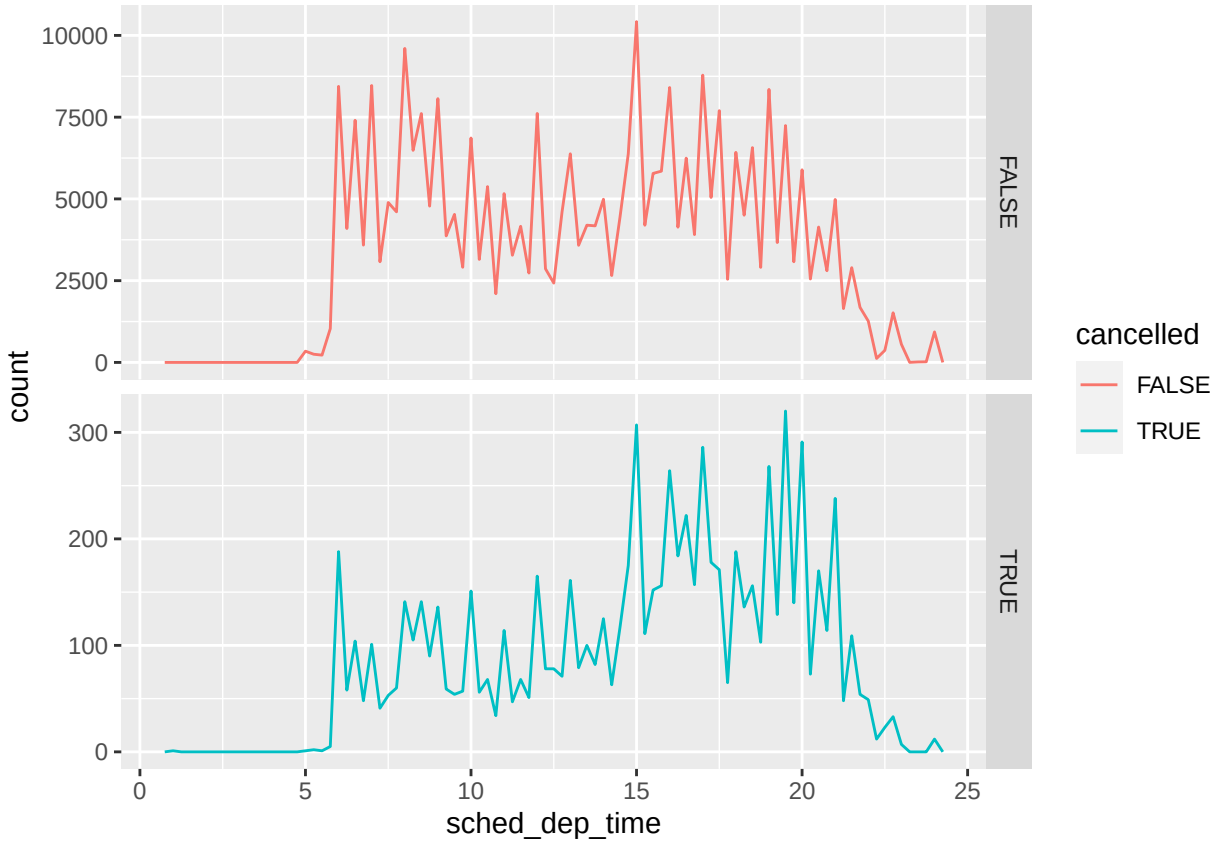
```
ggplot(data = diamonds) +
geom_histogram(aes(price), binwidth = 100)
```



Right skewed, peaks around 900-1000. Missing data for one range of values slightly above 1000.

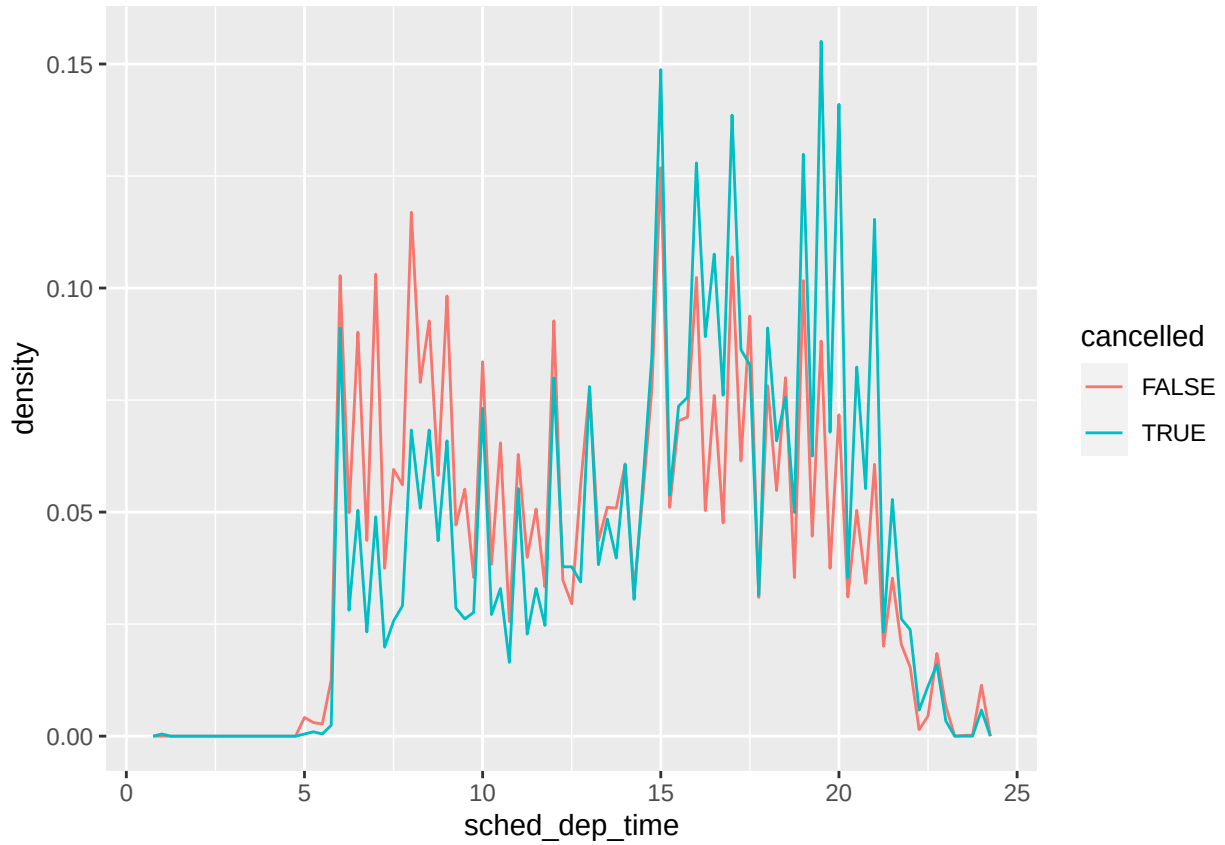
Q. EDA: Compare two distributions: Take the code which produces a plot with aes(x = sched_dep_time) using geom_freqpoly() in Section 11.4 of the textbook and modify it by faceting by the cancelled variable. The default value of facet_wrap(scales) is fixed. Read the help file for scales. What alternative value mitigates the effect of there being more non-cancelled flights than cancelled flights. What is one situation when you would prefer your alternative? What is one situation when you would prefer the default value?

```
nycflights13::flights %>%
mutate(
cancelled = is.na(dep_time),
sched_hour = sched_dep_time %/% 100,
sched_min = sched_dep_time %% 100,
sched_dep_time = sched_hour + sched_min / 60
) %>%
ggplot(mapping = aes(sched_dep_time)) +
geom_freqpoly(mapping = aes(colour = cancelled), binwidth = 1/4) +
facet_grid(rows = "cancelled", scales = "free_y")
```



