Data Visualisation with GGPlot

semicolon

Learning Outcomes

- At the end of the lesson, Natives will understand how to carry out different visualizations with Ggplot.

What’s ggplot

- ggplot2 is a data visualization library for R
- Developed by Hadley Wickham & Winston Chang
- ggplot2 is based on a **grammar**
- It’s powerful, flexible, beautiful, based on the features in your data, designed for iterative workflows

Packages and data

In this session, we will be learning how to visualise data using ggplot in R. We will be exploring a data on diamond. Let's start by loading the required dataset and pakage into R.

```
library(ggplot2)

dmd<-read.table("data/diamond.csv", header = T,
                dec = ".", sep = ",")
head(dmd)[,1:5]

##   carat       cut color clarity depth
## 1  0.23     Ideal     E     SI2  61.5
## 2  0.21   Premium     E     SI1  59.8
## 3  0.23      Good     E     VS1  56.9
## 4  0.29   Premium     I     VS2  62.4
## 5  0.31      Good     J     SI2  63.3
## 6  0.24 Very Good     J    VVS2  62.8
```

In creating plots using ggplot, there are three major things to specify:

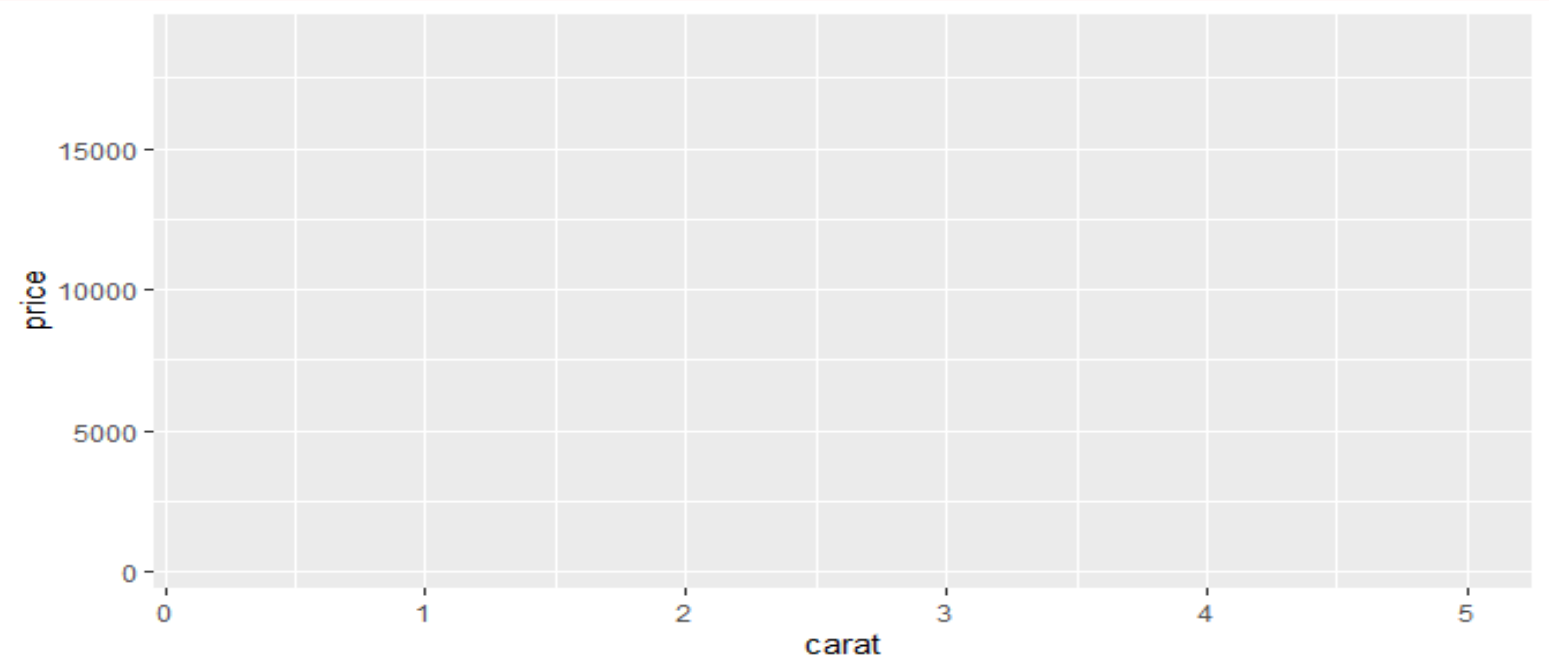
1: The data

2: The aesthetic mappings from the data variables to visual properties

3: The layer describing how to draw those properties

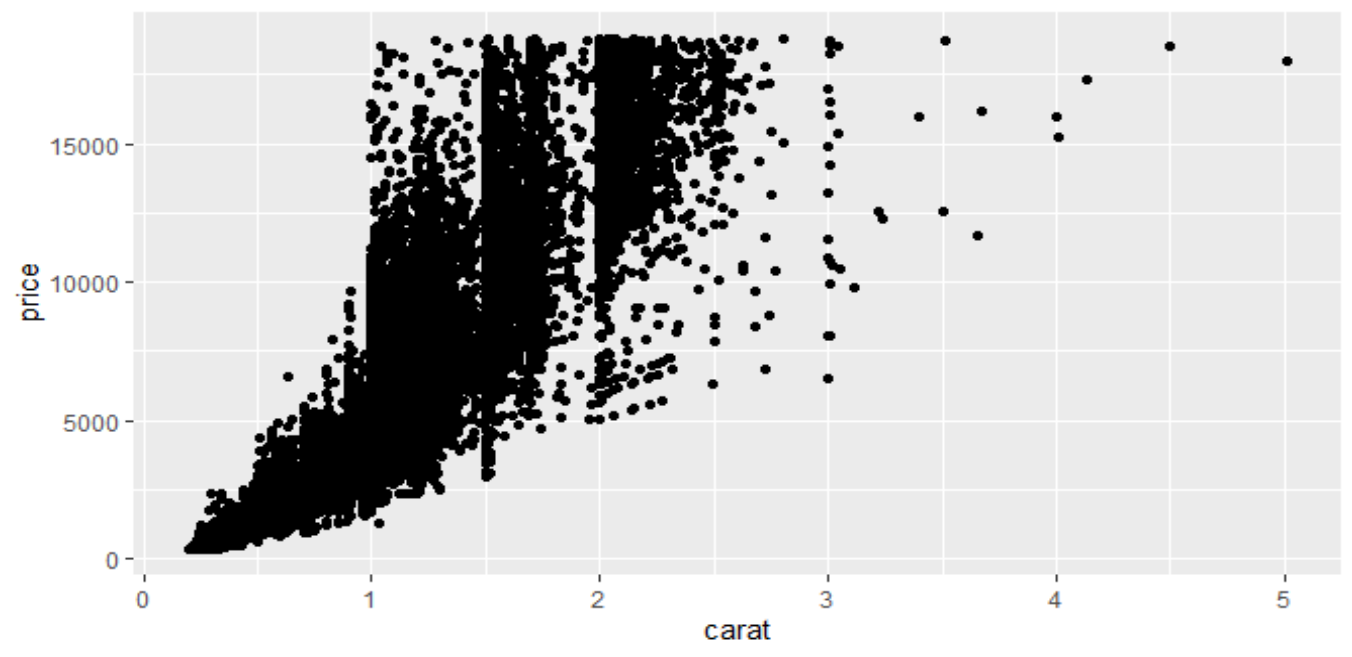
Data + aesthetic mapping

- **ggplot**(dmd, **aes**(x = carat, y = price))



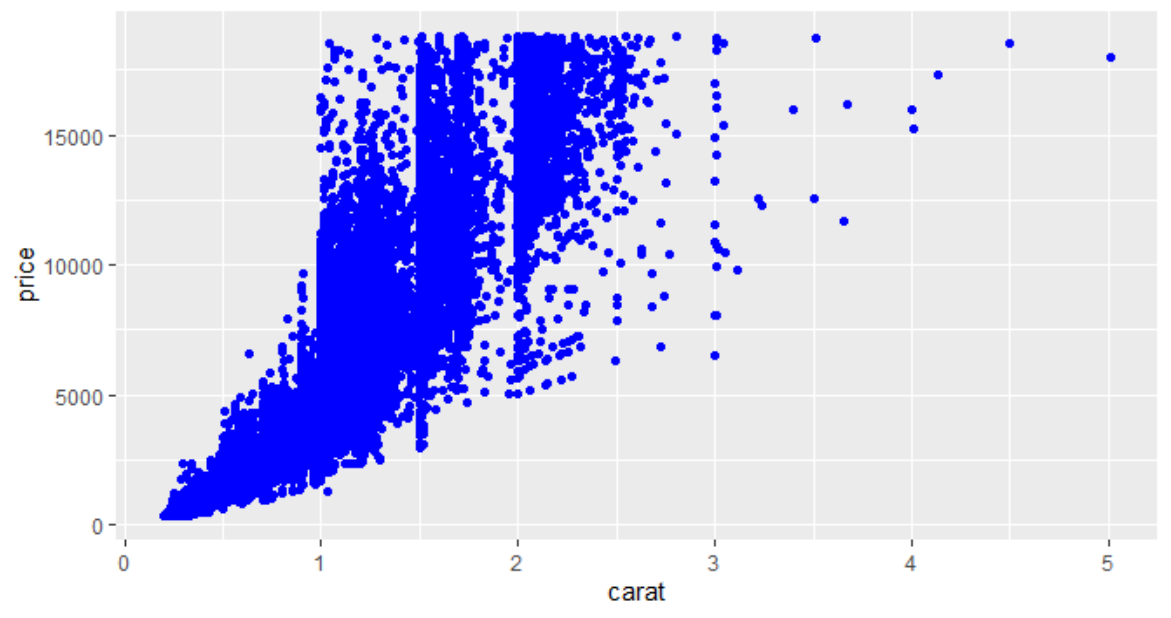
Scatter Plot using geom_point

- **ggplot**(dmd, **aes**(x = carat, y = price))+
  geom_point()



