STT 301: More graphics with ggplot2

Shawn Santo

October 8, 2018

Introduction

Learning objectives:

- `ggplot` function
  - `geom_bar`
  - `geom_histogram`
  - `geom_boxplot`
  - `geom_raster`
  - `geom_density`

The package `ggplot2` has many options and capabilities so you will probably find the following `ggplot2` help resources useful:

1. http://docs.ggplot2.org/current/ (http://docs.ggplot2.org/current/)
2. the graphics portion of the R cookbook site at http://www.cookbook-r.com/Graphs/ (http://www.cookbook-r.com/Graphs/)
3. http://zevross.com/blog/2014/08/04/beautiful-plotting-in-r-a-ggplot2-cheatsheet-3/ (http://zevross.com/blog/2014/08/04/beautiful-plotting-in-r-a-ggplot2-cheatsheet-3/)
4. RStudio “Data Visualization with ggplot2” cheatsheet
5. Google

Part 1

Bar plots - Diamonds data

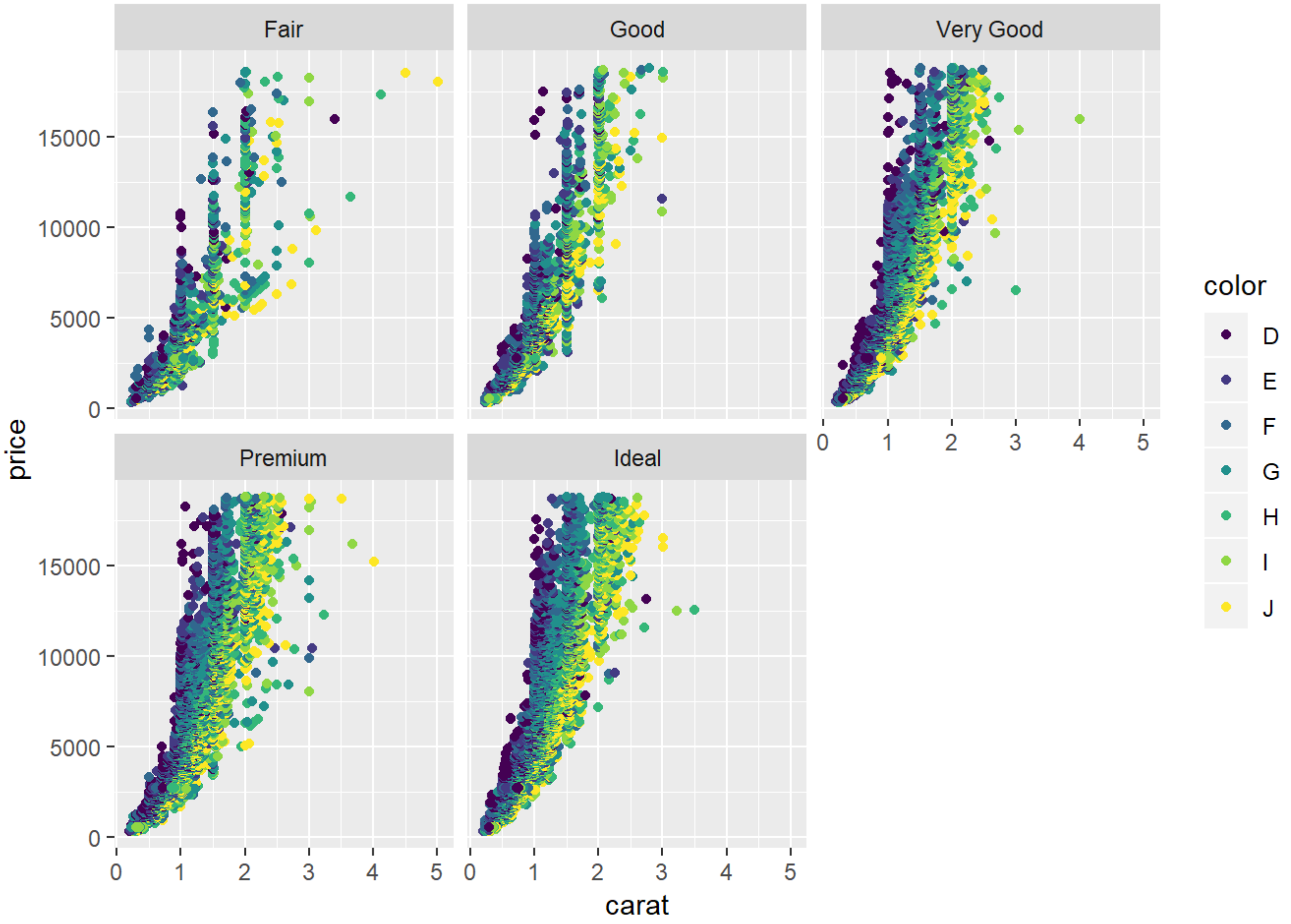
The ggplot2 package comes with a data set called `diamonds`. Let’s look at it below. To obtain further details type `?diamonds` in your console window.

```
library(ggplot2)
str(diamonds)
```

```
Classes 'tbl_df', 'tbl' and 'data.frame':   53940 obs. of  10 variables:
 $ carat  : num  0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23 ...
 $ cut    : Ord.factor w/ 5 levels "Fair"<"Good"<..: 5 4 2 4 2 3 3 3 1 3 ...
 $ color  : Ord.factor w/ 7 levels "D"<"E"<"F"<"G"<..: 2 2 2 6 7 7 6 5 2 5 ...
 $ clarity: Ord.factor w/ 8 levels "I1"<"SI2"<"SI1"<..: 2 3 5 4 2 6 7 3 4 5 ...
 $ depth  : num  61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...
 $ table  : num  55 61 65 58 58 57 57 55 61 61 ...
 $ price  : int  326 326 327 334 335 336 336 337 337 338 ...
 $ x      : num  3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...
 $ y      : num  3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...
 $ z      : num  2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ...
```