Now, instead of using facet, make a single panel plot which compares the distribution of departure times of cancelled vs. non-cancelled flights using a tool from Section 11.5

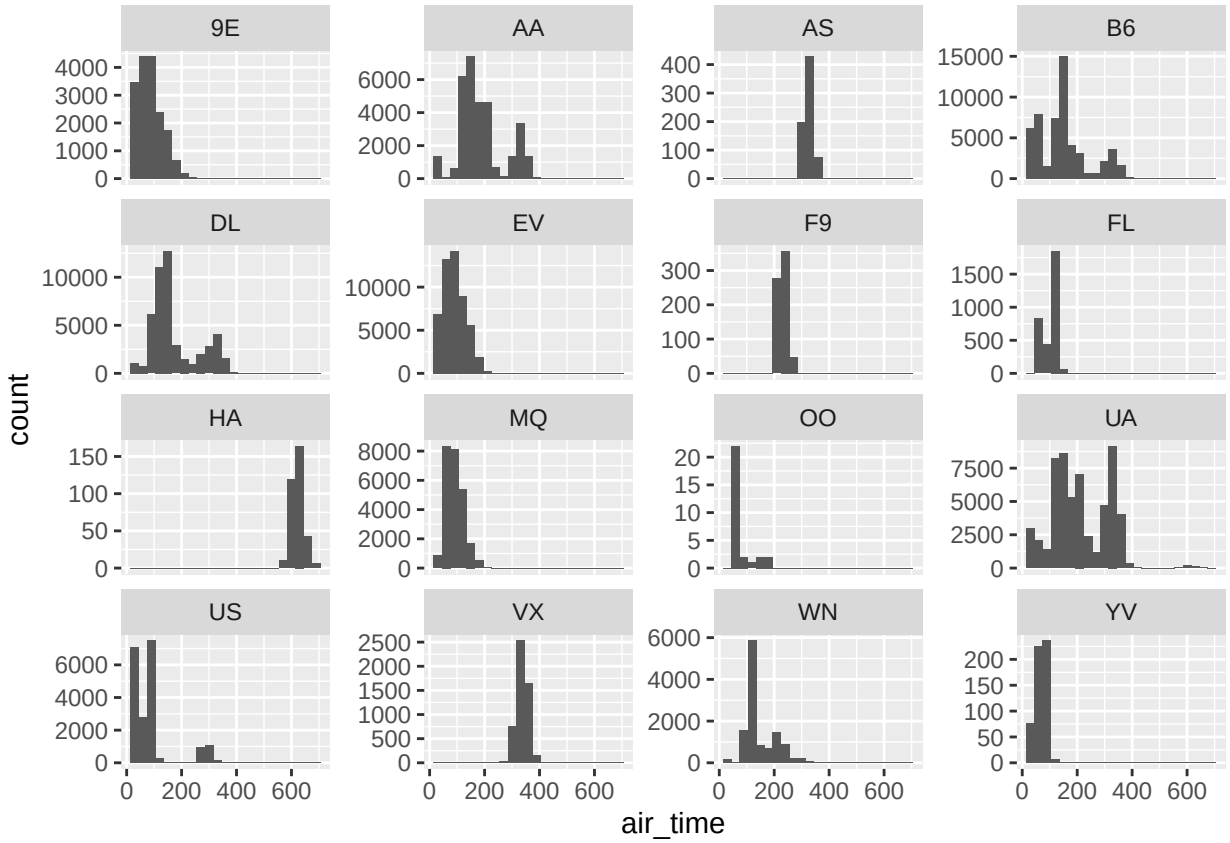
```
nycflights13::flights %>%
mutate(
cancelled = is.na(dep_time),
sched_hour = sched_dep_time %/% 100,
sched_min = sched_dep_time %% 100,
sched_dep_time = sched_hour + sched_min / 60
) %>%
ggplot(mapping = aes(x = sched_dep_time, y = after_stat(density))) +
geom_freqpoly(mapping = aes(colour = cancelled), binwidth = 1/4)
```



Q. Take a continuous variable and a categorical variable from nycflights13 and make three ggplots. Use a faceted geom_histogram(), a colored geom_freqpoly(), and geom_violin(). What are the pros and cons of each geom for plotting the combination of a continuous variable and a categorical variable?

```
nycflights13::flights %>%
ggplot(mapping = aes(air_time)) +
geom_histogram(binwidth = 30) +
facet_wrap(~carrier, nrow = 4, ncol = 4, scales = "free_y")
```

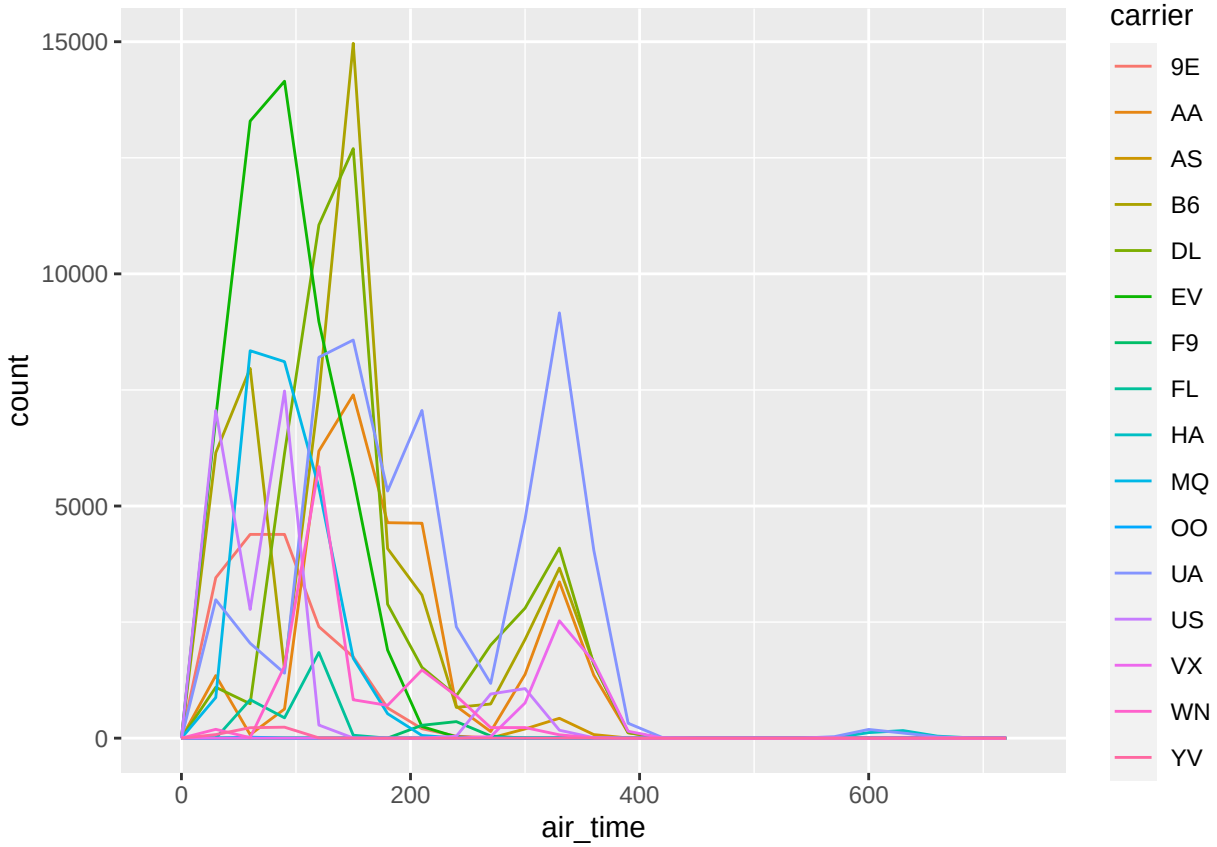
```
## Warning: Removed 9430 rows containing non-finite values (‘stat_bin()‘).
```