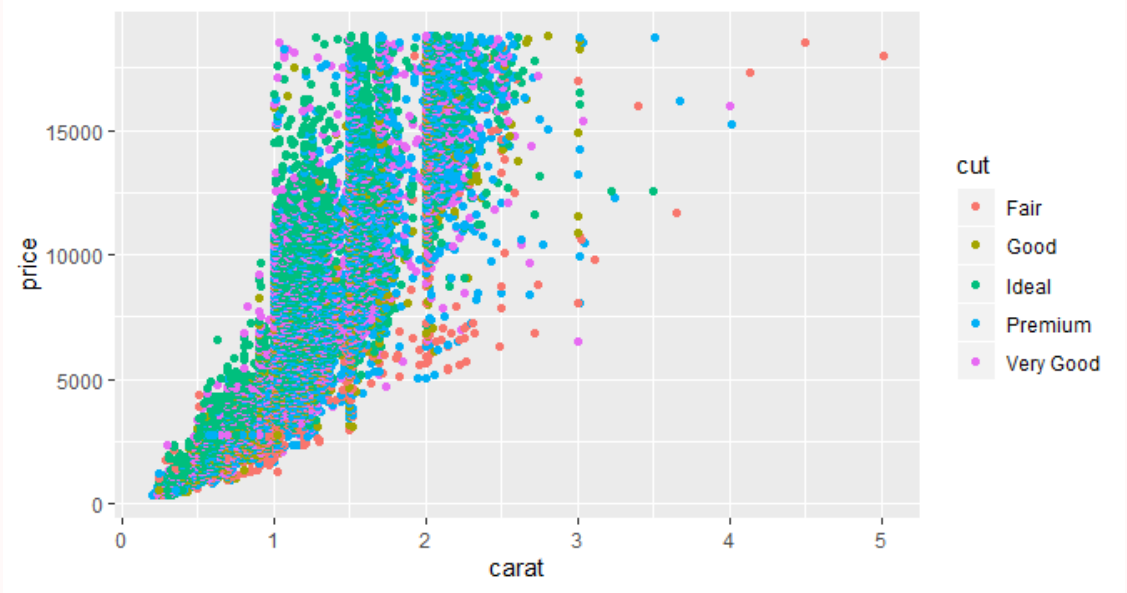
Scatter Plot using geom_point

- **ggplot**(dmd, **aes**(x = carat, y = price))+
  geom_point(colour="blue")



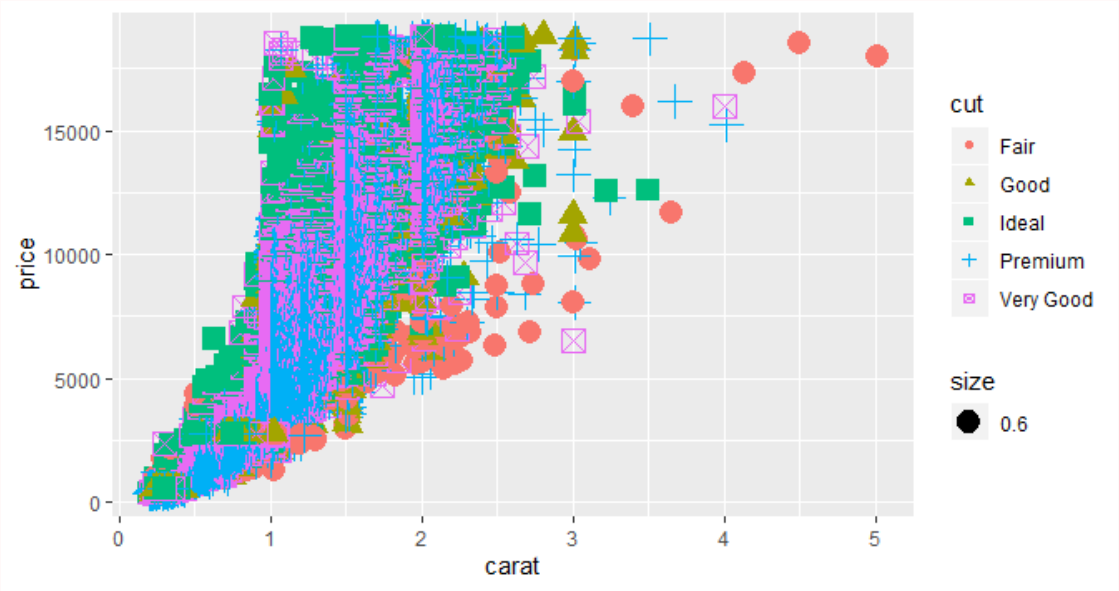
Scatter Plot using geom_point

- **ggplot**(dmd, **aes**(x = carat, y = price,
  colour=cut))+**geom_point**()



Adding shape to geom_point

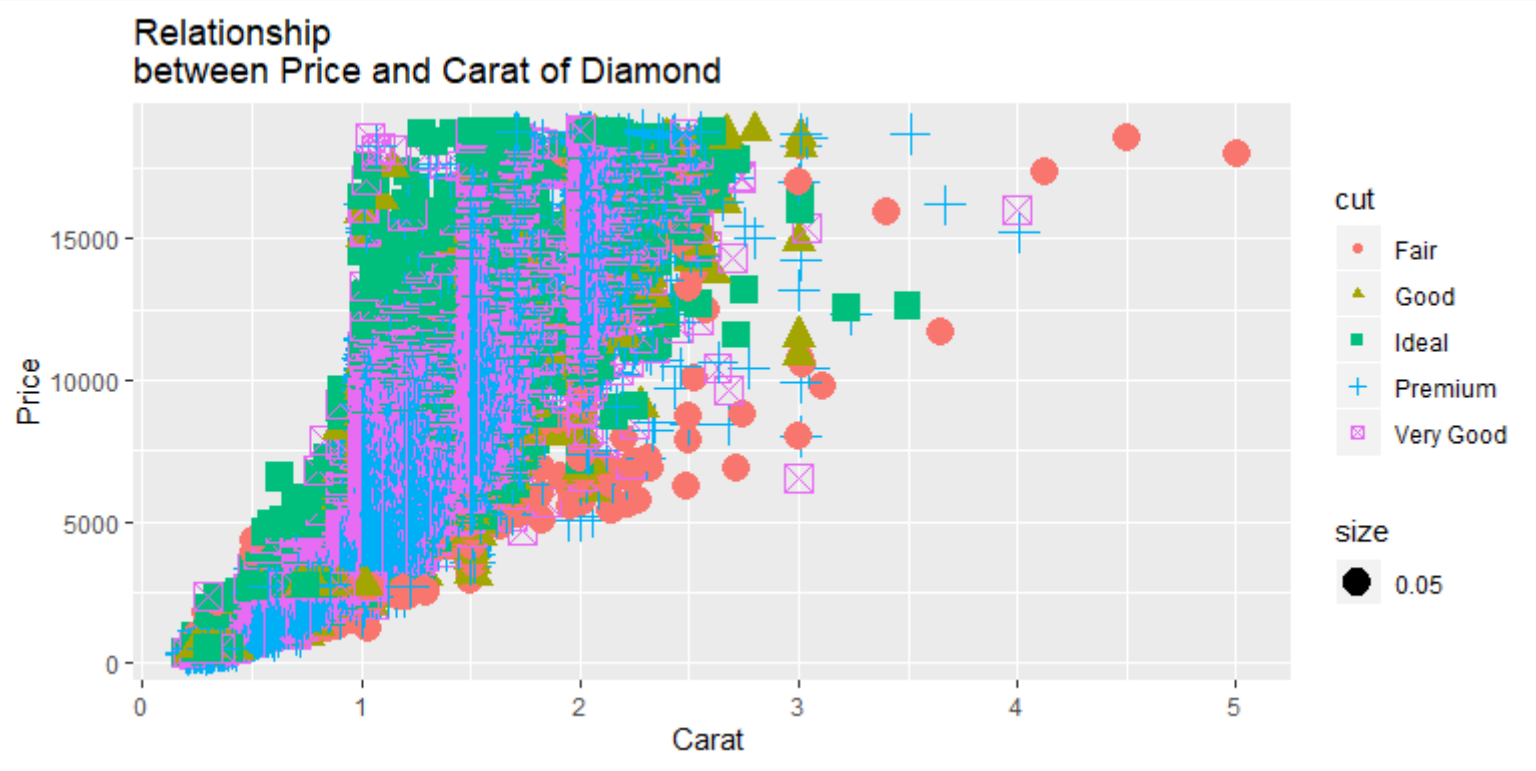
- **ggplot**(dmd, **aes**(x = carat, y = price))+
  geom_point(**aes**(colour=cut,shape=cut, size=0.6))



Adding Title to the Plot

- p1<-**ggplot**(dmd,**aes**(x = carat,y = price))+
  geom_point(**aes**(colour=cut,shape=cut,
  size=0.05))+
  labs(title="Relationship
  between Price and Carat of Diamond",
  x="Carat ", y="Price")
- p1

Adding Title to the Plot (Cont'd)

