Diamonds.R

```r
library(ggplot2)

data(diamonds)


View(diamonds)
names(diamonds)
```

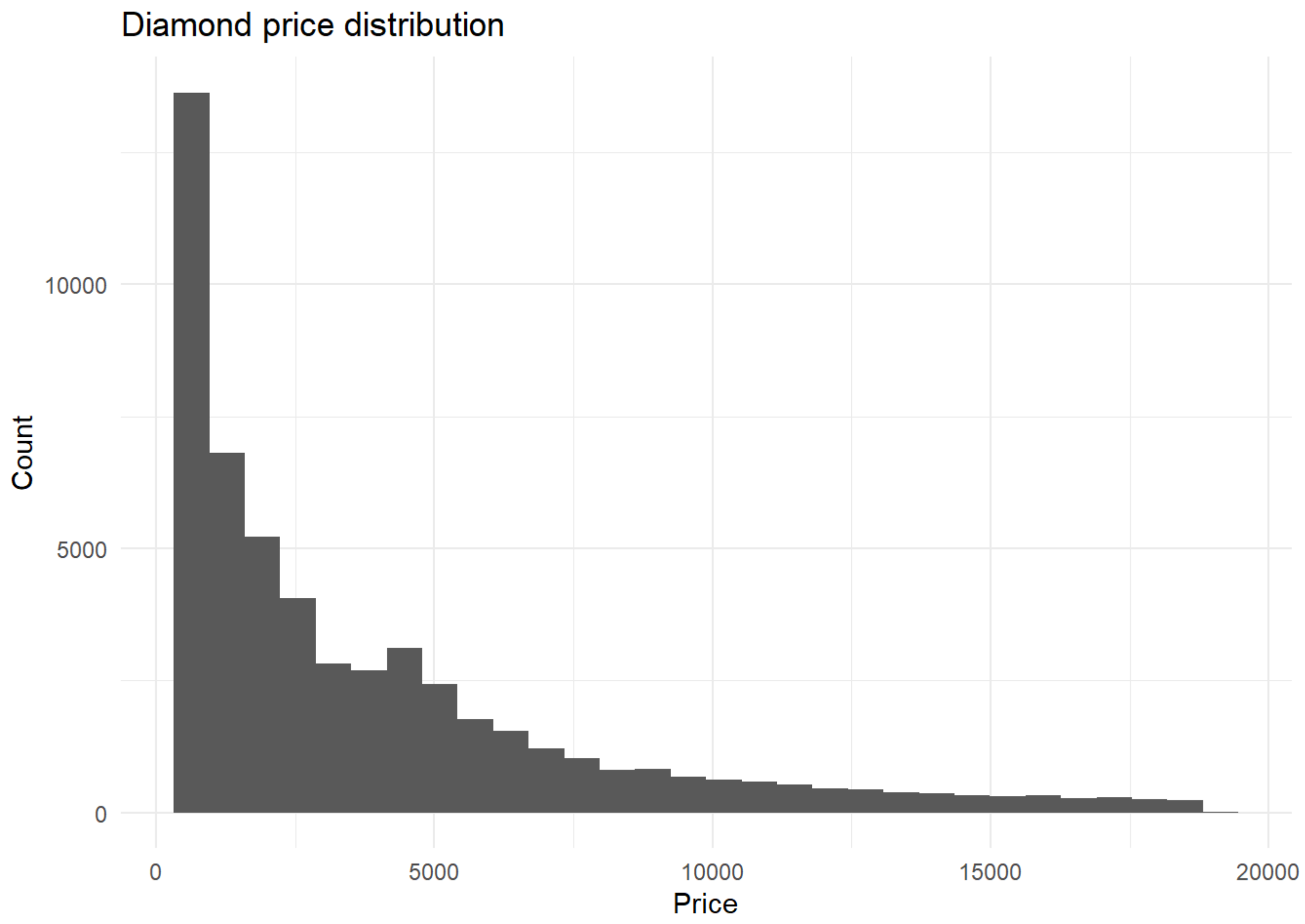
```
##  [1] "carat"   "cut"     "color"   "clarity" "depth"   "table"   "price"  
##  [8] "x"       "y"       "z"
```

```r
summary(diamonds)
```

```
##      carat               cut        color        clarity     
##  Min.   :0.2000   Fair     : 1610   D: 6775   SI1    :13065  
##  1st Qu.:0.4000   Good     : 4906   E: 9797   VS2    :12258  
##  Median :0.7000   Very Good:12082   F: 9542   SI2    : 9194  
##  Mean   :0.7979   Premium  :13791   G:11292   VS1    : 8171  
##  3rd Qu.:1.0400   Ideal    :21551   H: 8304   VVS2   : 5066  
##  Max.   :5.0100                     I: 5422   VVS1   : 3655  
##                                     J: 2808   (Other): 2531  
##      depth           table           price             x         
##  Min.   :43.00   Min.   :43.00   Min.   :  326   Min.   : 0.000  
##  1st Qu.:61.00   1st Qu.:56.00   1st Qu.:  950   1st Qu.: 4.710  
##  Median :61.80   Median :57.00   Median : 2401   Median : 5.700  
##  Mean   :61.75   Mean   :57.46   Mean   : 3933   Mean   : 5.731  
##  3rd Qu.:62.50   3rd Qu.:59.00   3rd Qu.: 5324   3rd Qu.: 6.540  
##  Max.   :79.00   Max.   :95.00   Max.   :18823   Max.   :10.740  
##                                                                  
##        y                z         
##  Min.   : 0.000   Min.   : 0.000  
##  1st Qu.: 4.720   1st Qu.: 2.910  
##  Median : 5.710   Median : 3.530  
##  Mean   : 5.735   Mean   : 3.539  
##  3rd Qu.: 6.540   3rd Qu.: 4.040  
##  Max.   :58.900   Max.   :31.800  
## 
```

```r
ggplot(data = diamonds, aes(x = price)) +
  geom_histogram() +
  ggtitle("Diamond price distribution") +
  xlab("Price") +
  ylab("Count") +
  theme_minimal()
```