A geom_freqpoly makes it quite easy to see each air_time and compare across them. We can quite easily see that United Airlines runs the most 300-350 min flights. A con is that this makes it hard to see smaller airlines, for example Hawaiian Airlines is almost impossible to see even though they dominate the 10 hour flight segment because in absolute terms the flights that are that long are quote rare.

```
nycflights13::flights %>%
ggplot(mapping = aes(air_time)) +
geom_freqpoly(mapping = aes(colour = carrier), binwidth = 30)
```

```
## Warning: Removed 9430 rows containing non-finite values (‘stat_bin()‘).
```

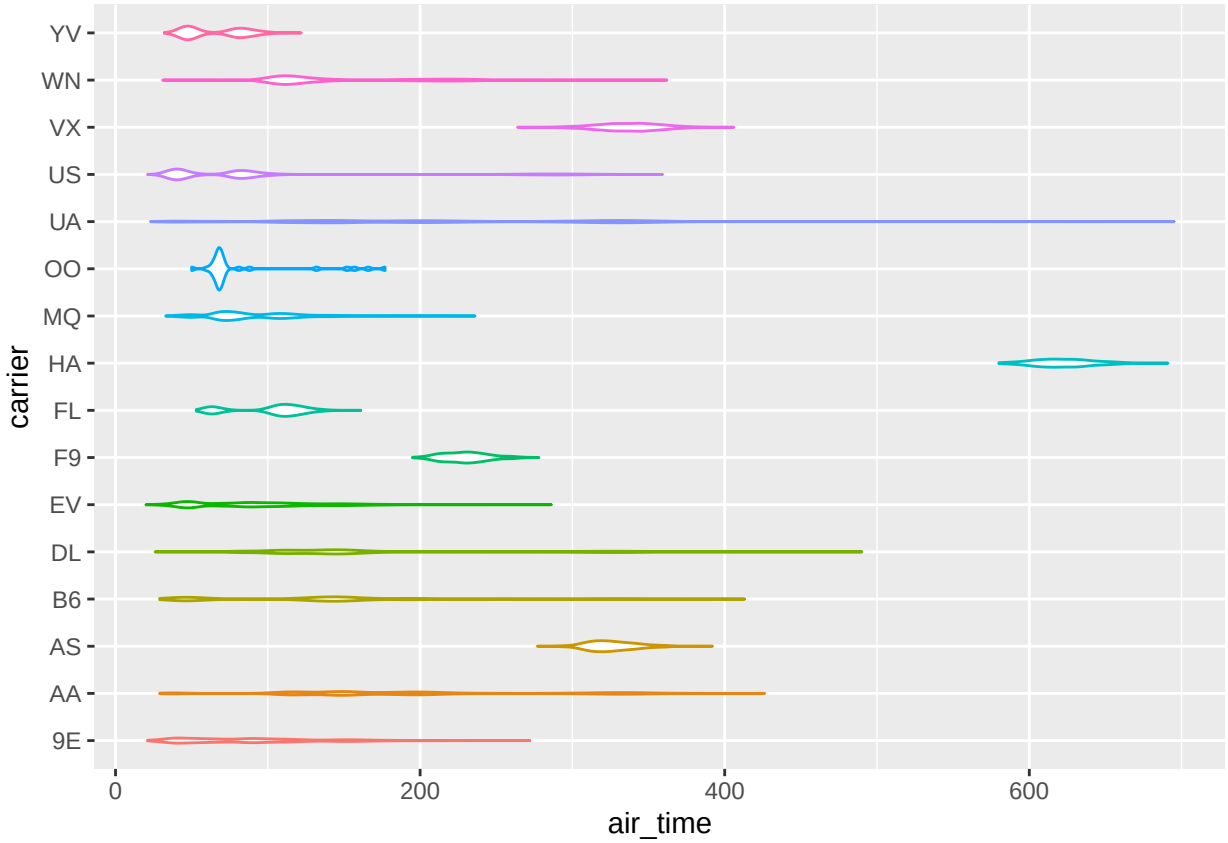


A violin plot makes it quite easy to see ranges of operations here, we can easily say that United Airlines runs flights all across the air_time spectrum with short, medium, and long haul flights, all being represented, while something like Frontier focusses primarily on mid-haul flights. It makes it hard to compare airlines, and to see any sort of absolute numbers about how many flights an airline runs.

```
nycflights13::flights %>%
ggplot(mapping = aes(air_time, carrier, color = carrier)) +
geom_violin(binwidth = 30, show.legend = F)
```

```
## Warning in geom_violin(binwidth = 30, show.legend = F): Ignoring unknown
## parameters: `binwidth`
```

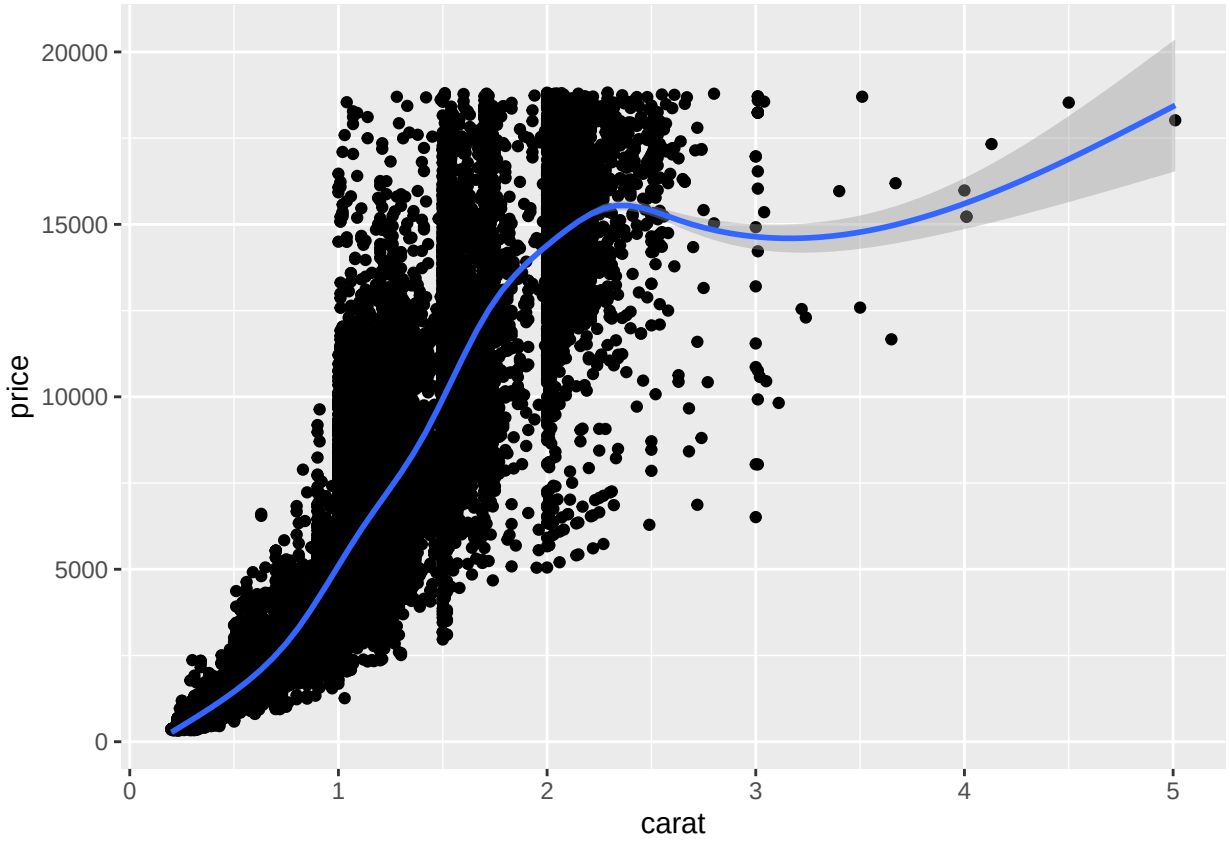
```
## Warning: Removed 9430 rows containing non-finite values (`stat_ydensity()`).
```