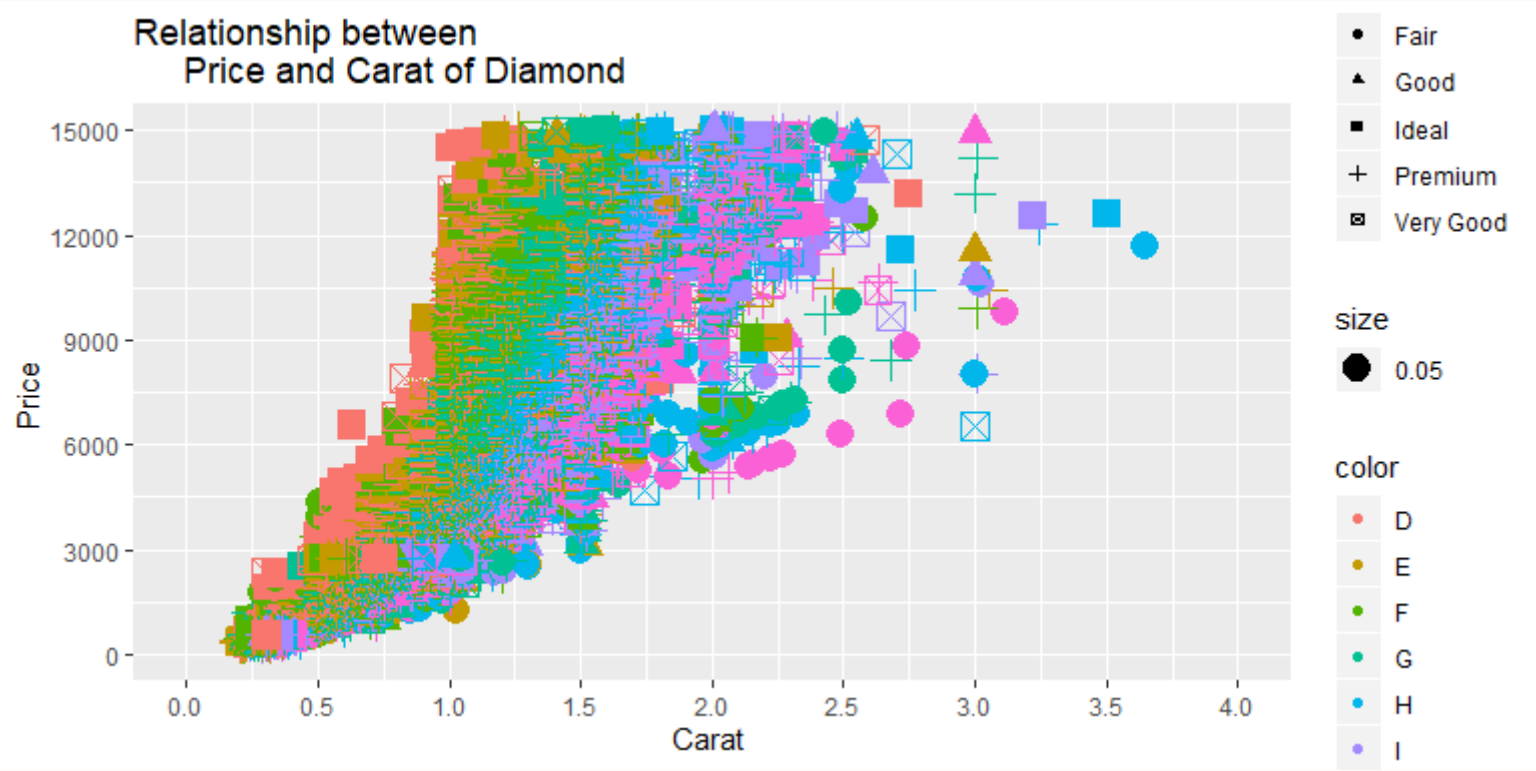


Setting Axis Limits

```
p2<-ggplot(dmd, aes(x = carat, y = price))+
geom_point(aes(colour=color,shape=cut,
                size=0.05))+
labs(title="Relationship between
     Price and Carat of Diamond")+
scale_y_continuous(name= "Price ",
                   limits = c(0,15000),
  breaks = seq(0, 15000, by = 3000))+
scale_x_continuous(name = "Carat ",
                   limits = c(0,4),
  breaks = seq(0, 4, by = 0.5))
```

```
suppressWarnings(print(p2))
```

Setting Axis Limits (Cont'd)

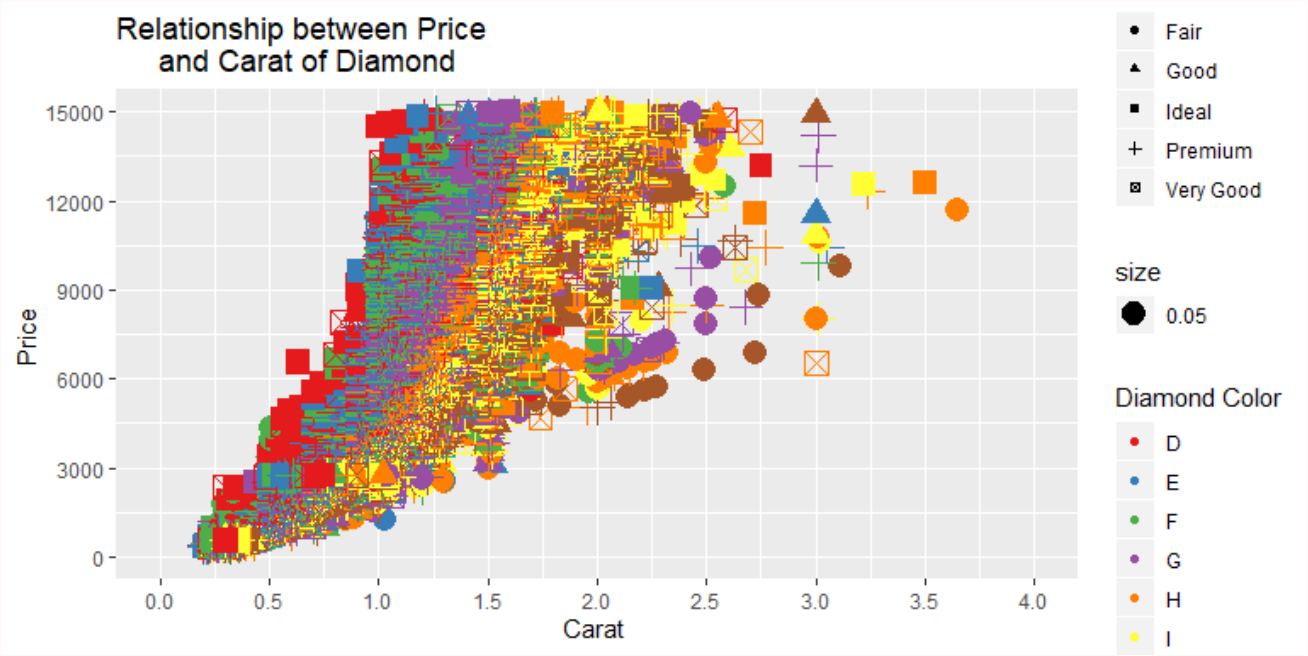


Adding Brewer Palette

```
p3<-ggplot(dmd, aes(x = carat, y = price))+
geom_point(aes(colour=color,shape=cut,
                size=0.05))+
labs(title="Relationship between Price
     and Carat of Diamond")+
scale_y_continuous(name= "Price ",
                   limits = c(0,15000),
breaks = seq(0, 15000, by = 3000))+
scale_x_continuous(name = "Carat ",
                   limits = c(0,4),
breaks = seq(0, 4, by = 0.5))+
  scale_color_brewer(name = "Diamond Color",
                     palette = "Set1")
```

Adding Brewer Palette (Cont'd)

- **suppressWarnings(print(p3))**

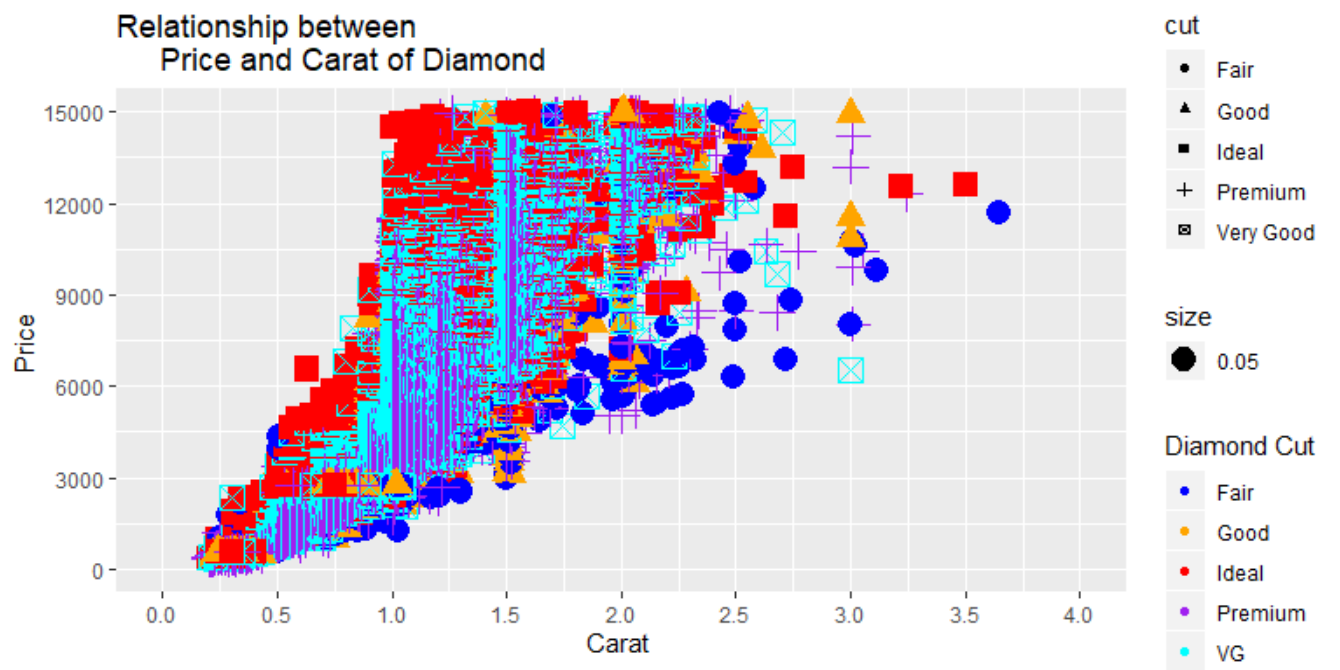


Customizing Brewer Palette

```
p4<-ggplot(dmd, aes(x = carat,
                    y = price))+
geom_point(aes(colour=cut,
               shape=cut,size=0.05))+
labs(title="Relationship between
     Price and Carat of Diamond")+
scale_y_continuous(name= "Price ",
                   limits = c(0,15000),
breaks = seq(0, 15000, by = 3000))+
scale_x_continuous(name = "Carat ",
limits = c(0,4), breaks = seq(0, 4, by = 0.5))+
  scale_color_manual(name = "Diamond Cut",
values = c("blue", "orange", "red", "purple", "cyan"),
labels=c("Fair","Good","Ideal","Premium","VG") )
```