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



```
sum(diamonds$price < 500)
```

```
## [1] 1729
```

```
sum(diamonds$price < 250)
```

```
## [1] 0
```

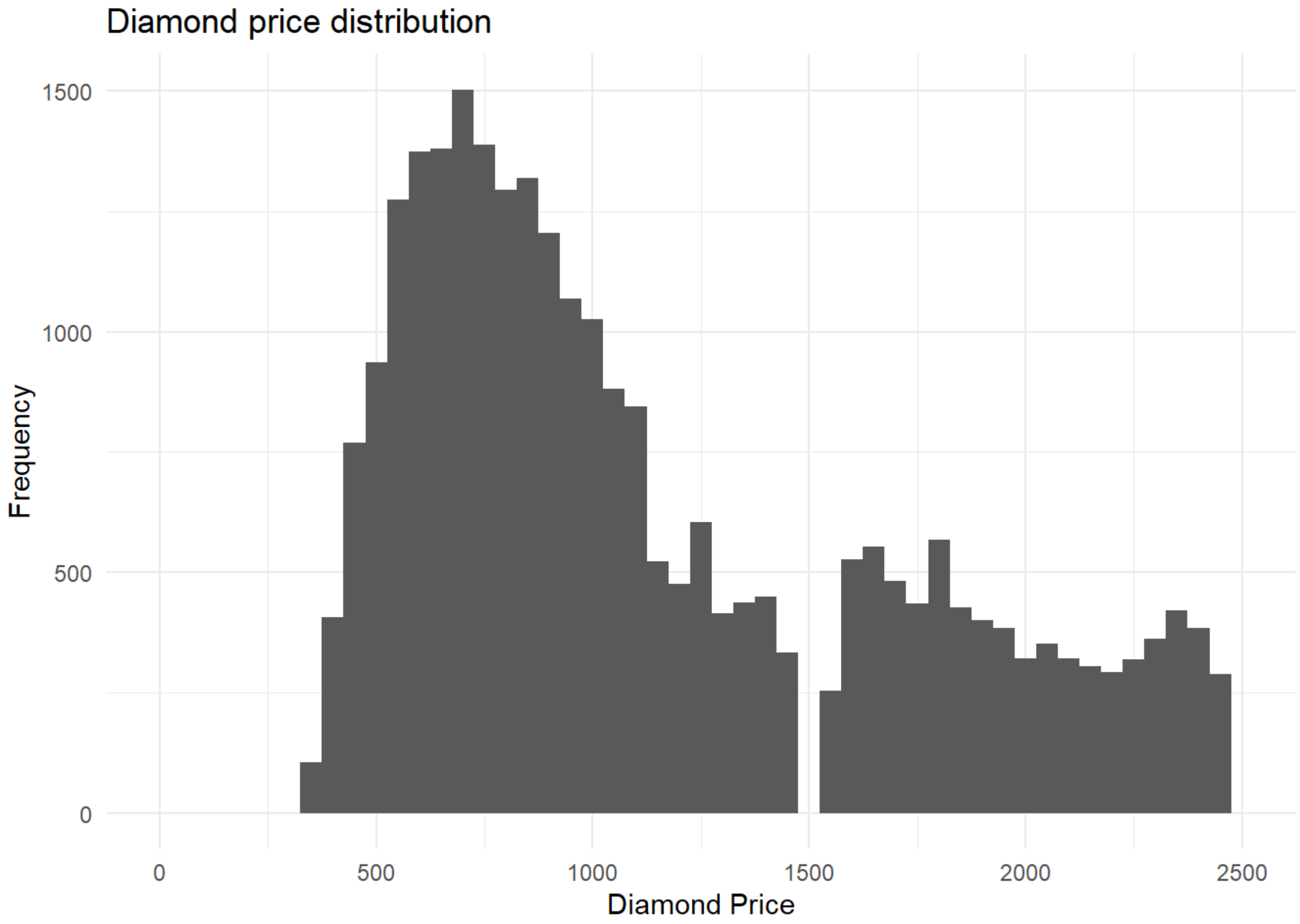
```
sum(diamonds$price >= 15000)
```

```
## [1] 1656
```