4. How does the price distribution of very large diamonds (as measured by carat) compare to small diamonds? Is it as you expect, or does it surprise you? Why?

```
ggplot(data = diamonds, mapping = aes(x = carat, y = price)) +
geom_point() +
geom_smooth()
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



It is what I would expect, price increases with carat size. Low variation in the low carat sizes, higher variation in the higher end. Our data seems capped at roughly 18,000 USD meaning prices probably not reliable for higher carat sizes

Q. EDA - Covariation - Use geom_tile() together with dplyr to explore how average flight delays vary by destination and month of year. What makes the plot difficult to read? How could you improve it?

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.1.2
```

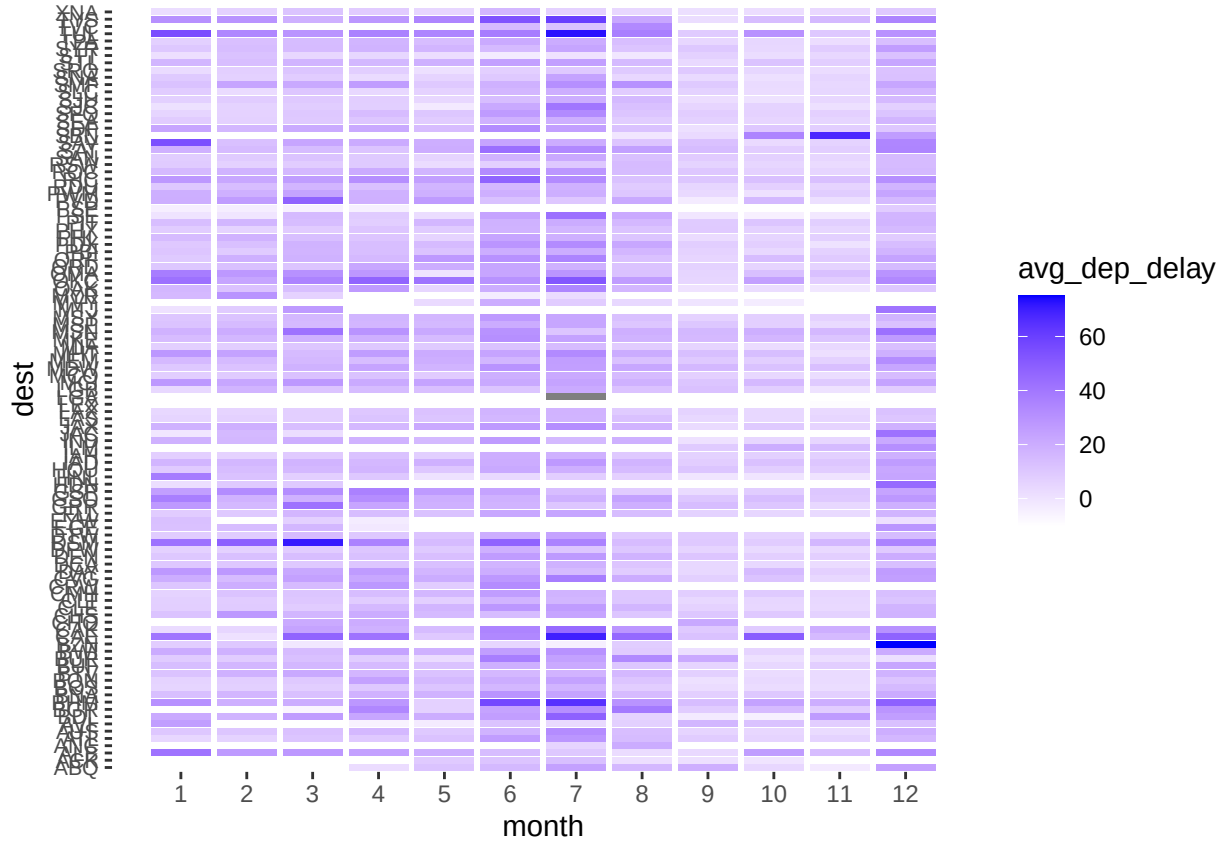
```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
nycflights13::flights %>%
mutate(month = month(time_hour)) %>%
group_by(dest, month) %>%
summarise(avg_dep_delay = mean(dep_delay, na.rm = T)) %>%
ggplot() +
geom_tile(aes(month, dest, fill = avg_dep_delay), height = 0.9, width = 0.9) +
scale_fill_gradient(low = "white", high = "blue") +
scale_x_continuous(breaks = 1:12) +
```

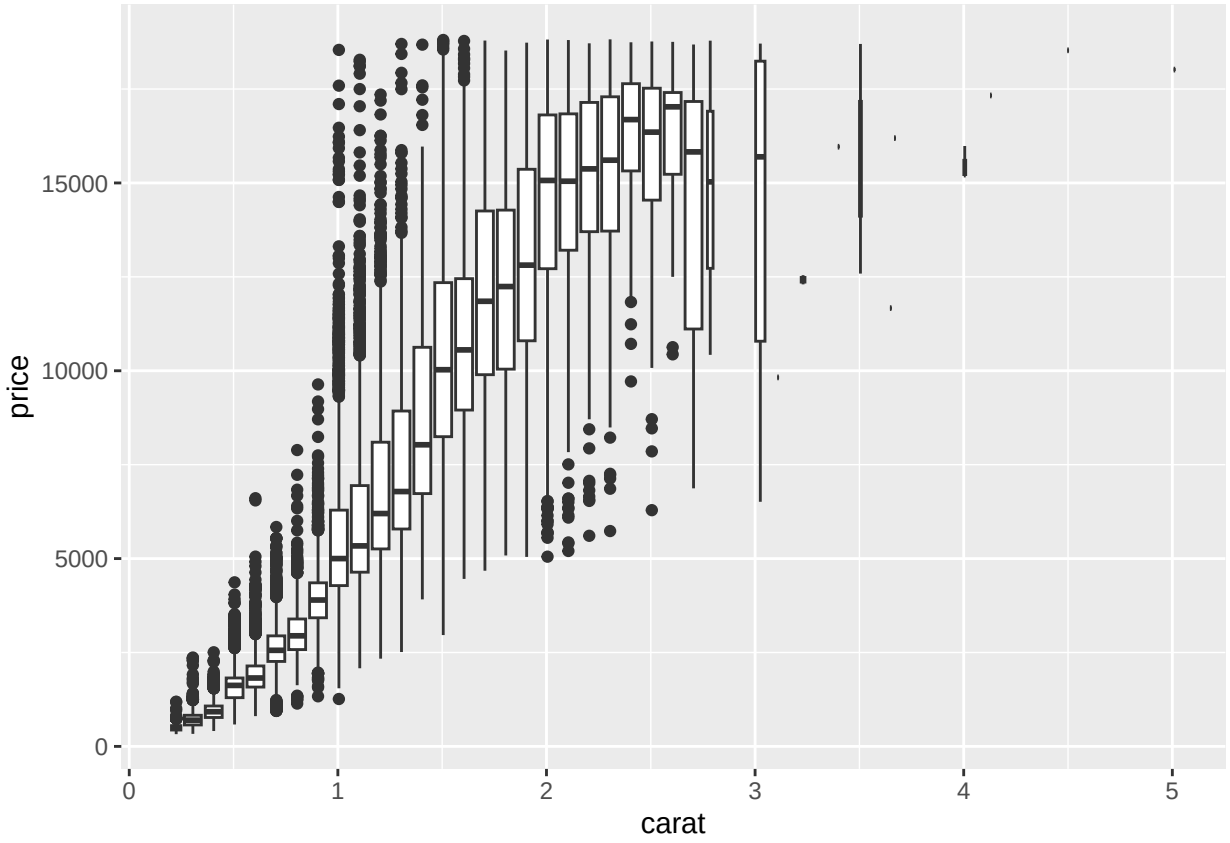
```
theme(
axis.text.y = element_text(size = 7.5),
panel.spacing = element_blank(),
panel.grid = element_blank(),
panel.background = element_blank()
)
```

```
## `summarise()` has grouped output by 'dest'. You can override using the
## `.groups` argument.
```