Customizing Brewer Palette (Cont'd)

- **suppressWarnings(print(p4))**

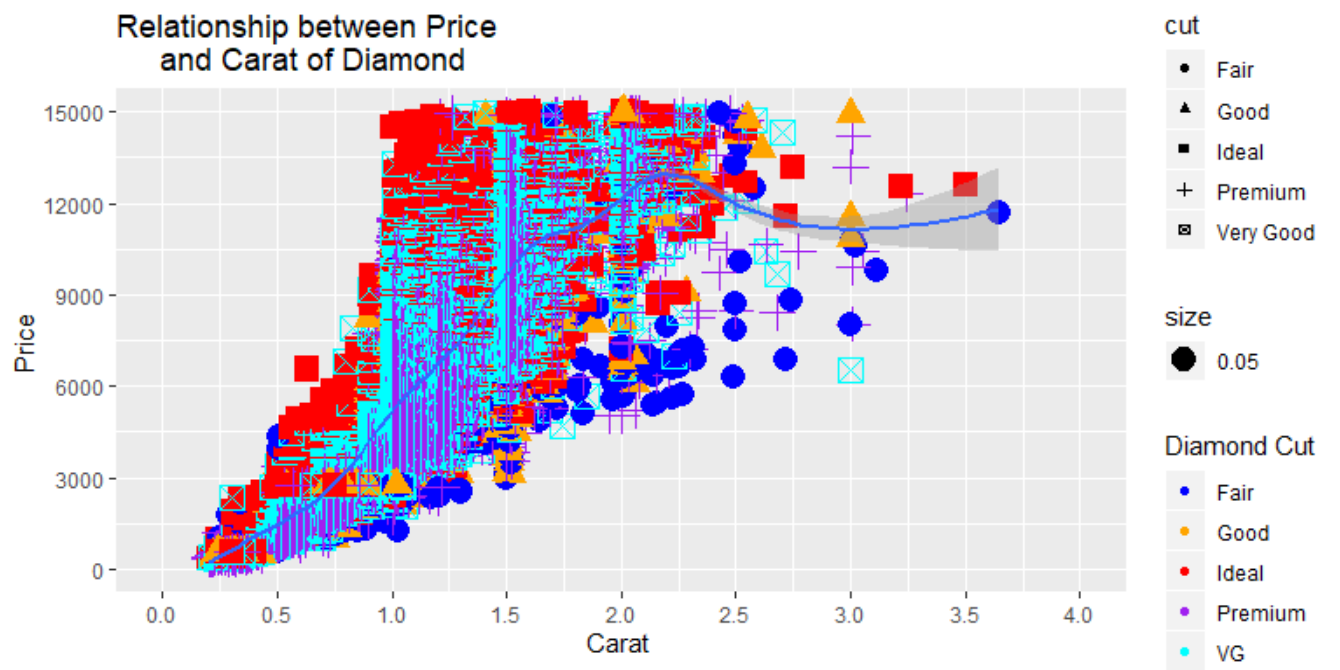


Adding geom_smooth()

```
p5<-ggplot(dmd, aes(x = carat, y = price))+
geom_point(aes(colour=cut,shape=cut,
               size=0.05))+
geom_smooth()+
labs(title="Relationship between Price
     and Carat of Diamond")+
scale_y_continuous(name= "Price ",
      limits = c(0,15000),
      breaks = seq(0, 15000, by = 3000))+
scale_x_continuous(name = "Carat ", limits = c(0,4),
                   breaks = seq(0, 4, by = 0.5))+
  scale_color_manual(name = "Diamond Cut",
  values = c("blue", "orange", "red", "purple", "cyan"),
    labels=c("Fair","Good","Ideal","Premium","VG") )
```

Adding geom_smooth() (Cont'd)

- **suppressWarnings(print(p5))**

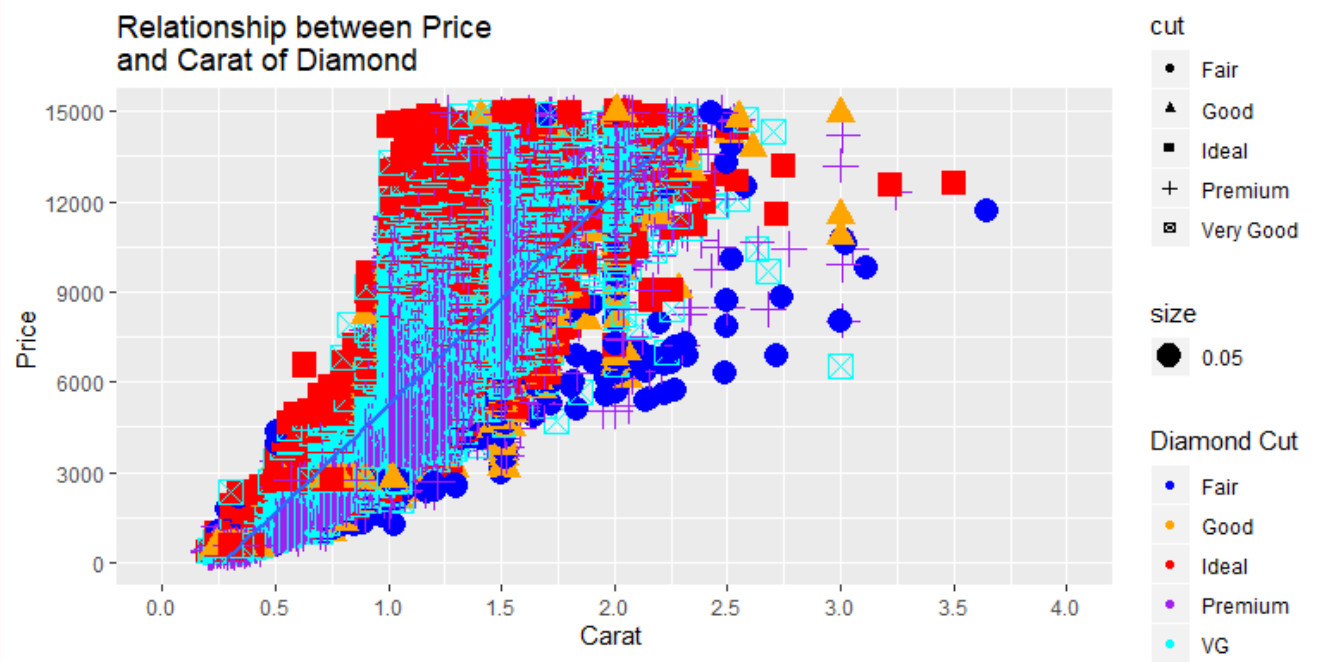


Adding geom_smooth() (StaightLine)

```
p6<-ggplot(dmd, aes(x = carat, y = price))+
geom_point(aes(colour=cut,
shape=cut,size=0.05))+geom_smooth(method = "lm") +
labs(title="Relationship between Price
and Carat of Diamond")+
scale_y_continuous(name= "Price ",
limits = c(0,15000),
breaks = seq(0, 15000, by = 3000))+
scale_x_continuous(name = "Carat ",
limits = c(0,4), breaks = seq(0, 4, by = 0.5))+
scale_color_manual(name = "Diamond Cut",
values = c("blue", "orange", "red", "purple", "cyan"),
labels=c("Fair","Good","Ideal","Premium","VG") )
```

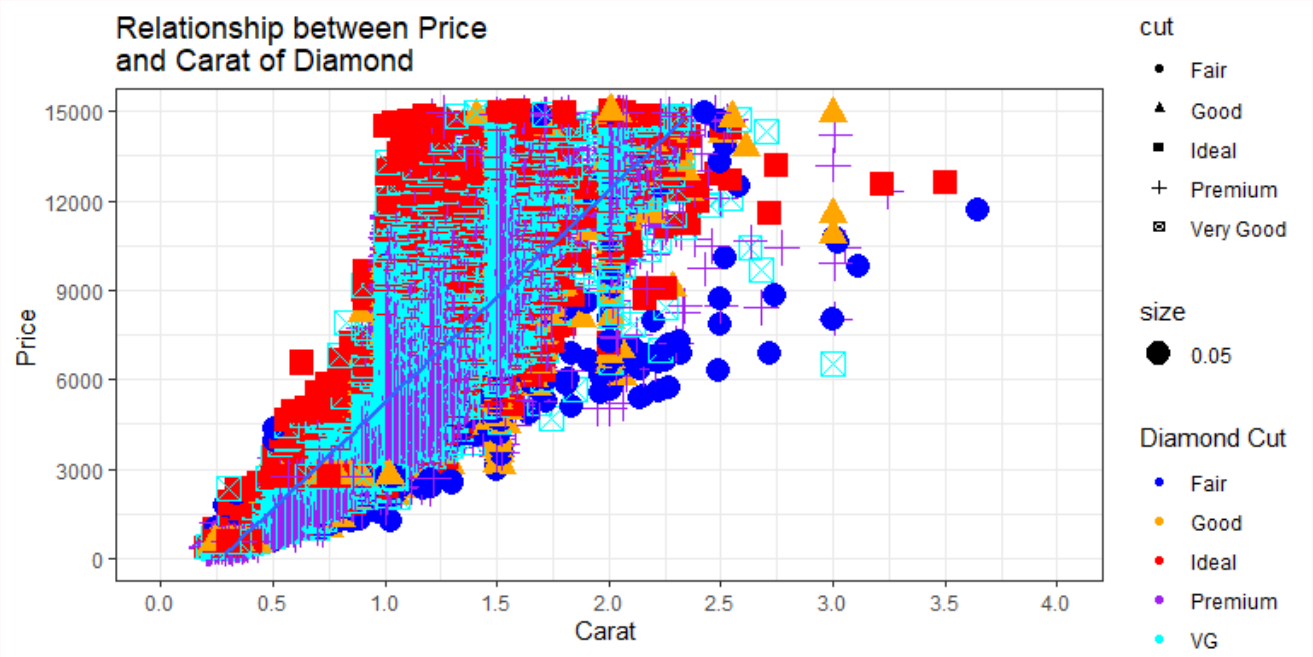
Adding geom_smooth() (StaightLine) (Cont'd)