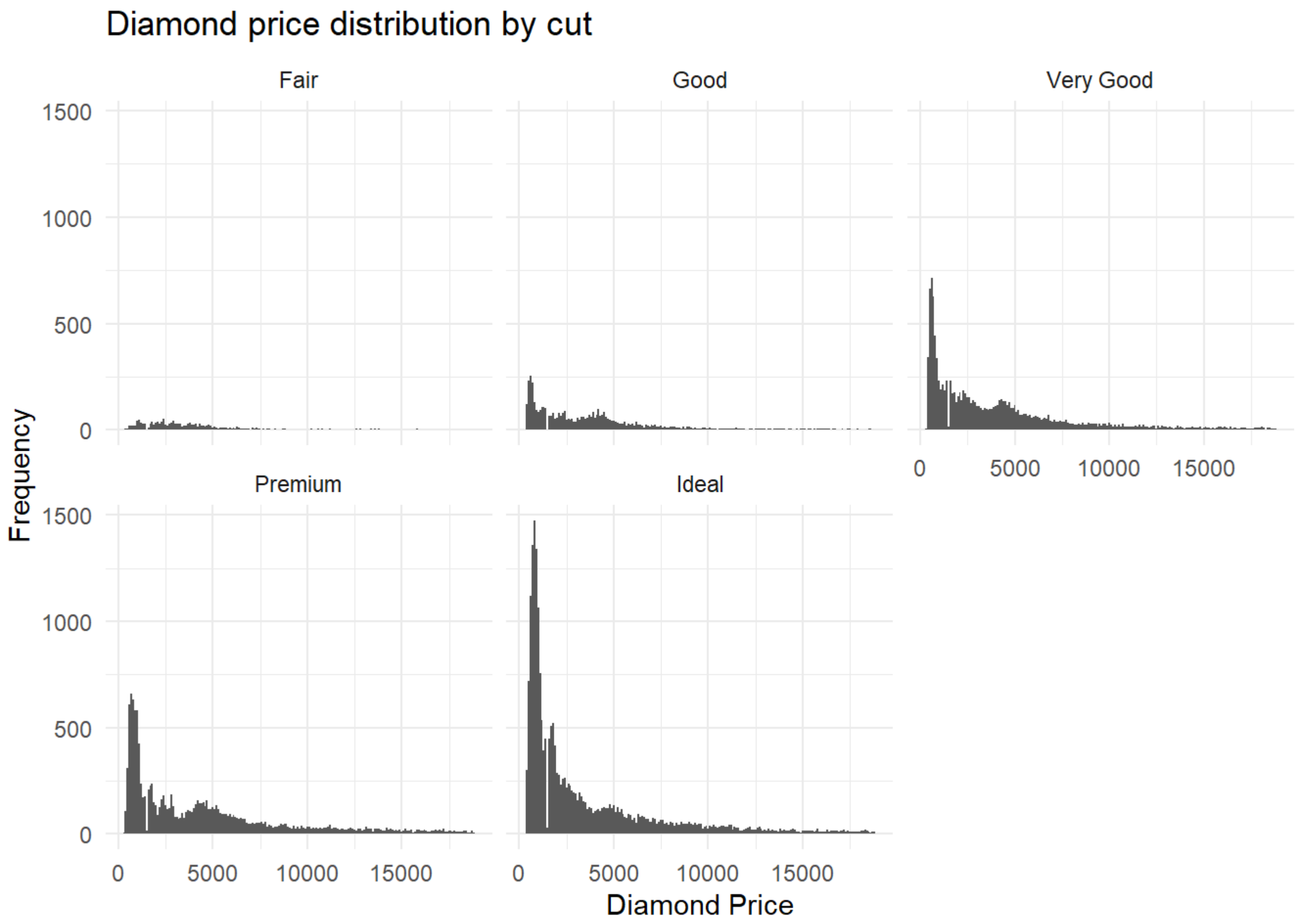
```
ggplot(data=diamonds) +
  geom_histogram(binwidth=50, aes(x=diamonds$price)) +
  ggtitle("Diamond price distribution") +
  xlab("Diamond Price") +
  ylab("Frequency") +
  theme_minimal() +
  xlim(0,2500)
```

```
## Warning: Removed 26398 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



```
ggplot(data=diamonds) +
  ggtitle("Diamond price distribution by cut") +
  xlab("Diamond Price") +
  ylab("Frequency") +
  theme_minimal() +
  geom_histogram(binwidth=100, aes(x=diamonds$price)) +
  facet_wrap(~cut)
```



```
subset(diamonds, price == max(price))
```

```
## # A tibble: 1 x 10
##   carat cut     color clarity depth table price     x     y     z
##   <dbl> <ord>   <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  2.29 Premium I     VS2      60.8    60 18823   8.5  8.47  5.16
```

```
subset(diamonds, price == min(price))
```

```
## # A tibble: 2 x 10
##   carat cut     color clarity depth table price     x     y     z
##   <dbl> <ord>   <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  0.23 Ideal   E     SI2      61.5    55   326  3.95  3.98  2.43
## 2  0.21 Premium E     SI1      59.8    61   326  3.89  3.84  2.31
```