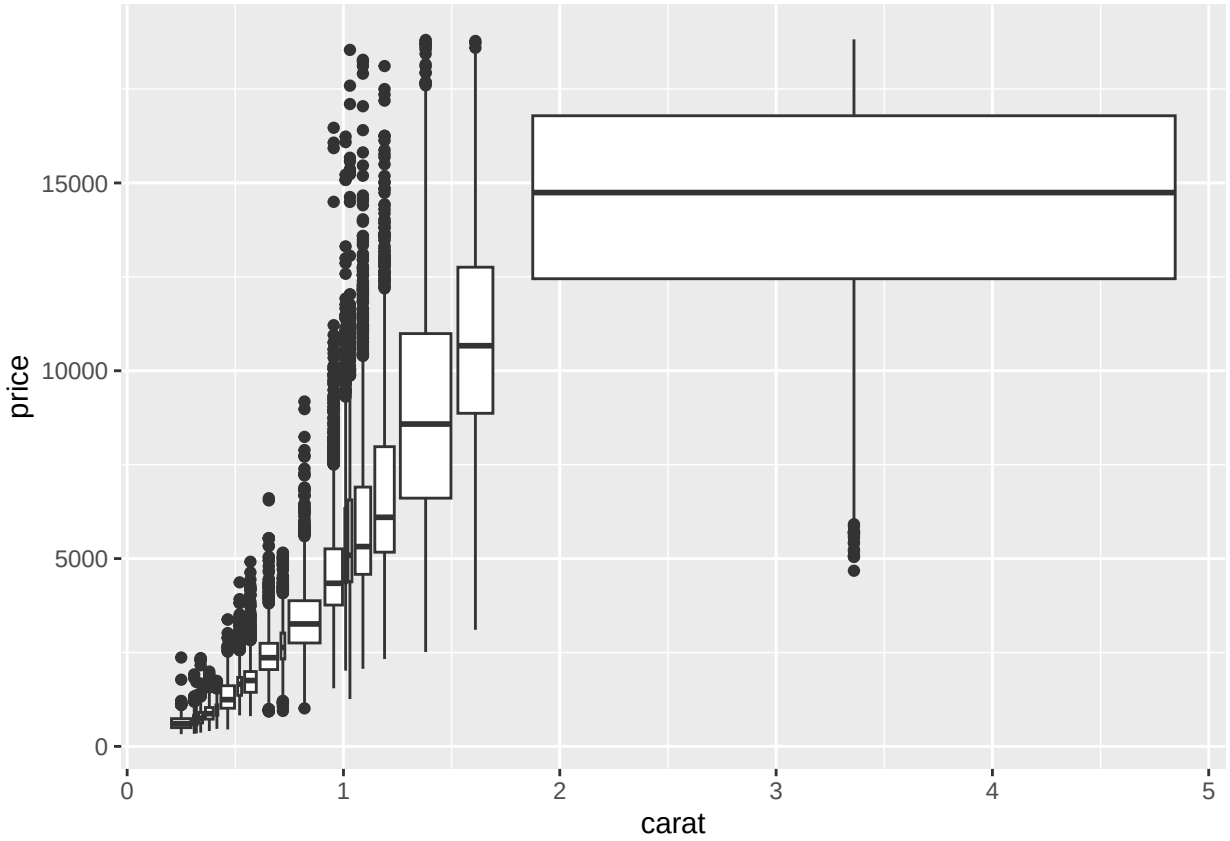


Q. Refer to a diamonds box plot using cut_width(). Instead, make a plot with ggplot(diamonds, aes(x = carat, y = price, group = cut_number(carat, 20))) + geom_boxplot(). What are the advantages and disadvantages of this approach as compared to cut_number()?

```
ggplot(diamonds, aes(x = carat, y = price, group = cut_width(carat, 0.1))) +
geom_boxplot()
```




```
ggplot(diamonds, aes(x = carat, y = price, group = cut_number(carat, 20))) + geom_boxplot()
```



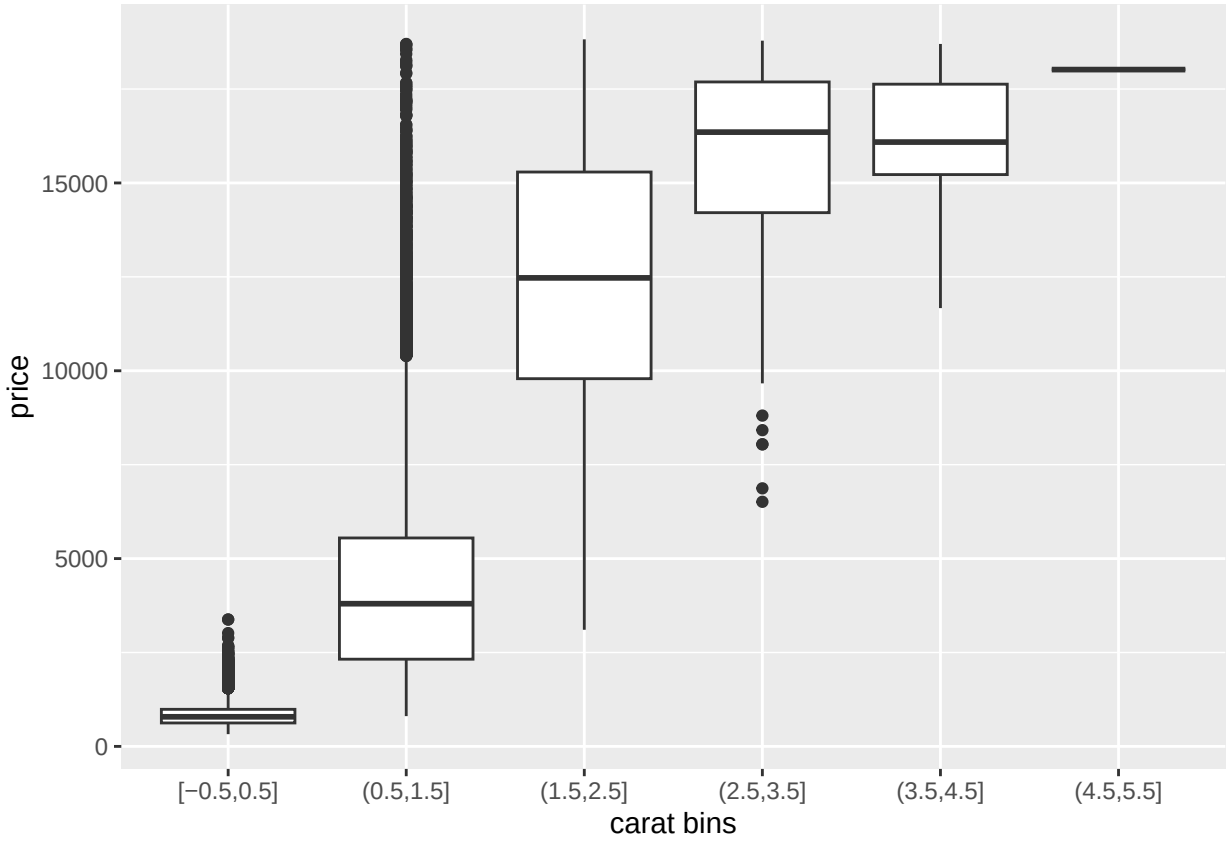
While the cut_width approach focusses on the range of values, cut_number focusses on how many observations per box. The advantage of cut_number is that it saves us from making misleading claims due to small sample size effects, as each box has the same number of observations, they can all be compared with some parity. A disadvantage is that it loses some of the granularity of the cut_width approach and the unevenly sized boxes make it harder to have quick takeaways.