- **suppressWarnings(print(p6))**



Adding theme_bw()(Cont'd)

- **suppressWarnings(print(p7))**

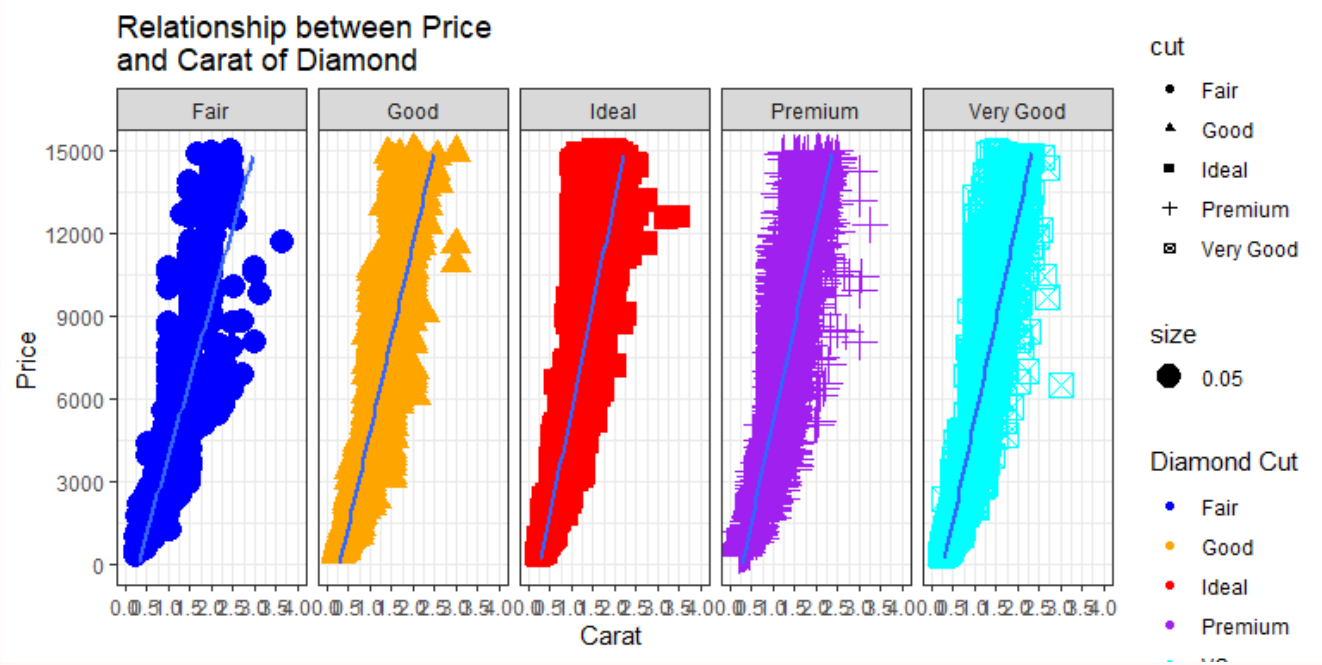


Facetting

```
p8<-ggplot(dmd, aes(x = carat, y = price))+
geom_point(aes(colour=cut,shape=cut,
size=0.05))+geom_smooth(method = "lm") +
labs(title="Relationship between Price
and Carat of Diamond")+
scale_y_continuous(name= "Price ",
limits = c(0,15000), breaks = seq(0, 15000, by = 3000))+
scale_x_continuous(name = "Carat ", limits = c(0,4),
  breaks = seq(0, 4, by = 0.5))+
  scale_color_manual(name = "Diamond Cut",
values = c("blue", "orange", "red", "purple", "cyan"),
labels=c("Fair","Good","Ideal","Premium","VG") )+
facet_grid( ~ cut)+theme_bw()
```

Facetting Cont'd

- **suppressWarnings(print(p8))**

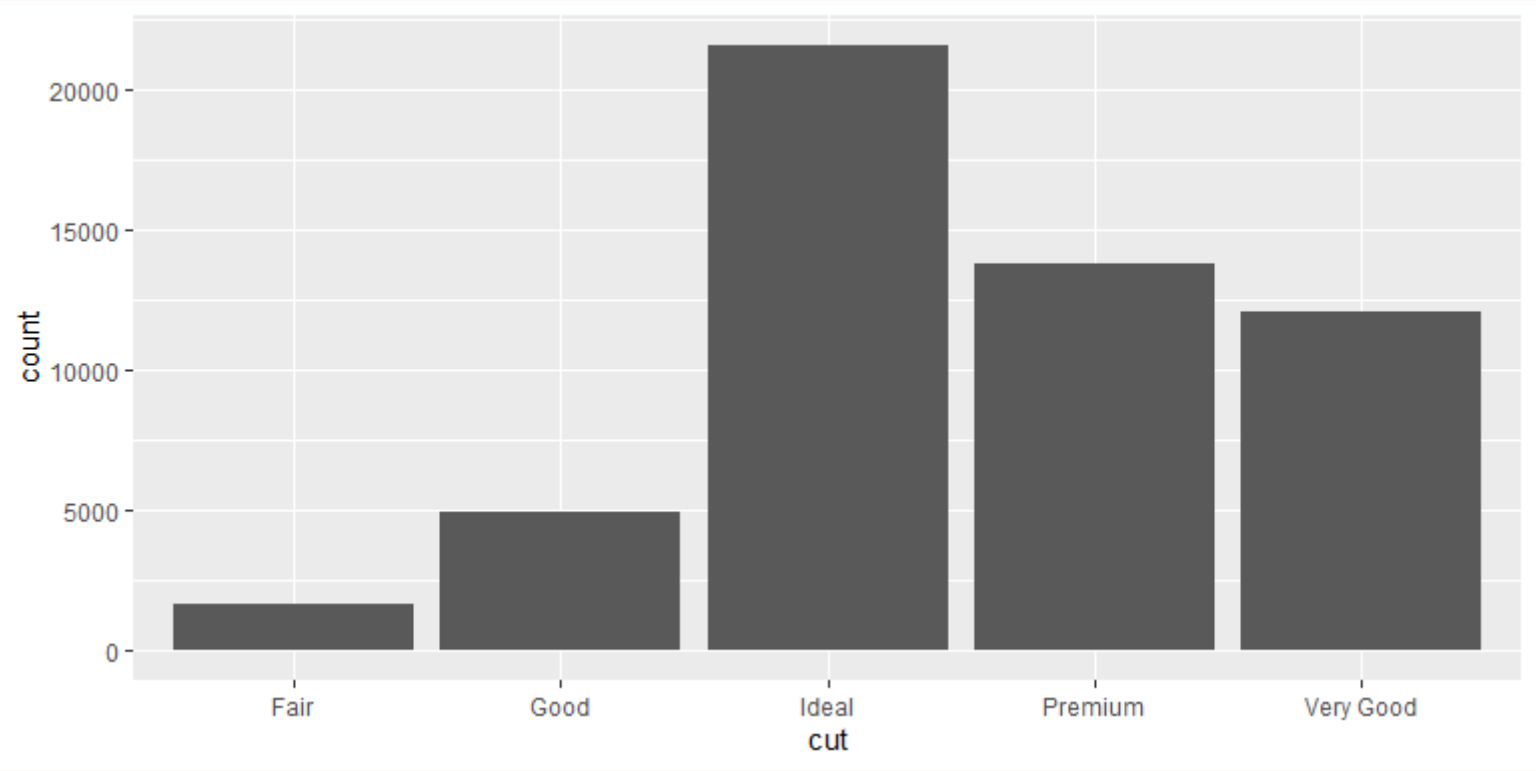


semicolon

Bar charts in ggplot using geom_bar

```
p9<-ggplot(dmd, aes(x = cut))+
  geom_bar()
suppressWarnings(print(p9))
```