```
a = diamonds[which(diamonds$cut == "Fair"),]
b = diamonds[which(diamonds$cut == "Good"),]
c = diamonds[which(diamonds$cut == "Very Good"),]
d = diamonds[which(diamonds$cut == "Premium"),]
e = diamonds[which(diamonds$cut == "Ideal"),]

median(a$price)
```

```
## [1] 3282
```

```
median(b$price)
```

```
## [1] 3050.5
```

```
median(c$price)
```

```
## [1] 2648
```

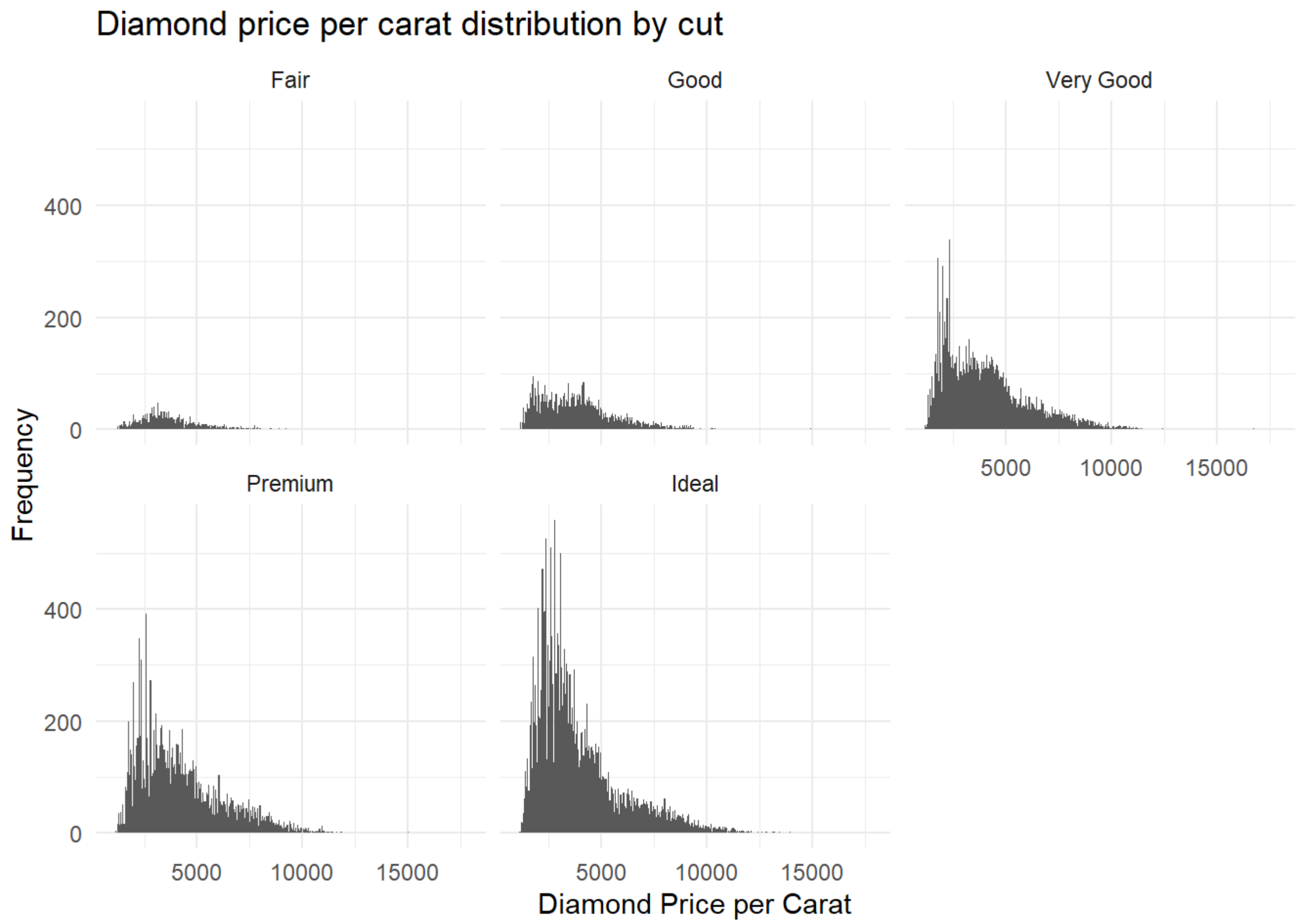
```
median(d$price)
```