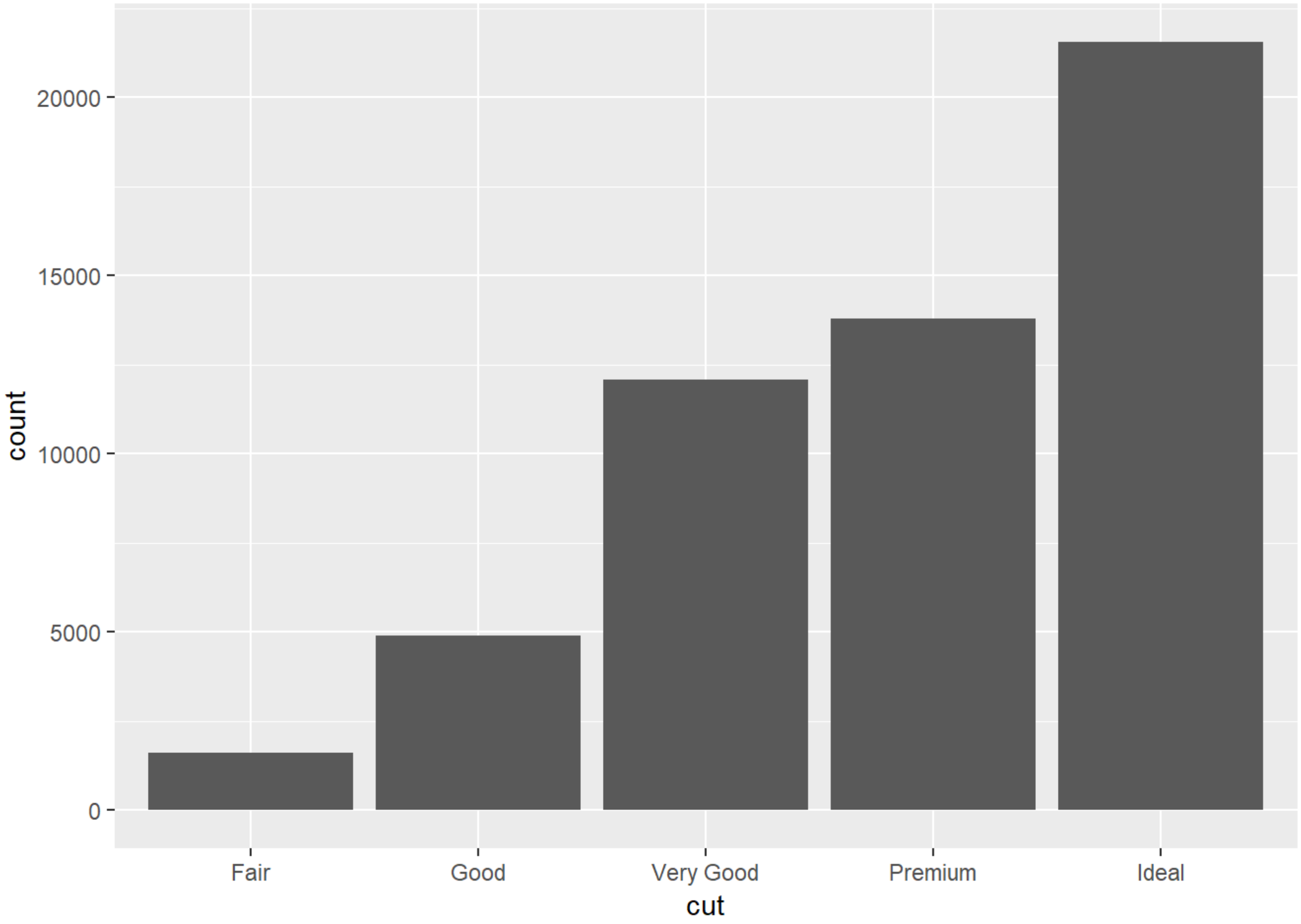
Last time you created scatter plots such as the one below by making use of `geom_point` .

```
diamond.plot <- ggplot(data=diamonds, aes(x=carat, y=price, colour = color))
diamond.plot + geom_point() + facet_wrap(~cut)
```