Bar charts in ggplot using geom_bar

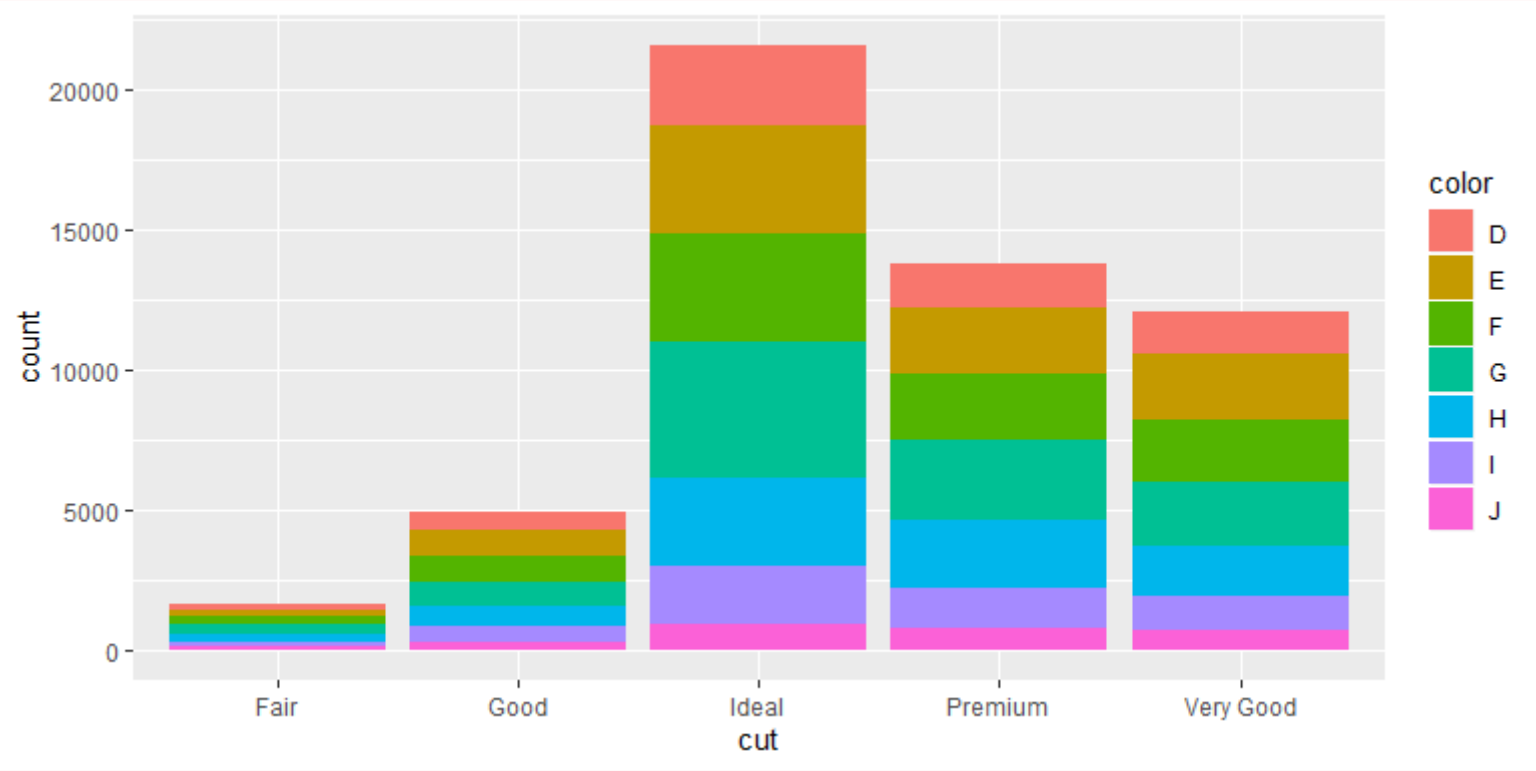


semicolon

Bar charts with Fill

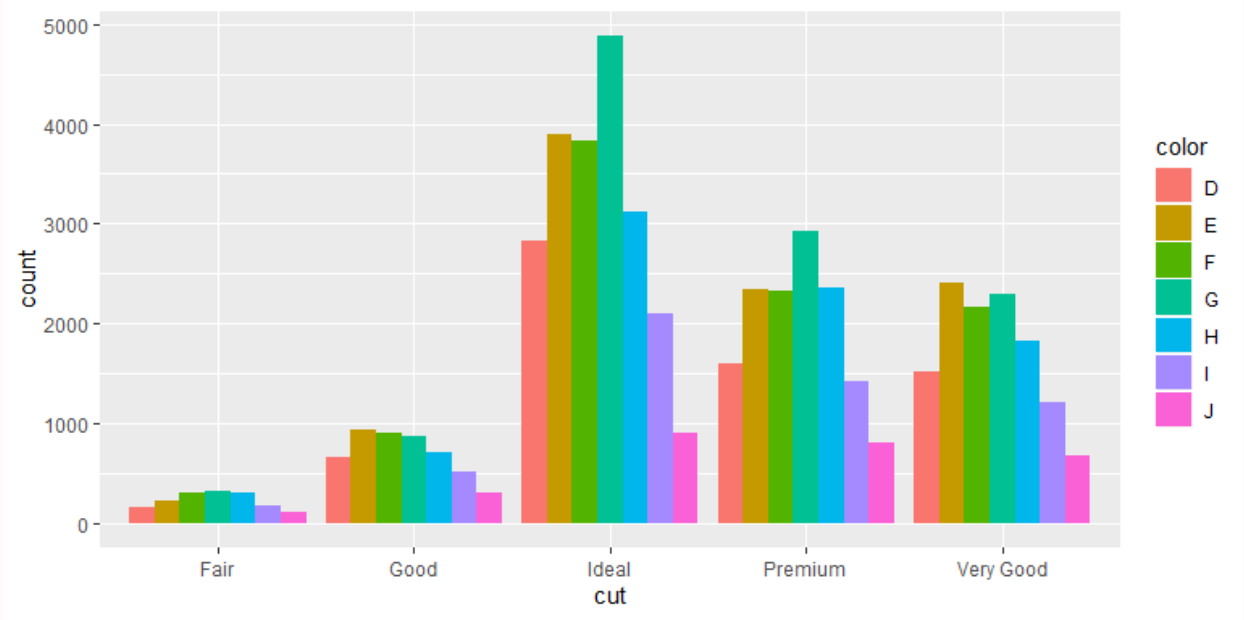
- **ggplot**(dmd, **aes**(x = cut, fill = color))+
  geom_bar()