```
## [1] 3185
```

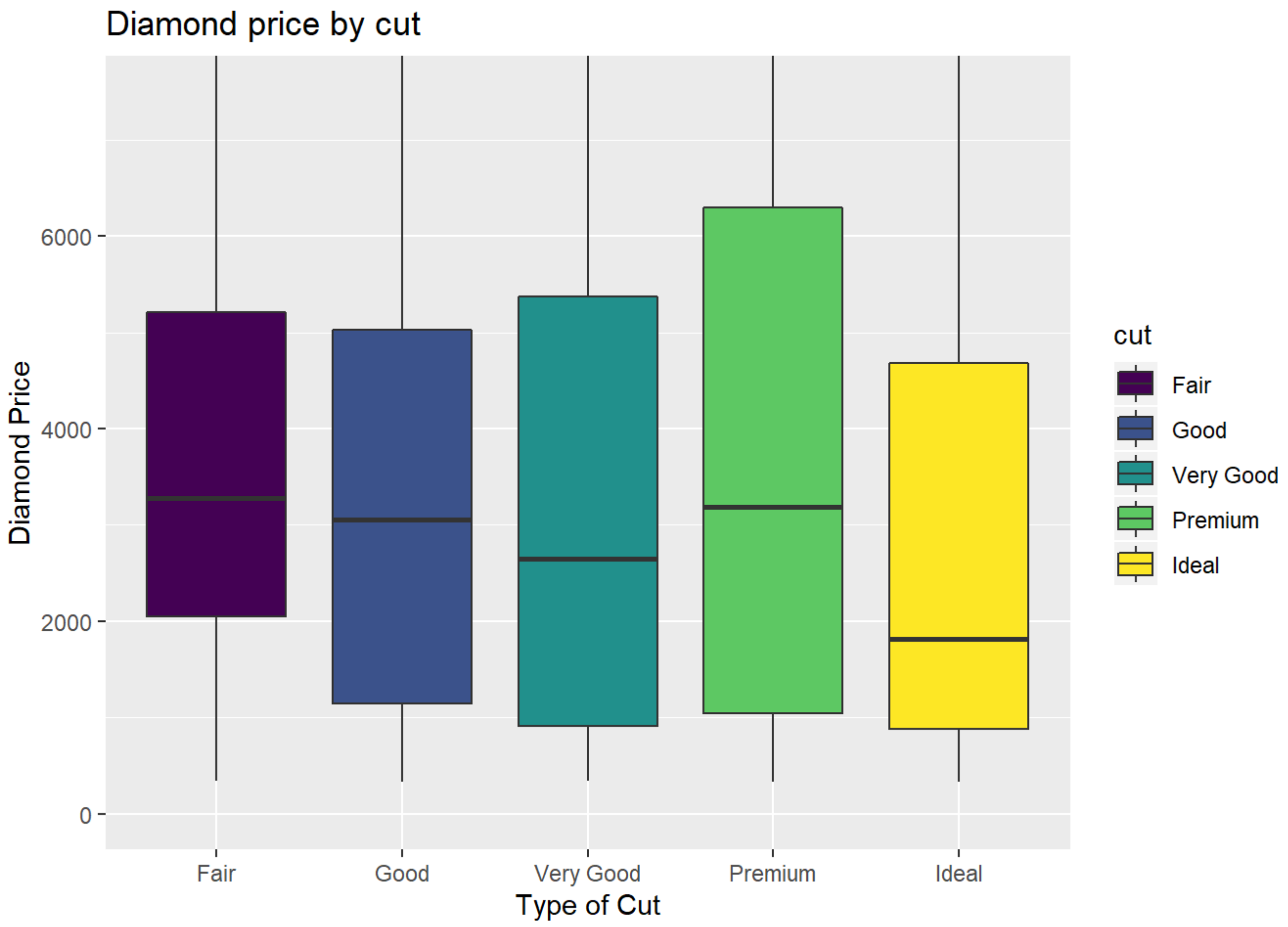
```
median(e$price)
```