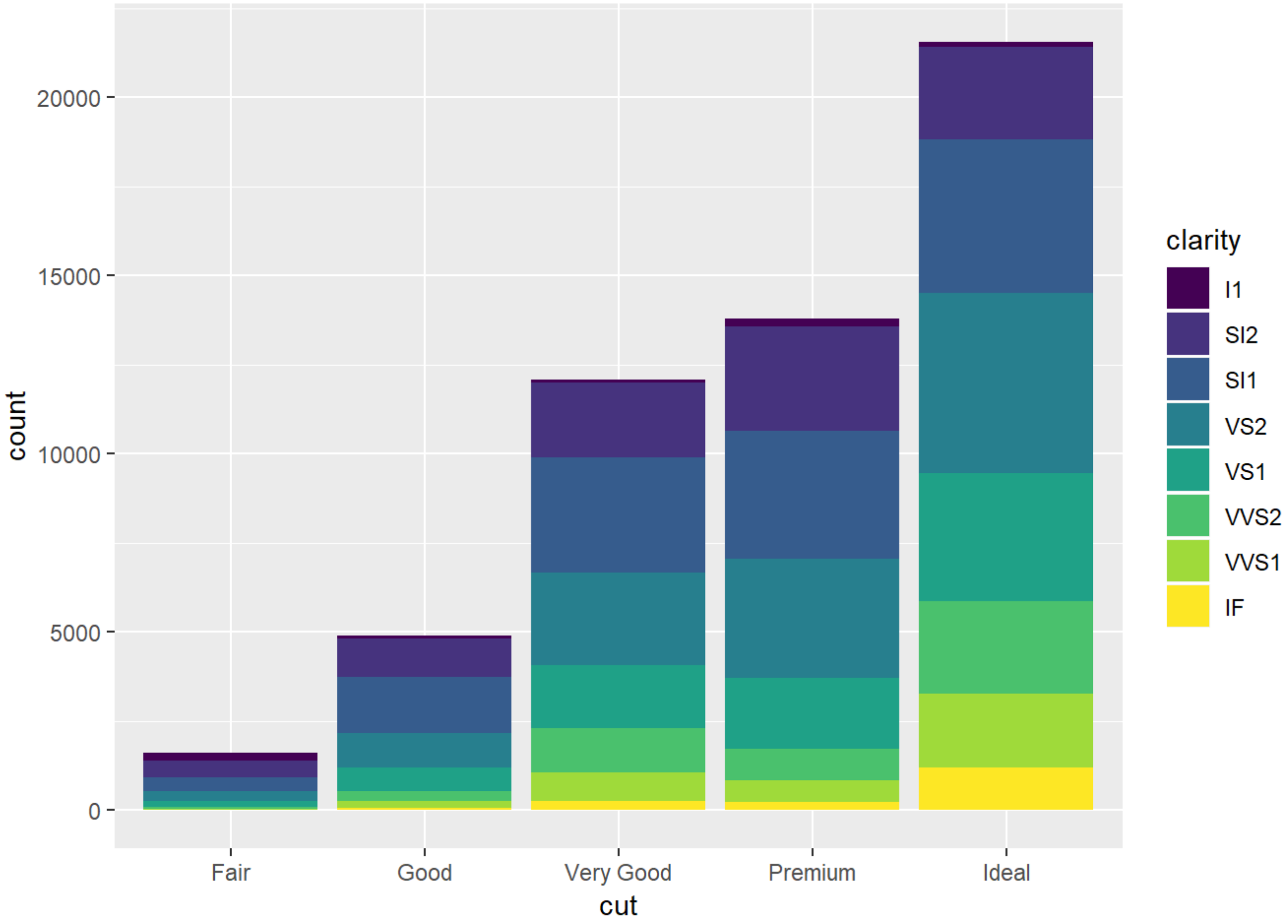


Use `geom_bar` to create the bar plots you see below.

Plot a


20000
15000
10000
5000
0
count
Fair
Good
Very Good
Premium
Ideal
cut

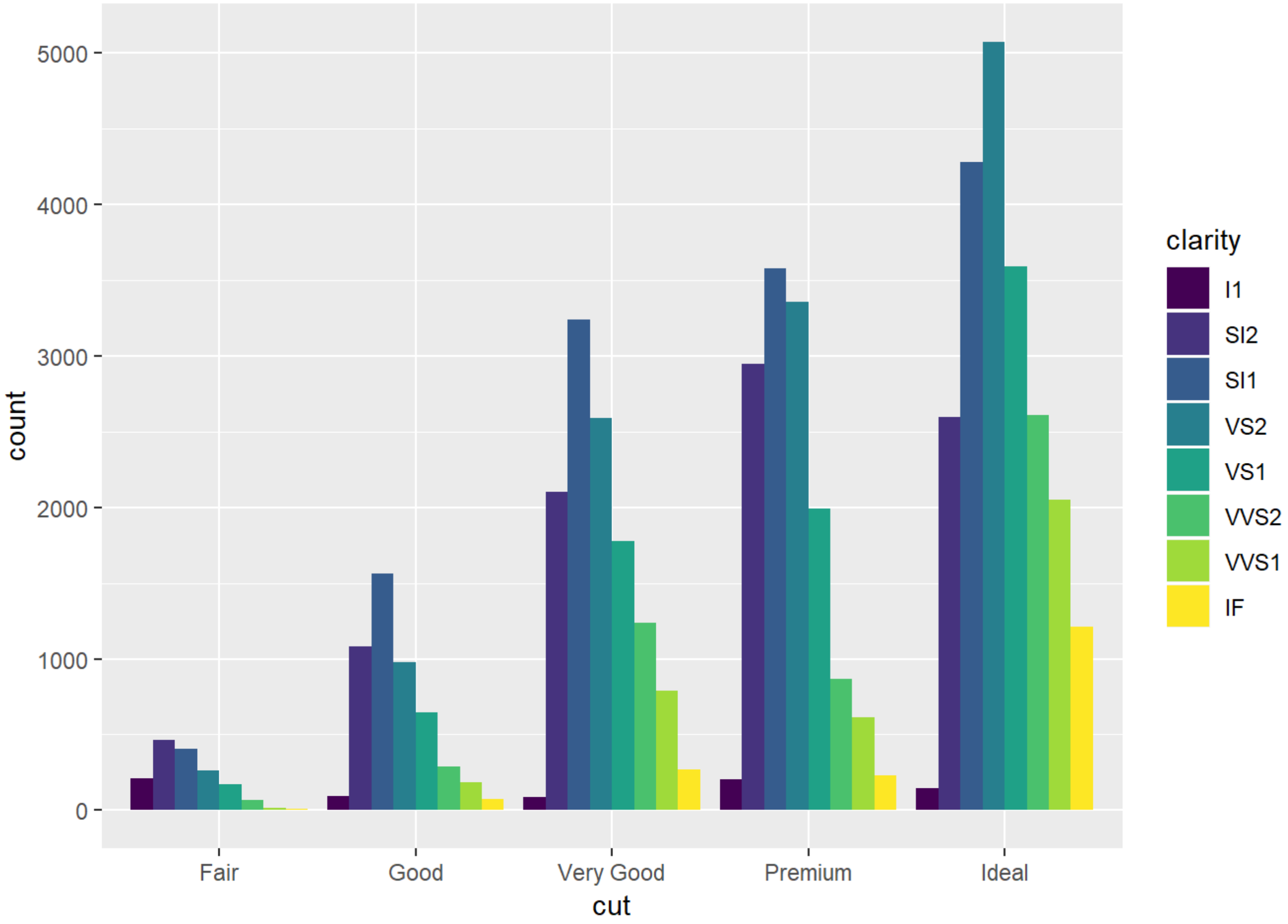
Plot b



Plot c

Details:

- specify the position argument as “dodge”