Q. Looking at the table below it appears that fair is nearly the highest price cut of diamond and ideal the worst cut. But there is an omitted variable problem!

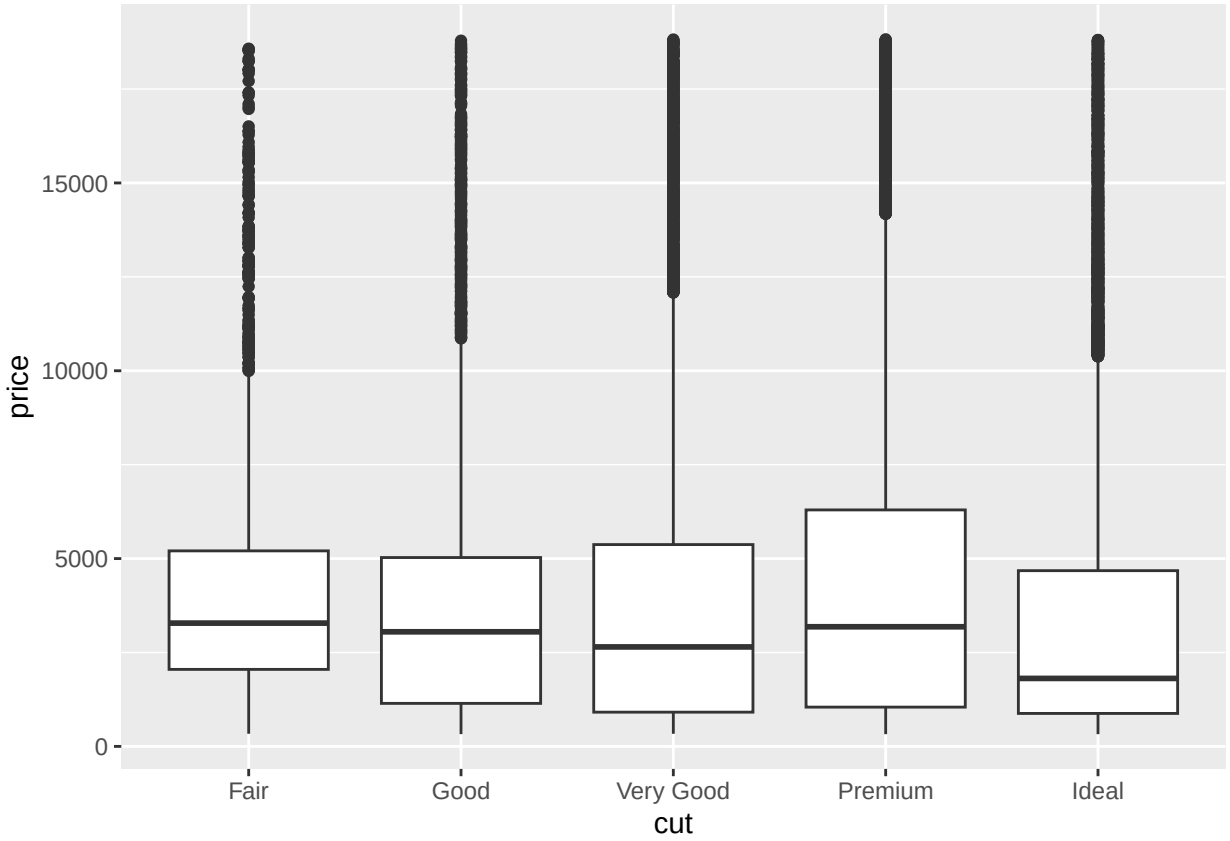
Fair: mean price = 4359 Good: mean price = 3929 Very Good : mean price = 3982 Premium : mean price = 4584 Ideal: mean price = 3458

What variable in the diamonds dataset is most important for predicting the price of a diamond? How is that variable correlated with cut? Explain why the table above is misleading.

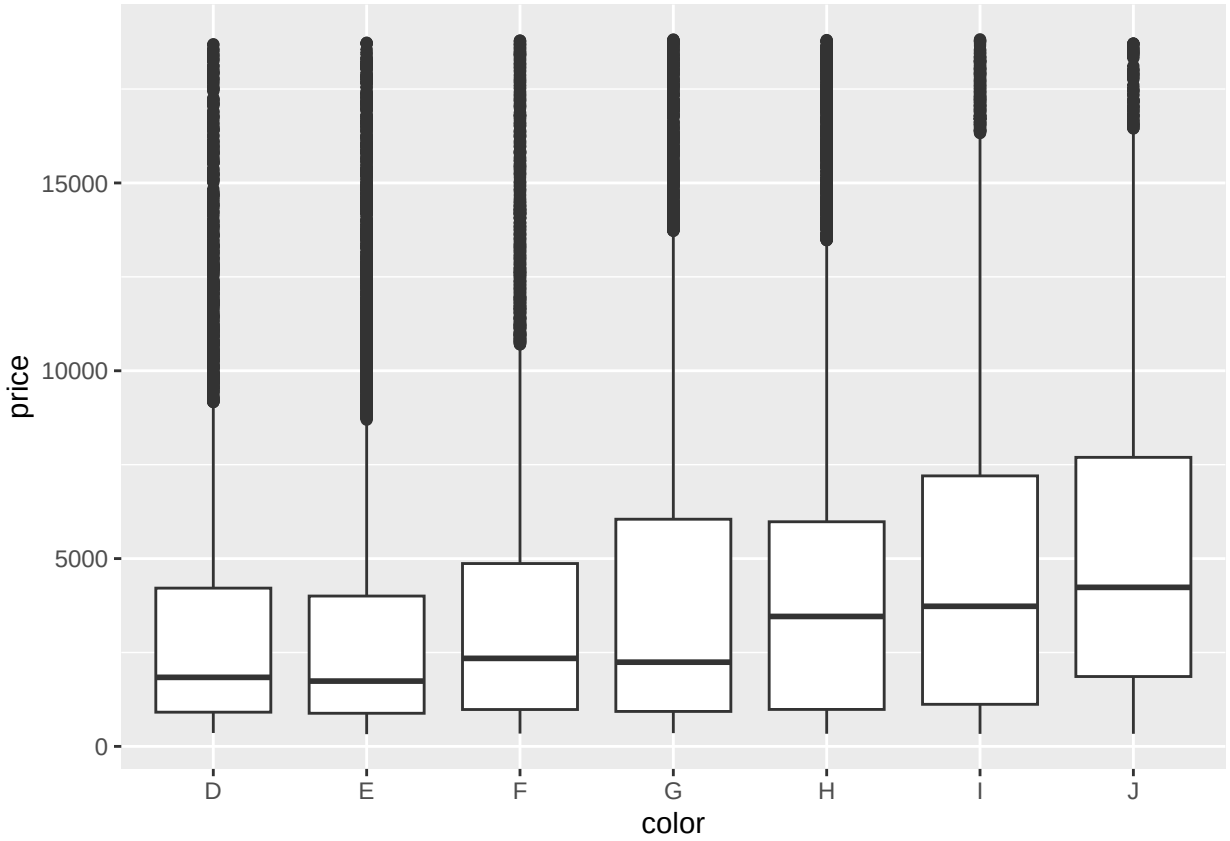
```
diamonds %>%
ggplot(aes(x = cut_width(carat, 1), y = price)) + geom_boxplot() + labs(x = "carat bins")
```