```
## [1] 1810
```

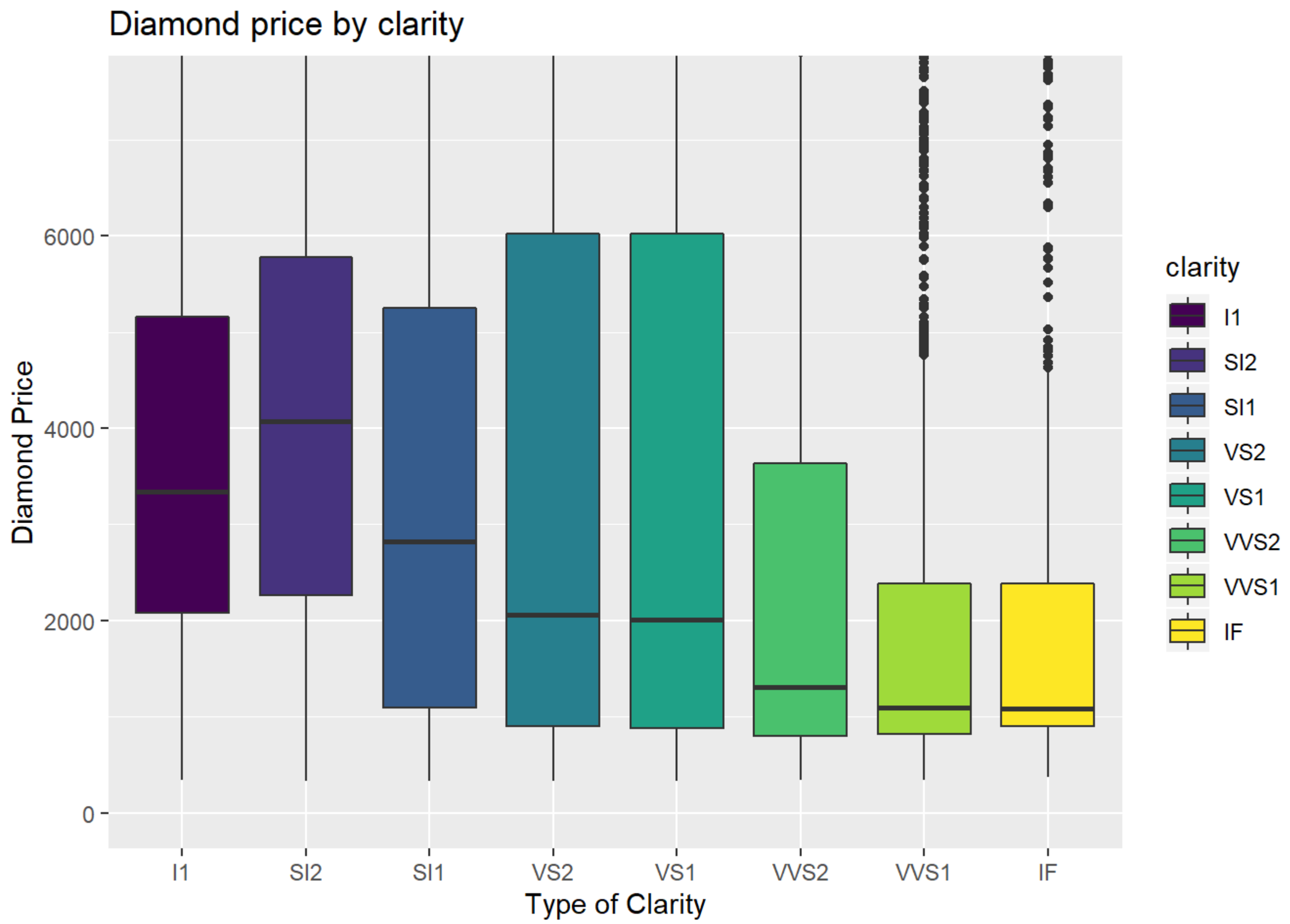
```
ggplot(data=diamonds) +
  geom_histogram(binwidth=50, aes(x=diamonds$price/diamonds$carat)) +
  ggtitle("Diamond price per carat distribution by cut") +
  xlab("Diamond Price per Carat") +
  ylab("Frequency") + theme_minimal() +
  facet_wrap(~cut)
```



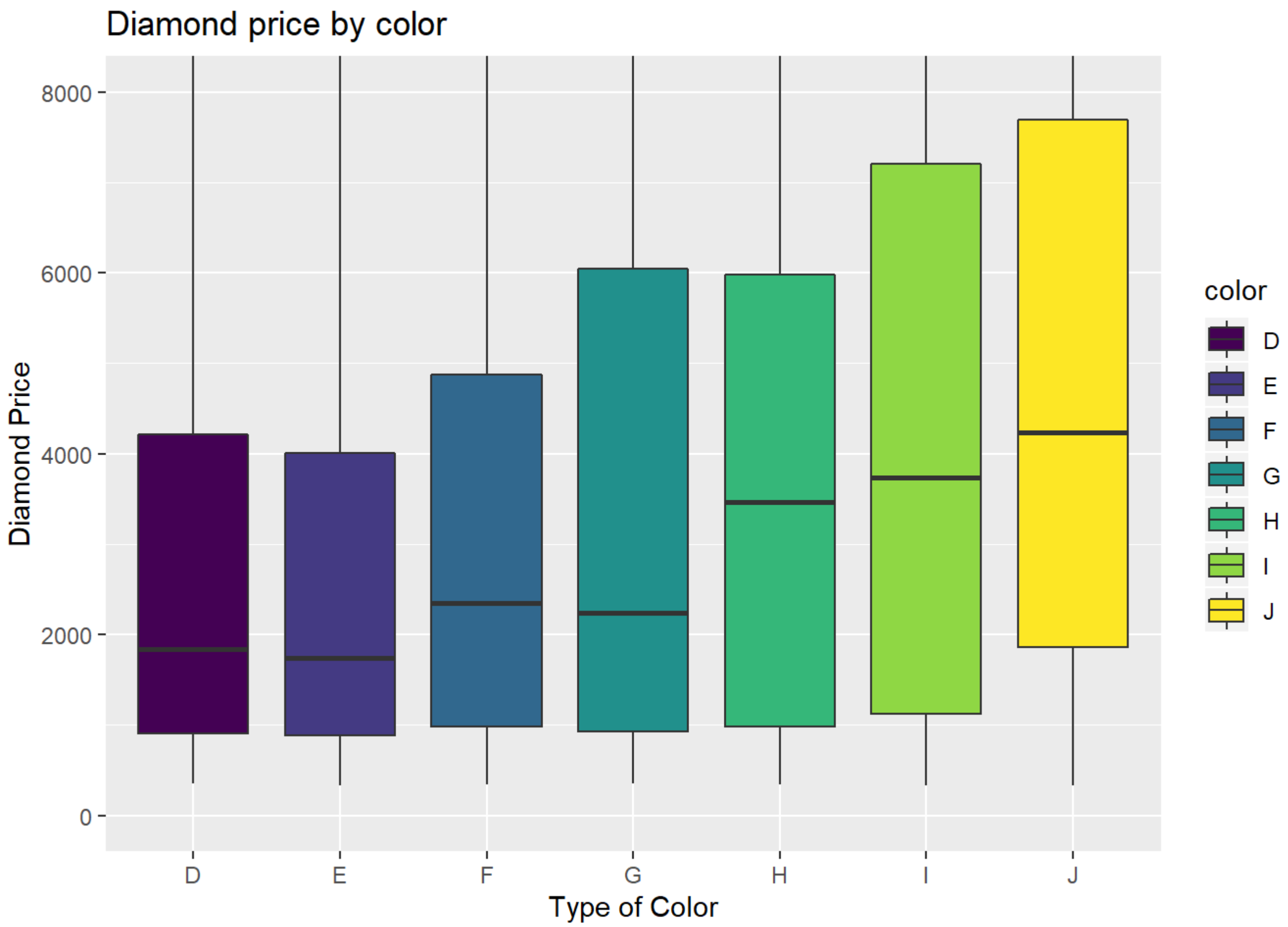
```
ggplot(diamonds, aes(factor(cut), price, fill=cut)) +
  geom_boxplot() + ggtitle("Diamond price by cut") +
  xlab("Type of Cut") +
  ylab("Diamond Price") +
  coord_cartesian(ylim=c(0,7500))
```