Part 2

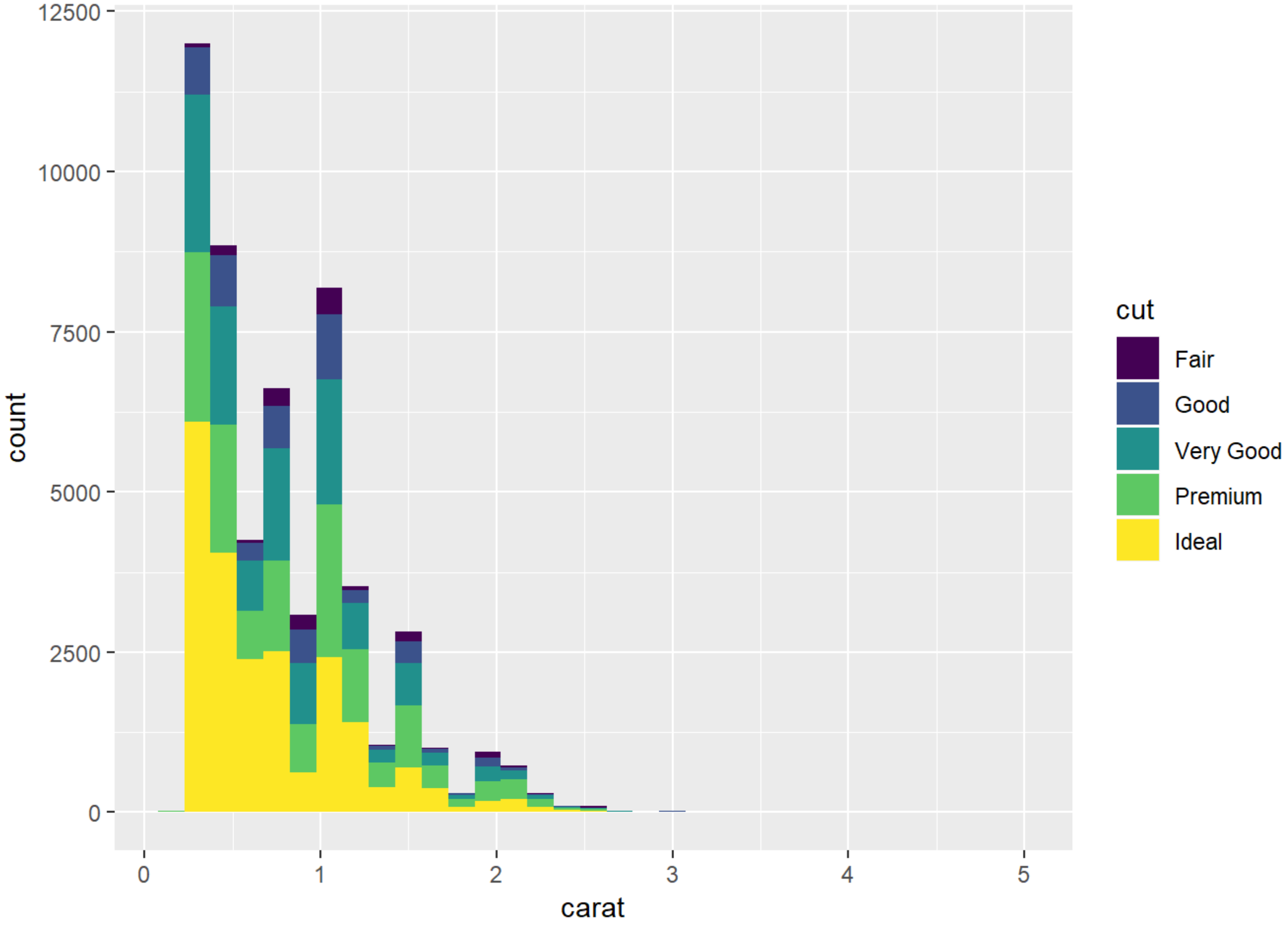
Histograms - Diamonds data

Use `geom_histogram` to create the histograms you see below.

Plot a

Details:

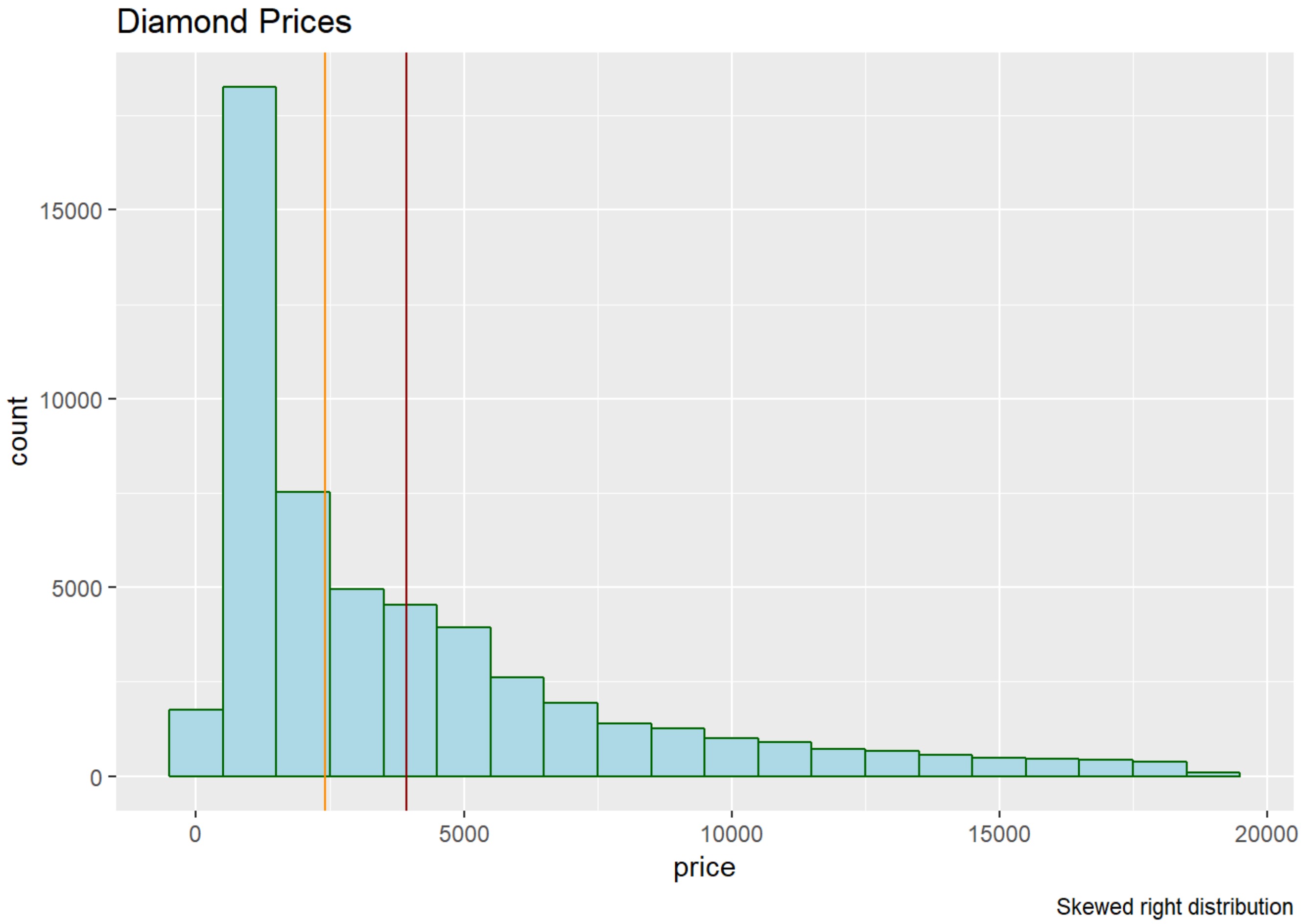
- binwidth used 0.15



Plot b

Details:

- binwidth used: 1000
- colors used: “lightblue”, “darkgreen”, “darkorange”, “darkred”
- red line represents the mean price
- orange line represents the median price
- create vertical lines with `geom_vline`



Part 3

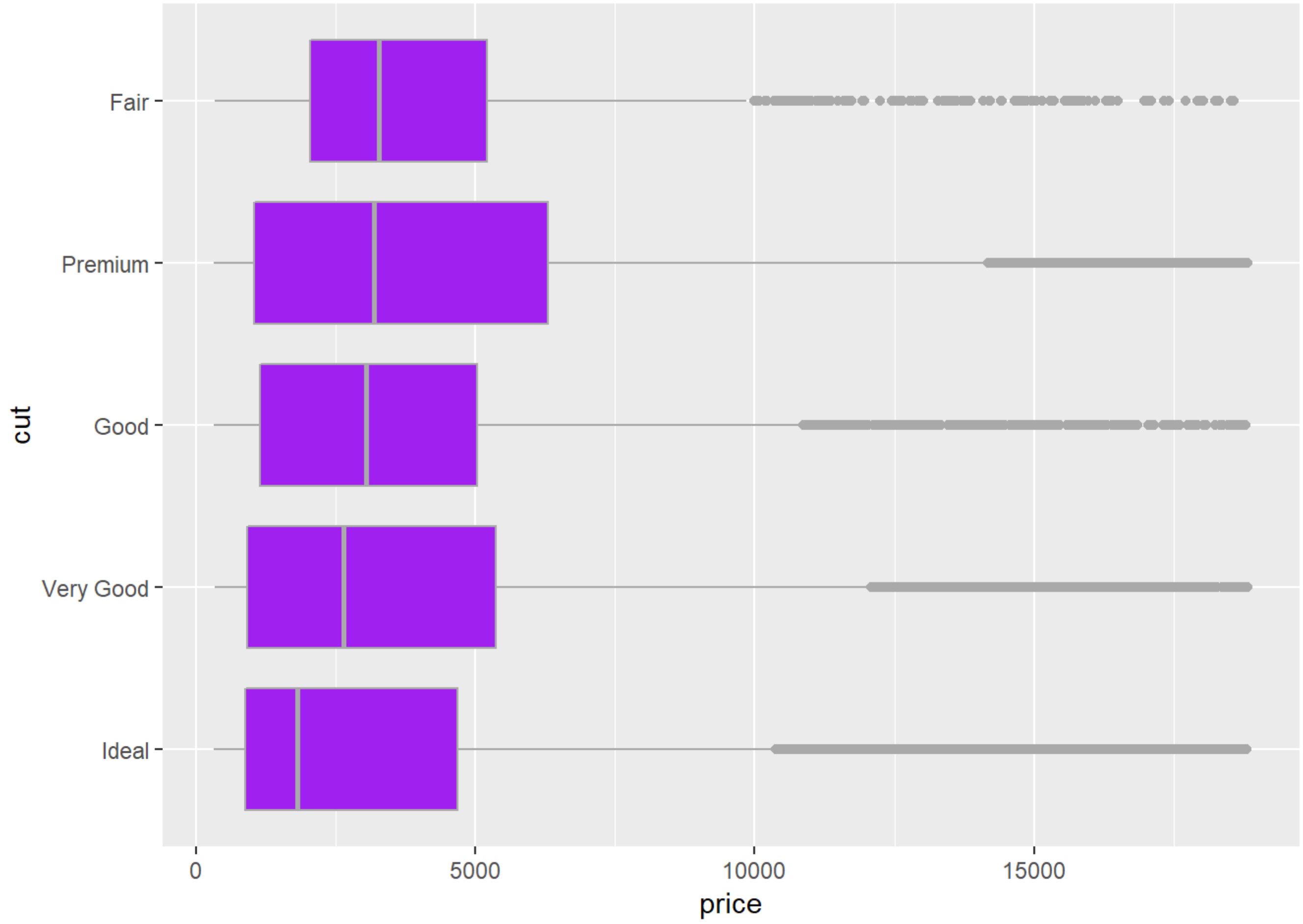
Box plots - Diamonds data

Use `geom_boxplot` to create box plots for the plots you see below.