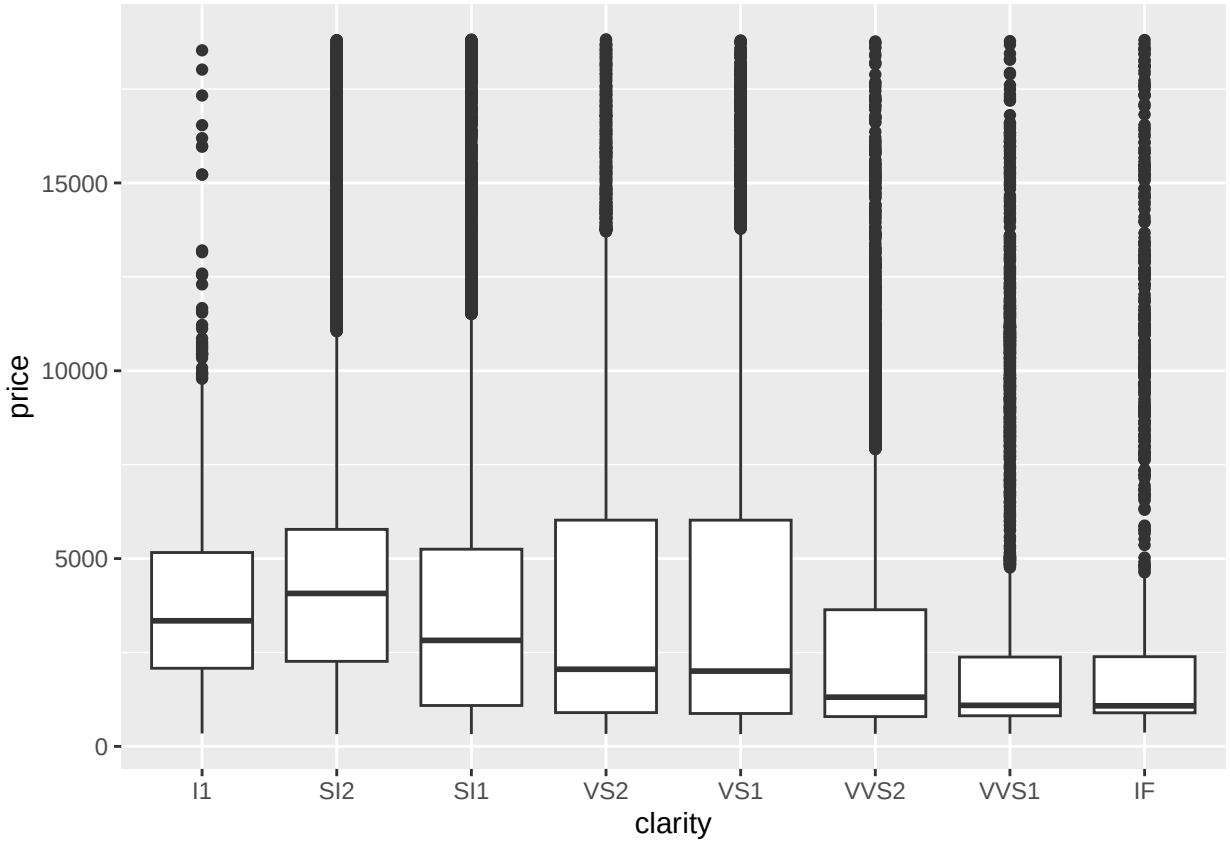

```
diamonds %>%
ggplot(aes(x = cut, y = price)) + geom_boxplot()
```



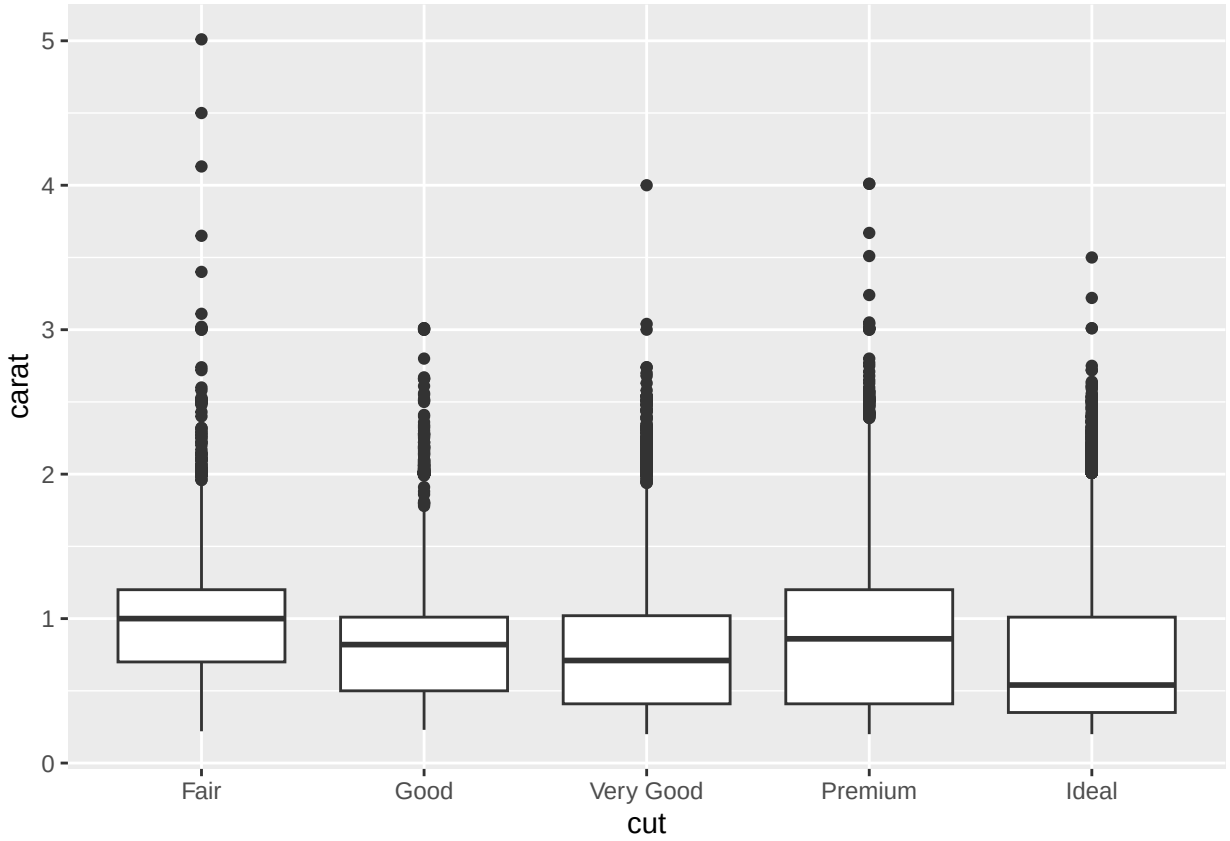

```
diamonds %>%
ggplot(aes(x = color, y = price)) + geom_boxplot()
```




```
diamonds %>%
ggplot(aes(x = clarity, y = price)) + geom_boxplot()
```




```
diamonds %>%
ggplot(aes(x = cut, y = carat)) + geom_boxplot()
```



Answer: This contradiction is explained by carat size, which is the biggest driver of prices. The box plot above mirrors the table in terms of price movements, with Fair and Premium diamonds having the highest average prices and being the largest at various points in the distribution. While Ideal are the best cut, they are generally smaller diamonds.

Q. Take diamonds %>% count(color, cut) and add a column to more clearly show the distribution of cut within color. Which cut is most common in every color category? Again take take diamonds %>% count(color, cut) but now add a column to show distribution of color within cut. Using the dataframe you just produced as input, reproduce the following graph.