Bar charts with Fill

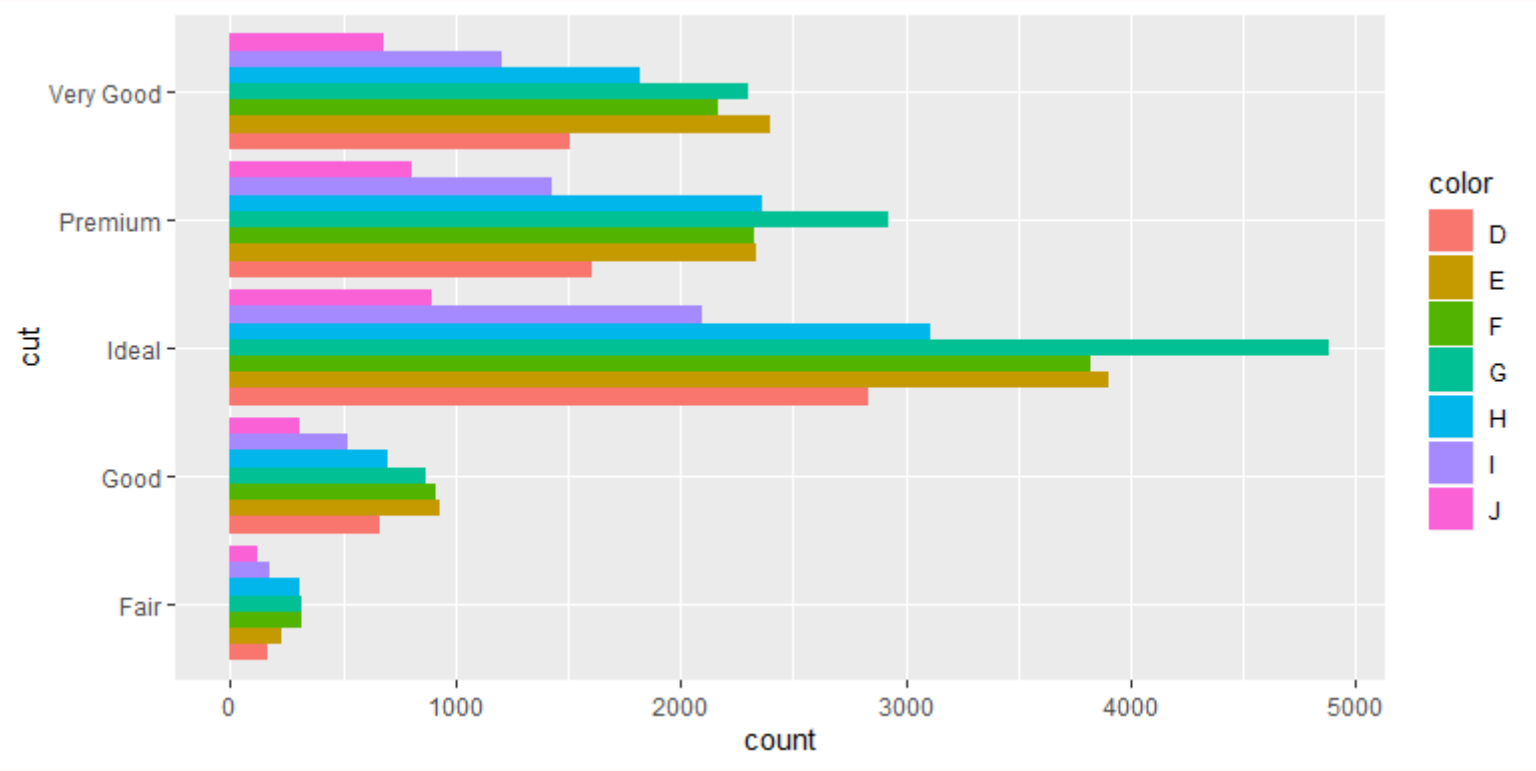
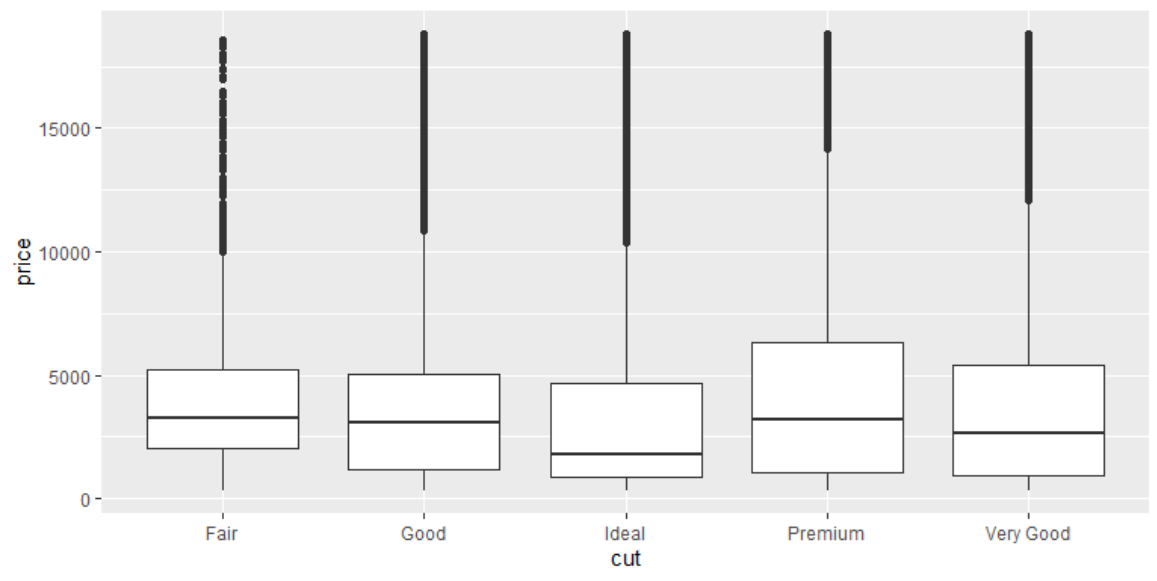
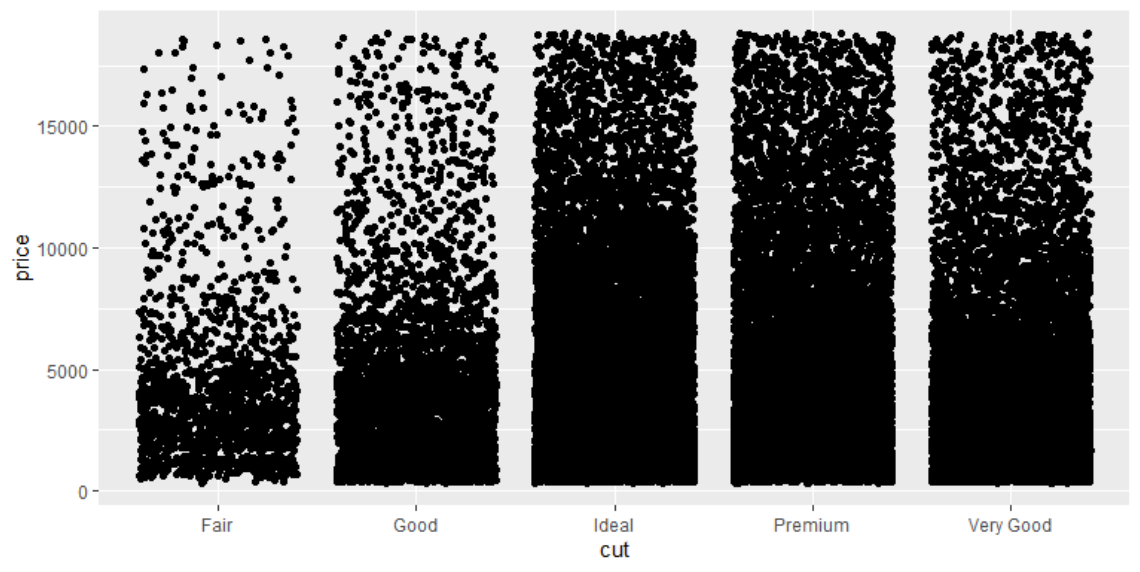
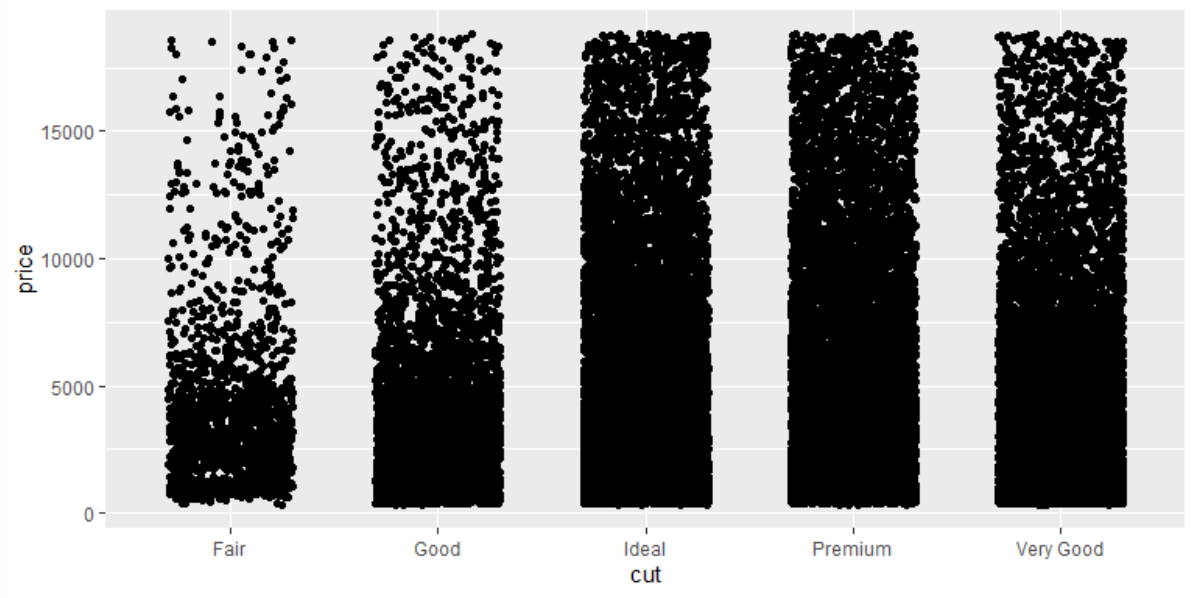


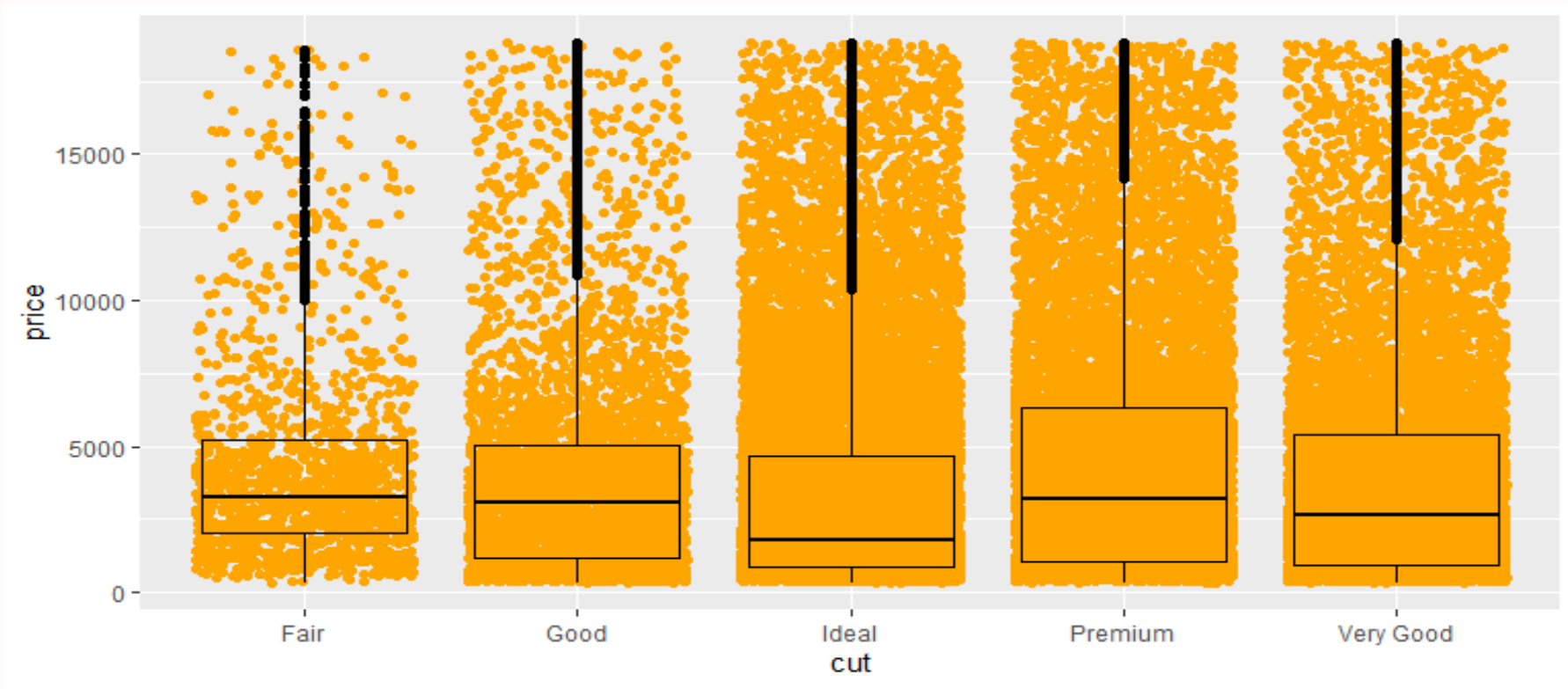
Positionings

- **ggplot**(dmd, **aes**(x = cut, fill = color))+
  geom_bar(position = "dodge")