```
ggplot(diamonds, aes(factor(clarity), price, fill=clarity)) +
  geom_boxplot() + ggtitle("Diamond price by clarity") +
  xlab("Type of Clarity") +
  ylab("Diamond Price") +
  coord_cartesian(ylim=c(0,7500))
```

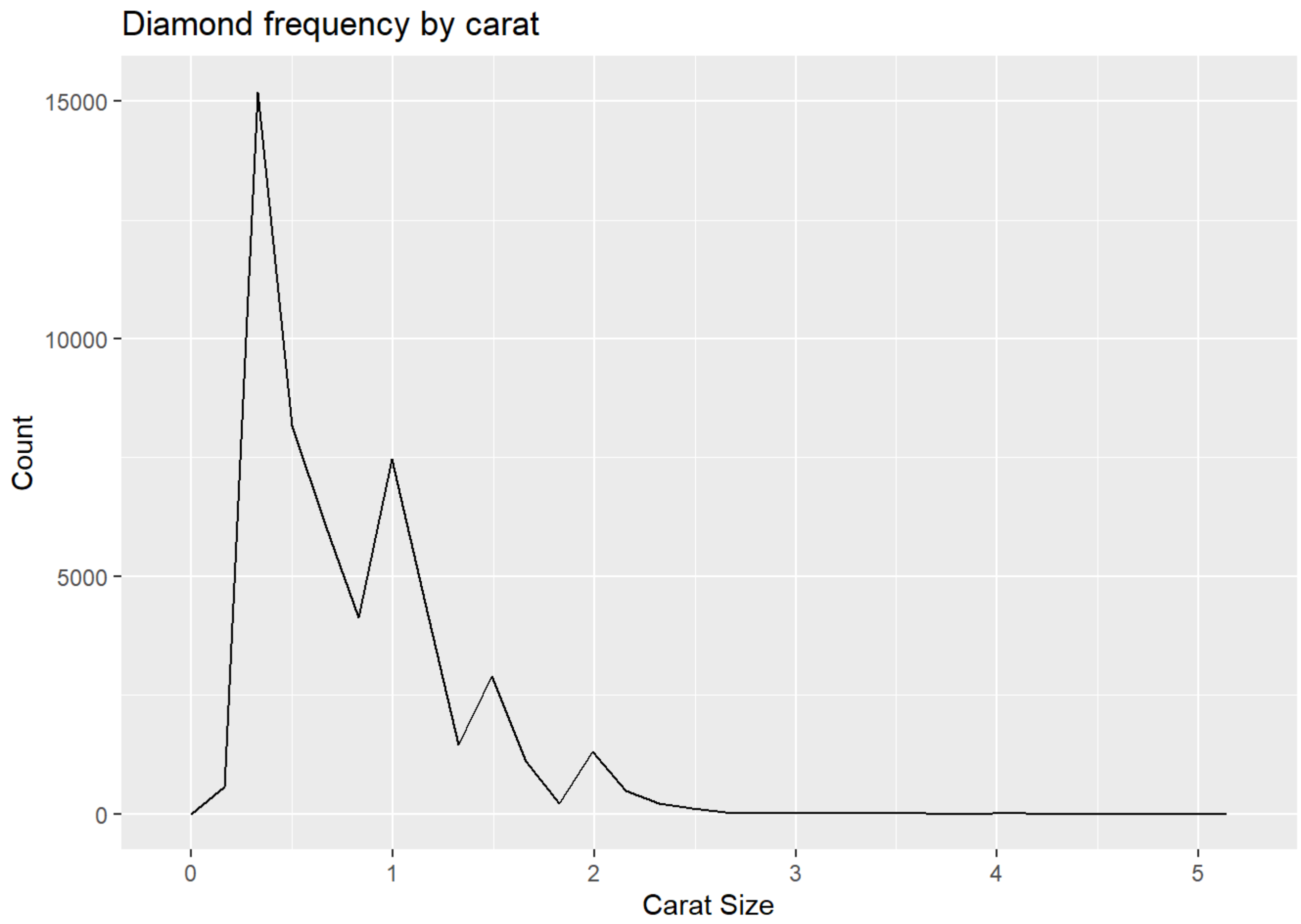