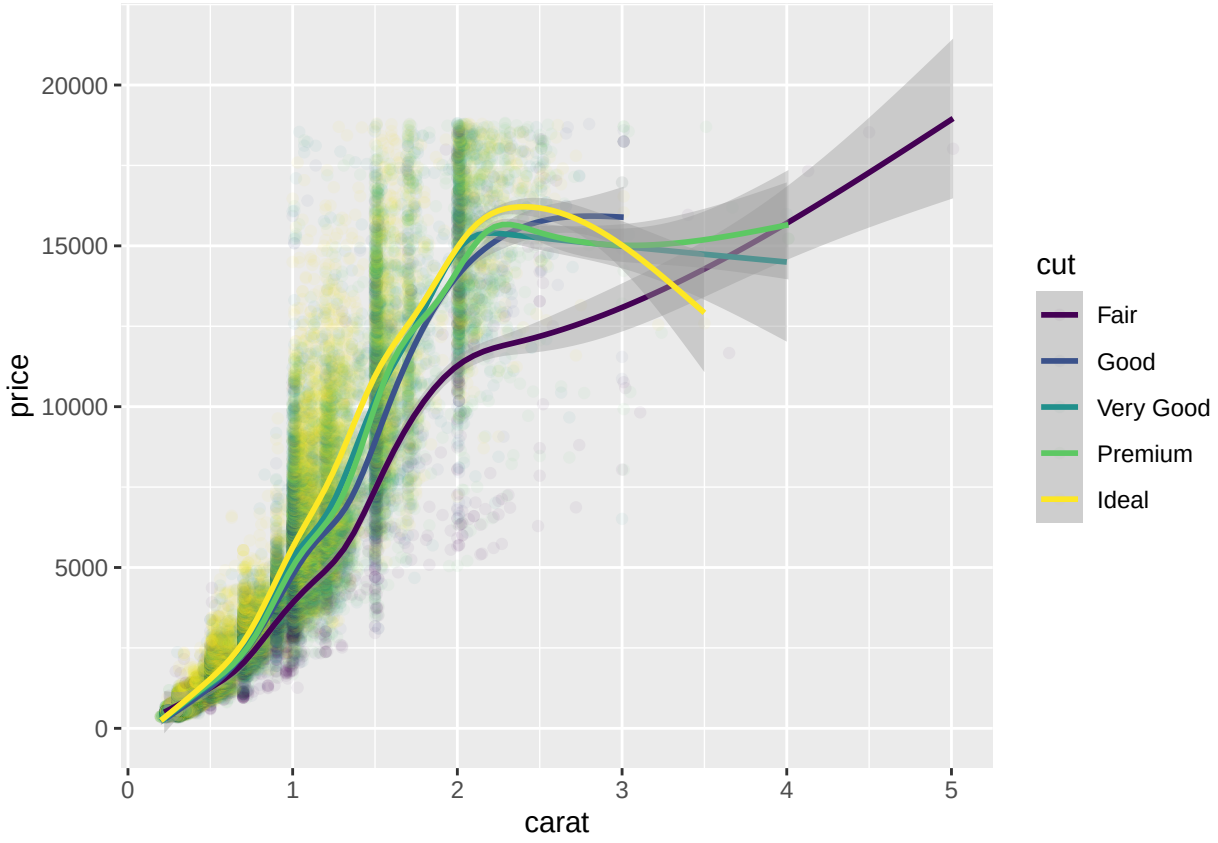
```
cut_within_color <-
diamonds %>%
count(color, cut) %>% group_by(color) %>%
mutate(prop = round(n / sum(n), 3))
cut_within_color
```

```
## # A tibble: 35 x 4
## # Groups:   color [7]
##    color cut           n  prop
##    <ord> <ord>     <int> <dbl>
##  1 D     Fair        163 0.024
##  2 D     Good        662 0.098
##  3 D     Very Good  1513 0.223
##  4 D     Premium    1603 0.237
##  5 D     Ideal      2834 0.418
##  6 E     Fair        224 0.023
##  7 E     Good        933 0.095
```

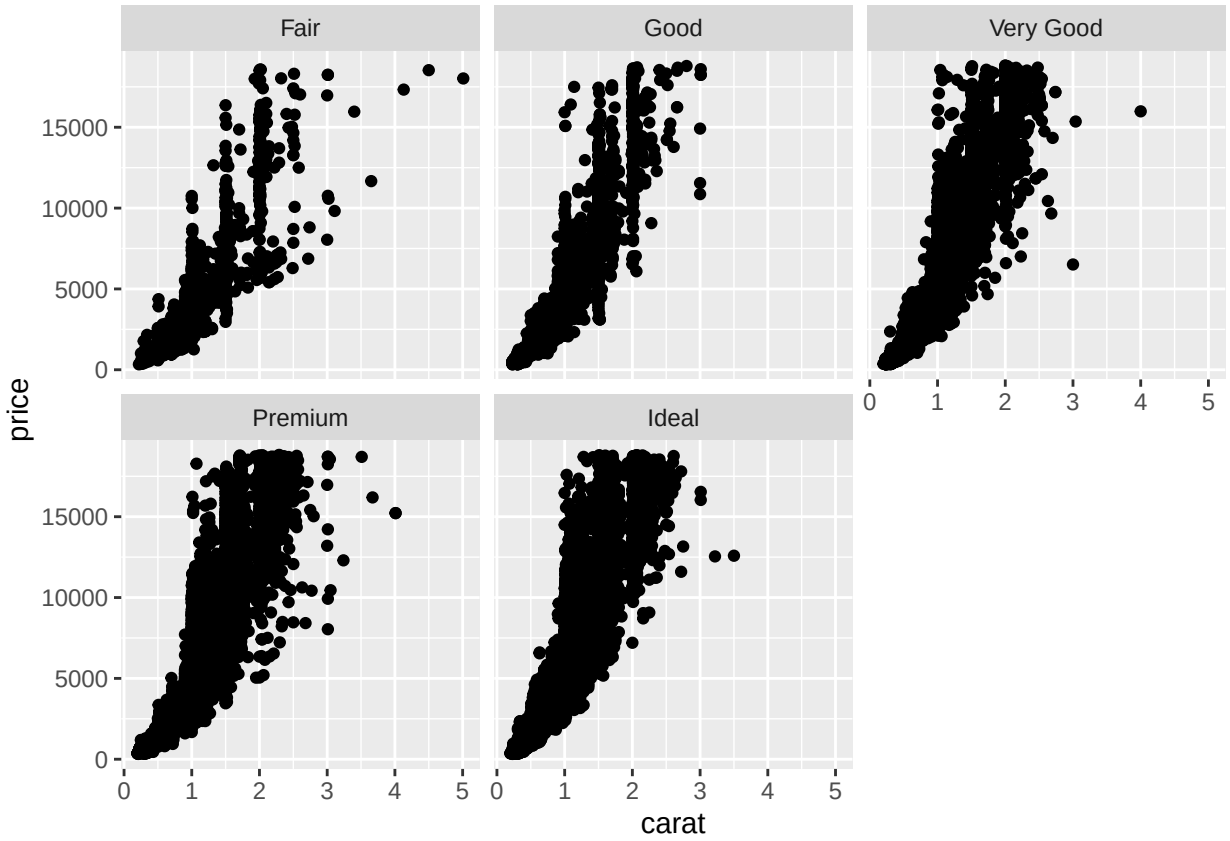
```
##  8 E     Very Good  2400 0.245
##  9 E     Premium    2337 0.239
## 10 E     Ideal      3903 0.398
## # i 25 more rows
```

```
color_within_cut <-
diamonds %>%
count(color, cut) %>% group_by(cut) %>%
mutate(prop = round(n / sum(n), 3))

ggplot(data = diamonds, aes(carat, price, color = cut)) + geom_point(alpha = 0.05) +
geom_smooth()
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```




```
ggplot(data = diamonds) + geom_point(aes(carat, price)) + facet_wrap(~cut)
```


Fair
Good
Very Good
Premium
Ideal
price
carat
0
5000
10000
15000
0
1
2
3
4
5