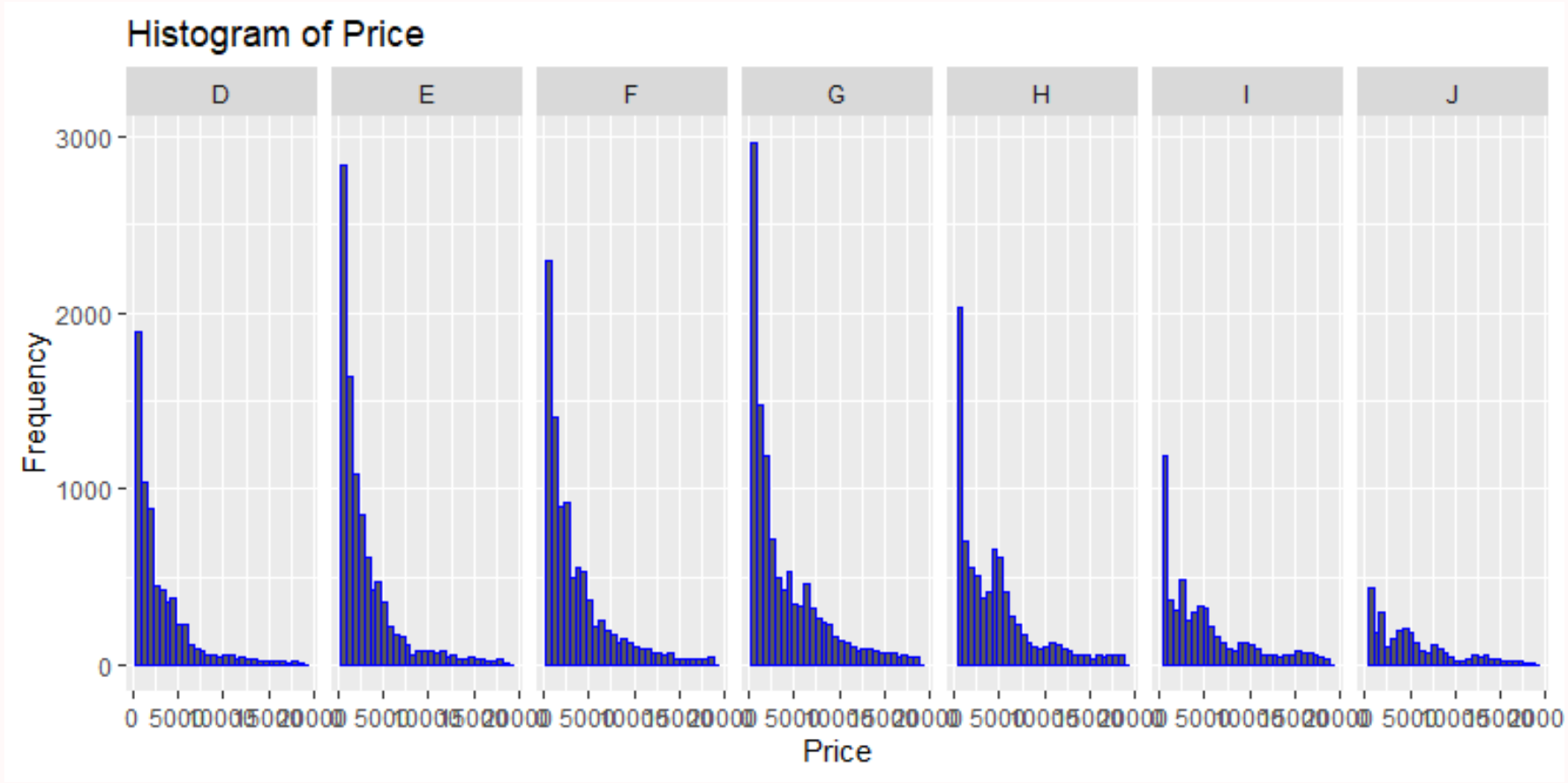
Flipping the x and the y axis

- ```
  ggplot(dmd, aes(x = cut, fill = color))+
  geom_bar(position = "dodge")+
  coord_flip()
  ```
  ```

Flipping the x and the y axis



semicolon

Continous and categorical variable: geom_boxplot()

- **ggplot**(dmd, **aes**(y = price, x = cut)) +
  geom_boxplot()



continuous and a categorical variable: geom_jitter()

- **ggplot**(dmd, **aes**(y = price, x = cut)) +
  geom_jitter()



geom_jitter() Cont'd

- **ggplot**(dmd, **aes**(y = price, x = cut)) +
  geom_jitter(height = 0, width = 0.3)



Combining geom_jitter() and geom_boxplot()

- ```
  ggplot(dmd, aes(y = price, x = cut)) +
    geom_jitter(color = "orange") +
    geom_boxplot(fill = NA, color = "black")
  ```
  ```


price
15000
10000
5000
0
Fair
Good
Ideal
Premium
Very Good
cut

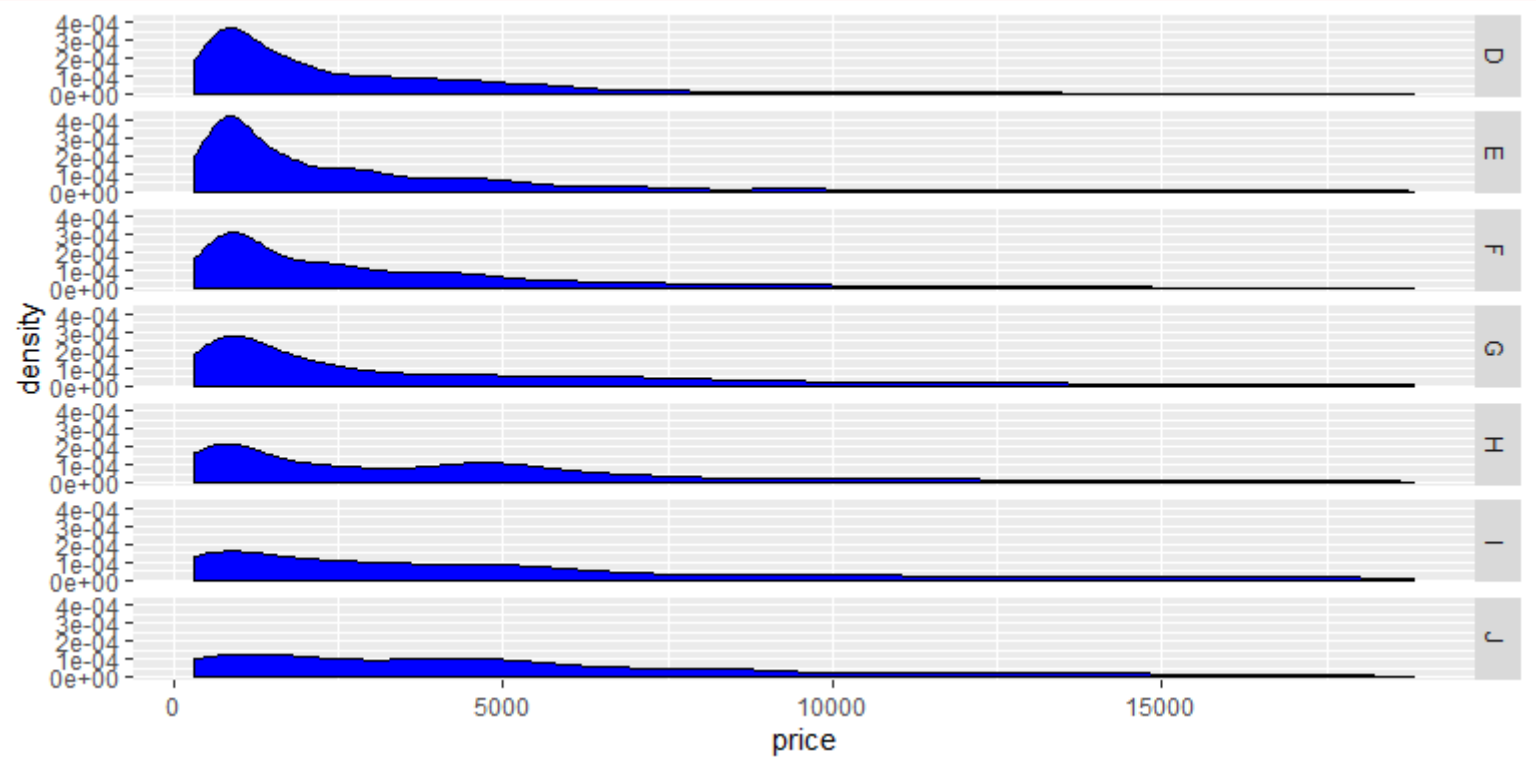
Using geom_hist

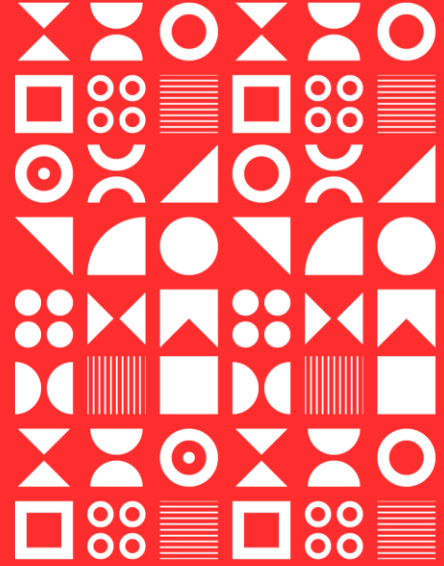
```
ggplot(dmd, aes(x = price)) +
geom_histogram(color =  "blue")+
 labs(title = "Histogram of Price",
x="Price",y="Frequency")+ facet_grid(~color)

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


Histogram of Price
D
E
F
G
H
I
J
Frequency
3000
2000
1000
0
0 5000 10000 15000 20000
Price

semicolon

Using geom_density





semicolon