Plot a

Details:

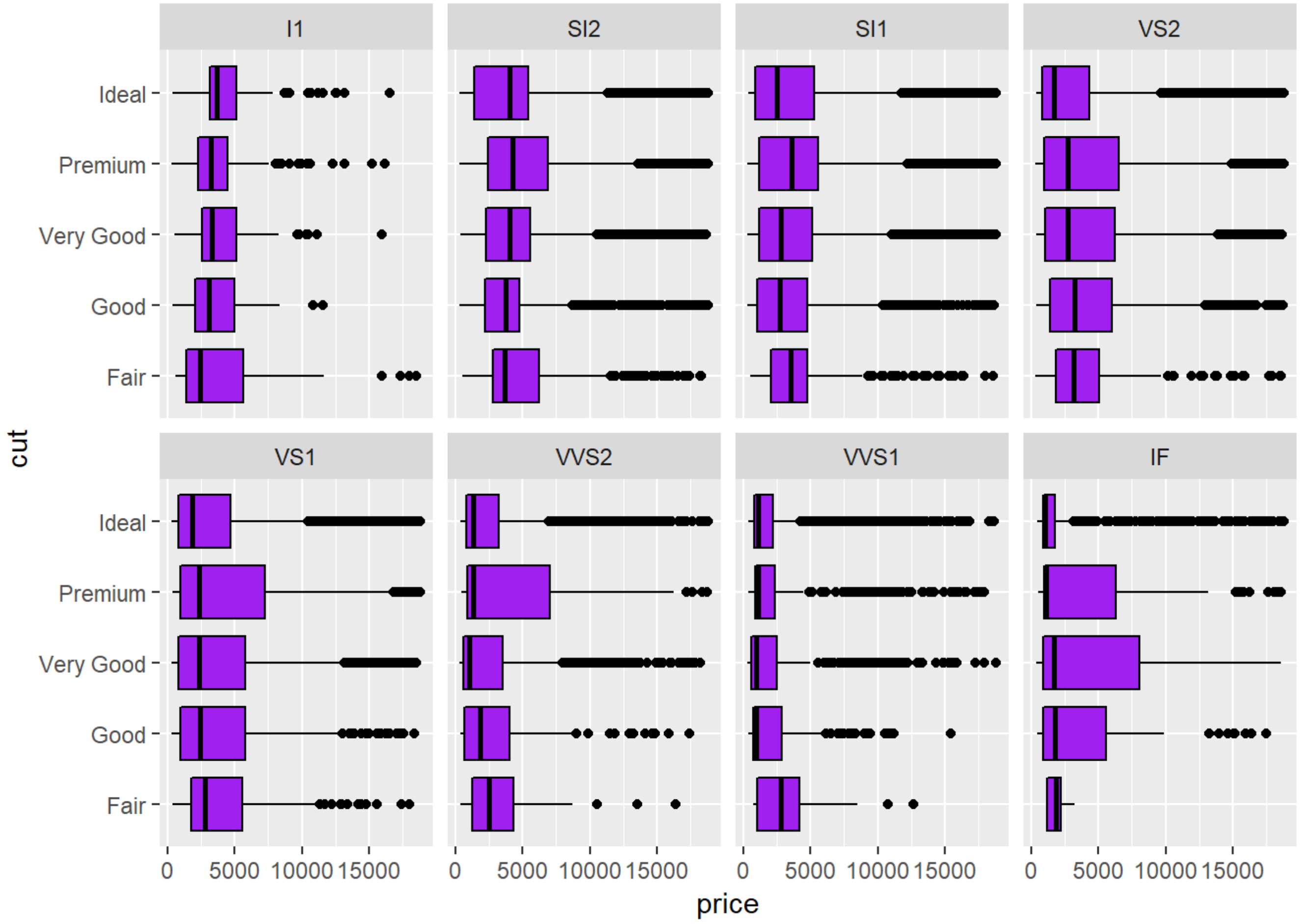
- colors used: “darkgrey”, “purple”
- use `x = reorder(cut, price, FUN = median)` to arrange box plots by median value
- use `coord_flip` for horizontal box plots



Plot b

Details:

- colors used: “black”, “purple”



Part 4

Heat maps - Diamonds data

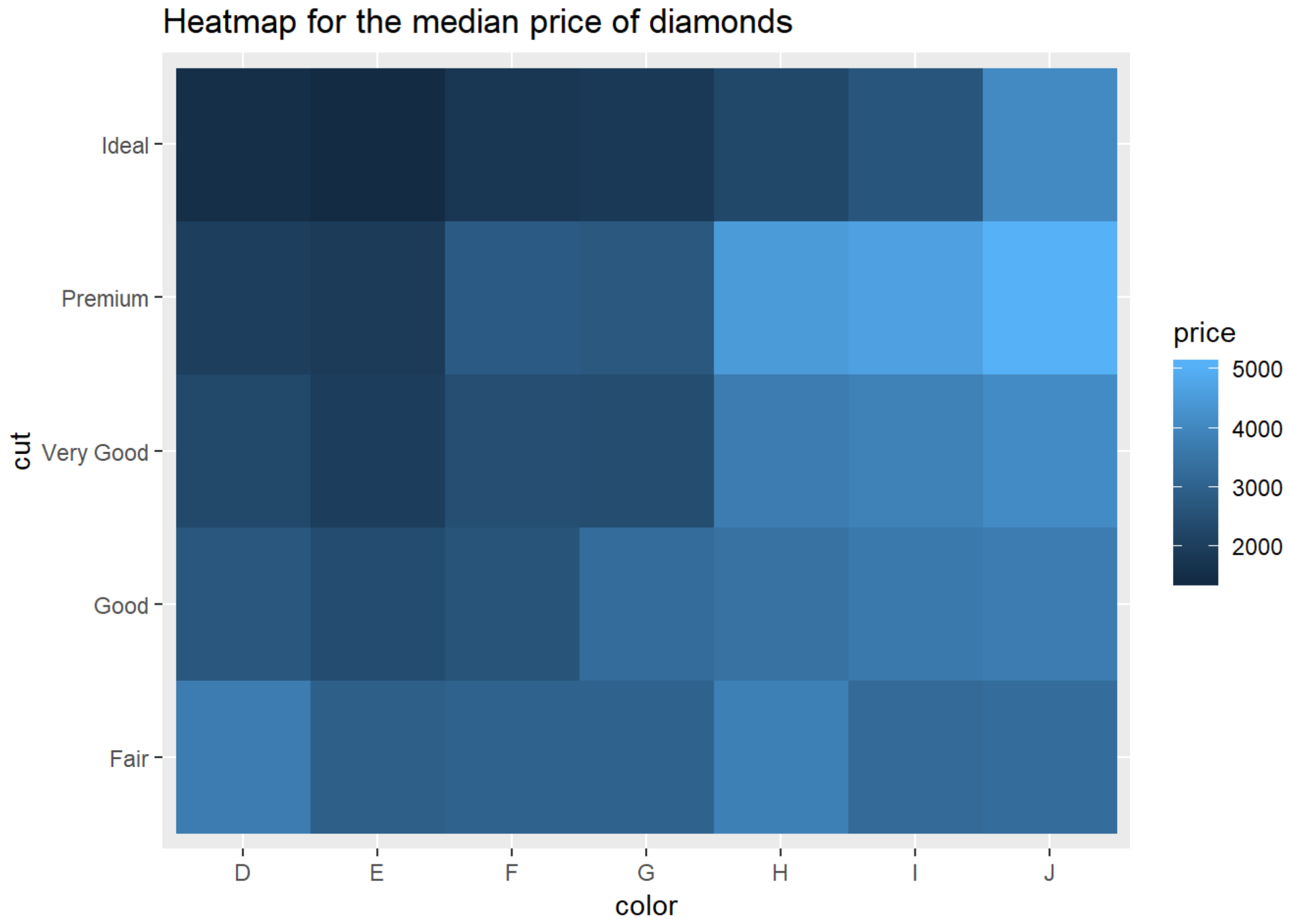
Plot a

Use `geom_raster` to create the heat maps for the plots you see below. The code provided below will create a new data frame that has the median price for each color and cut combination of the diamonds data frame.

```
diamonds.median <- aggregate(formula = price ~ color + cut,
                             data = diamonds, FUN = median)
str(diamonds.median)
```

```
'data.frame':   35 obs. of  3 variables:
 $ color: Ord.factor w/ 7 levels "D"<"E"<"F"<"G"<..: 1 2 3 4 5 6 7 1 2 3 ...
 $ cut  : Ord.factor w/ 5 levels "Fair"<"Good"<..: 1 1 1 1 1 1 1 2 2 2 ...
 $ price: num  3730 2956 3035 3057 3816 ...
```

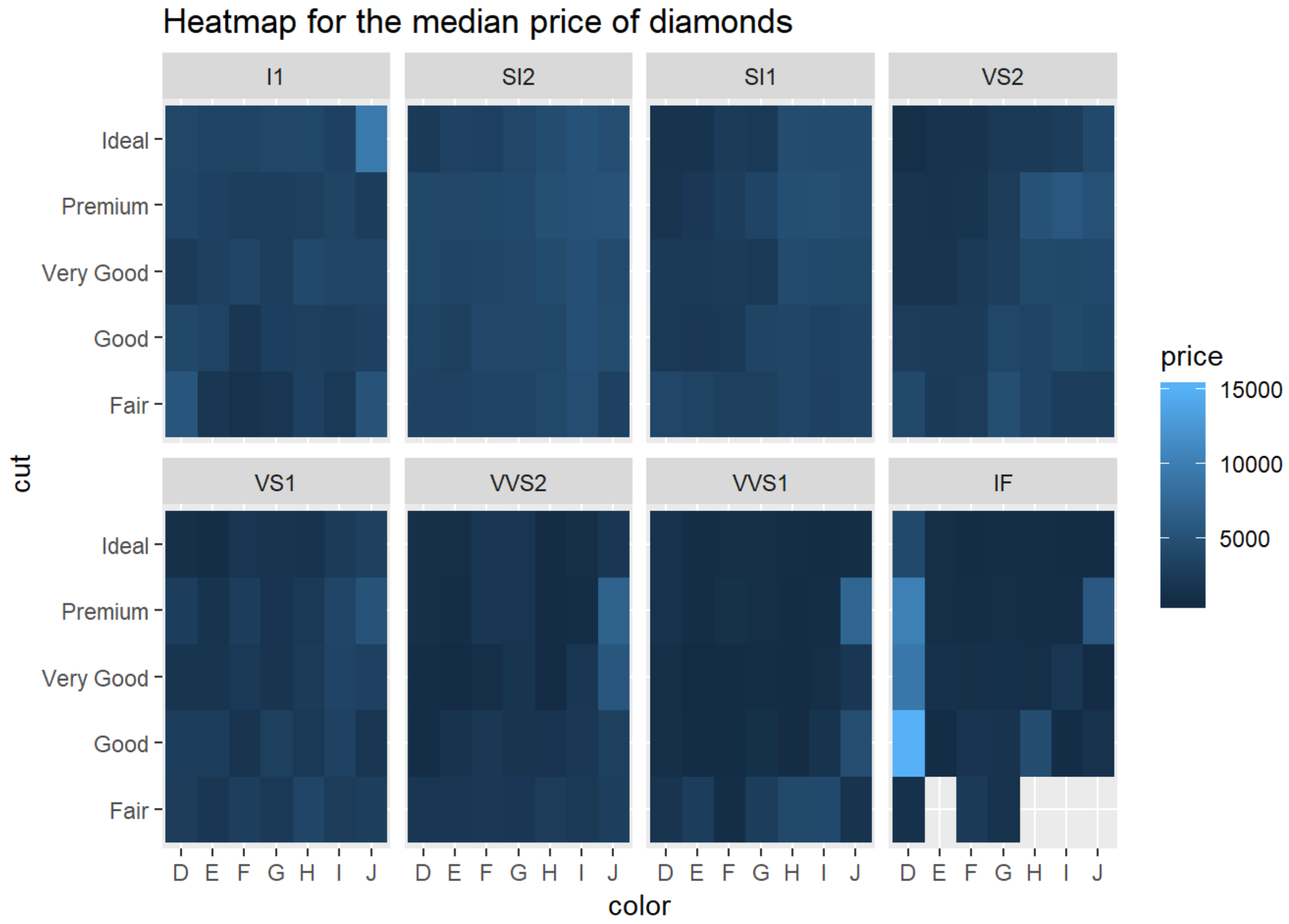
Use the data frame `diamonds.median` to create the below heatmap with `geom_raster`.



```
diamonds.median2 <- aggregate(formula = price ~ color + cut + clarity,
                              data = diamonds, FUN = median)
str(diamonds.median2)
```

```
'data.frame':   276 obs. of  4 variables:
 $ color  : Ord.factor w/ 7 levels "D"<"E"<"F"<"G"<..: 1 2 3 4 5 6 7 1 2 3 ...
 $ cut    : Ord.factor w/ 5 levels "Fair"<"Good"<..: 1 1 1 1 1 1 1 2 2 2 ...
 $ clarity: Ord.factor w/ 8 levels "I1"<"SI2"<"SI1"<..: 1 1 1 1 1 1 1 1 1 1 ...
 $ price  : num  5538 2036 1570 1954 3340 ...
```

Use the data frame `diamonds.median2` to create the below faceted heat maps with `geom_raster`.



Part 5

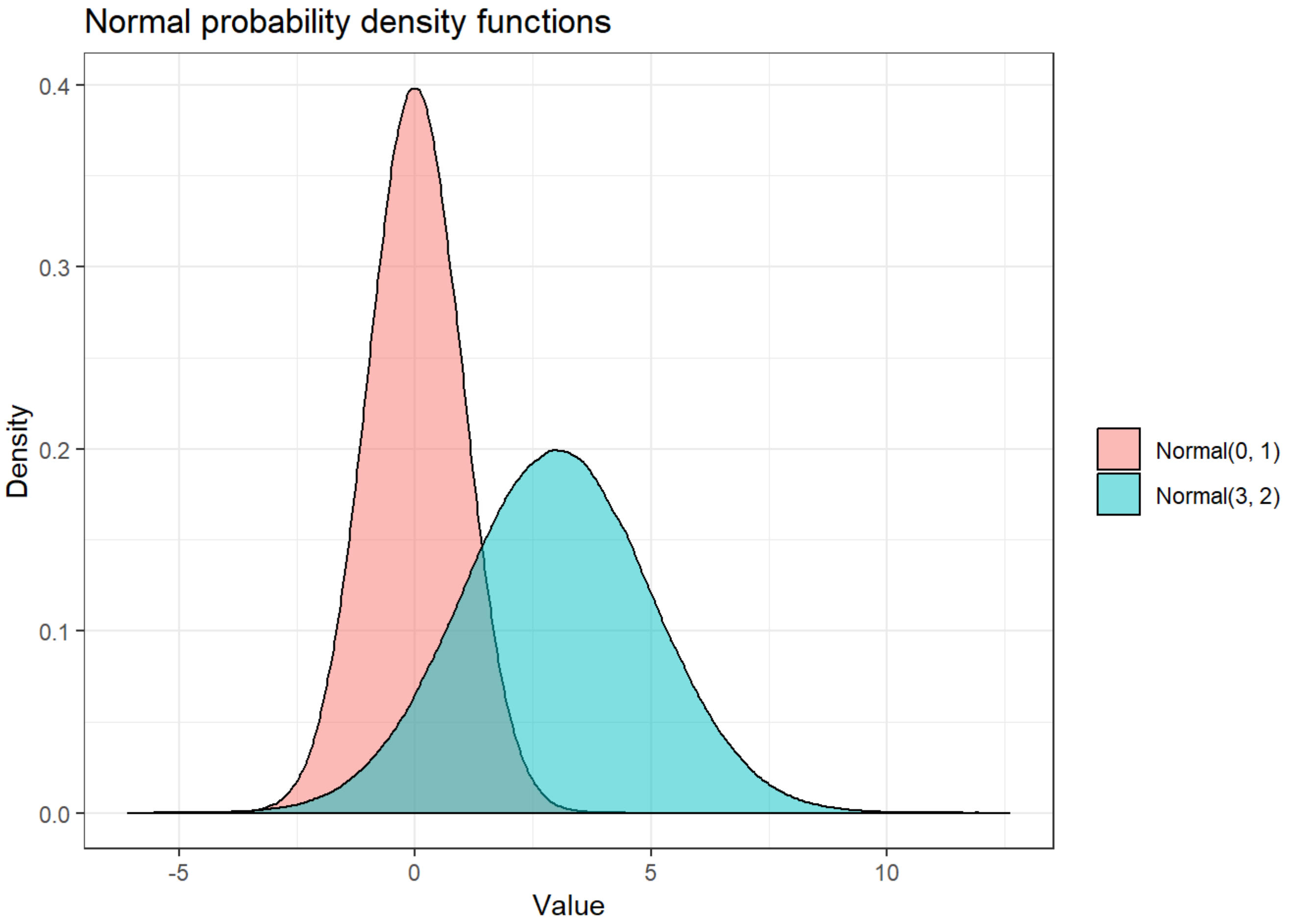
Density plot

Below is an example of a density plot. Run the ggplot part line by line to see what is happening as each layer is added.

```
# generate 1000000 random variables from two normal distributions
x <- rnorm(1000000, mean = 0, sd = 1)
y <- rnorm(1000000, mean = 3, sd = 2)

# create a data frame
norm_df <- data.frame(value = c(x, y), norm = rep(c("x", "y"), each = 1000000))

# plot the densities
ggplot(data = norm_df, aes(x = value, fill = norm)) +
  geom_density(alpha = 0.50) +
  labs(x = "Value", y = "Density",
       title = "Normal probability density functions")+
  theme_bw() +
  scale_fill_discrete(name = "",
                      breaks = c("x", "y"),
                      labels = c("Normal(0, 1)", "Normal(3, 2)"))
```


Normal probability density functions
Density
0.4
0.3
0.2
0.1
0.0
-5
0
5
10
Value
Normal(0, 1)
Normal(3, 2)