```
ggplot(diamonds, aes(factor(color), price, fill=color)) +
  geom_boxplot() + ggtitle("Diamond price by color") +
  xlab("Type of Color") +
  ylab("Diamond Price") +
  coord_cartesian(ylim=c(0,8000))
```



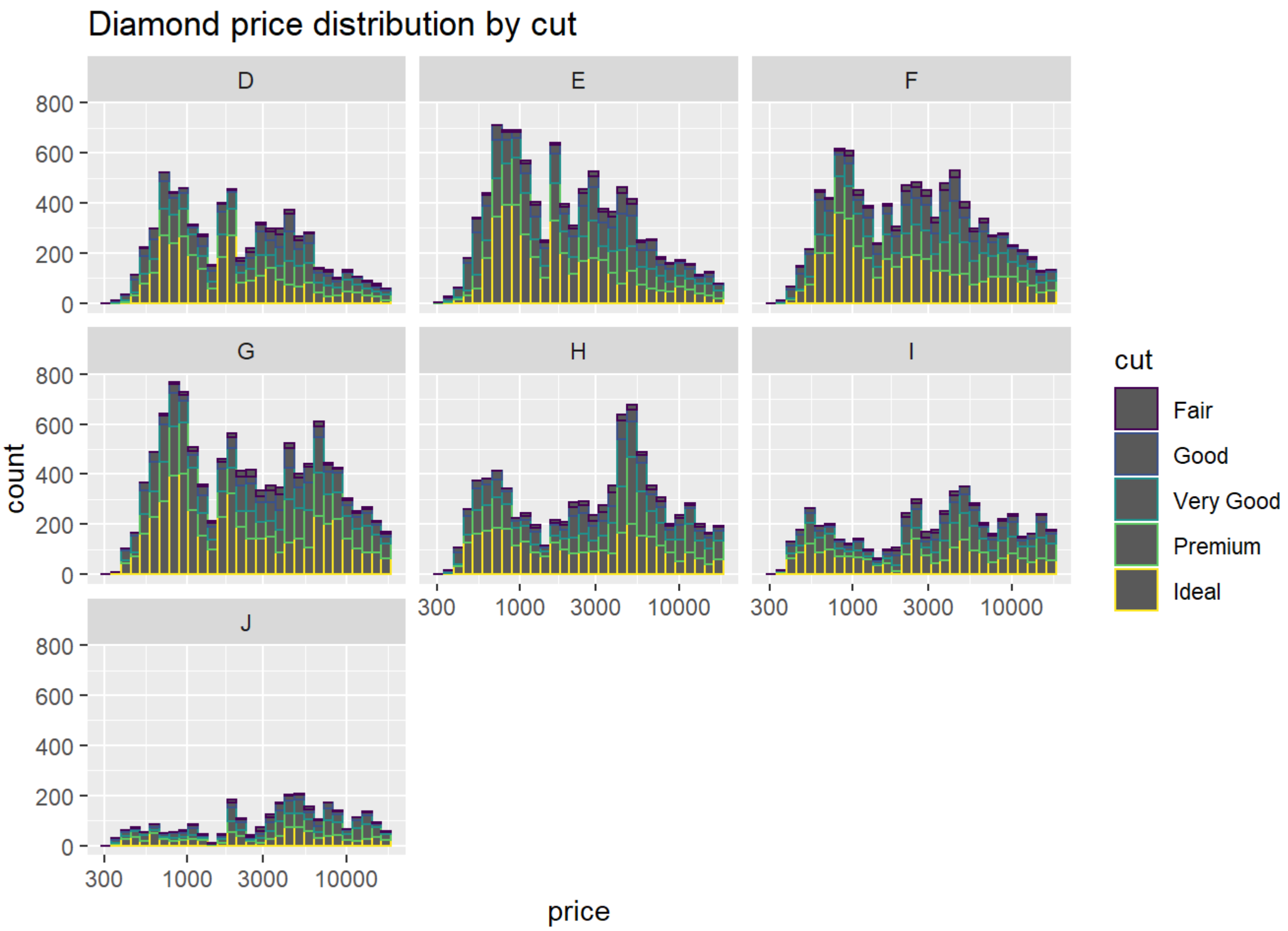
```
ggplot(data=diamonds, aes(x=carat)) +
  geom_freqpoly() +
  ggtitle("Diamond frequency by carat") +
  xlab("Carat Size") +
  ylab("Count")
```

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



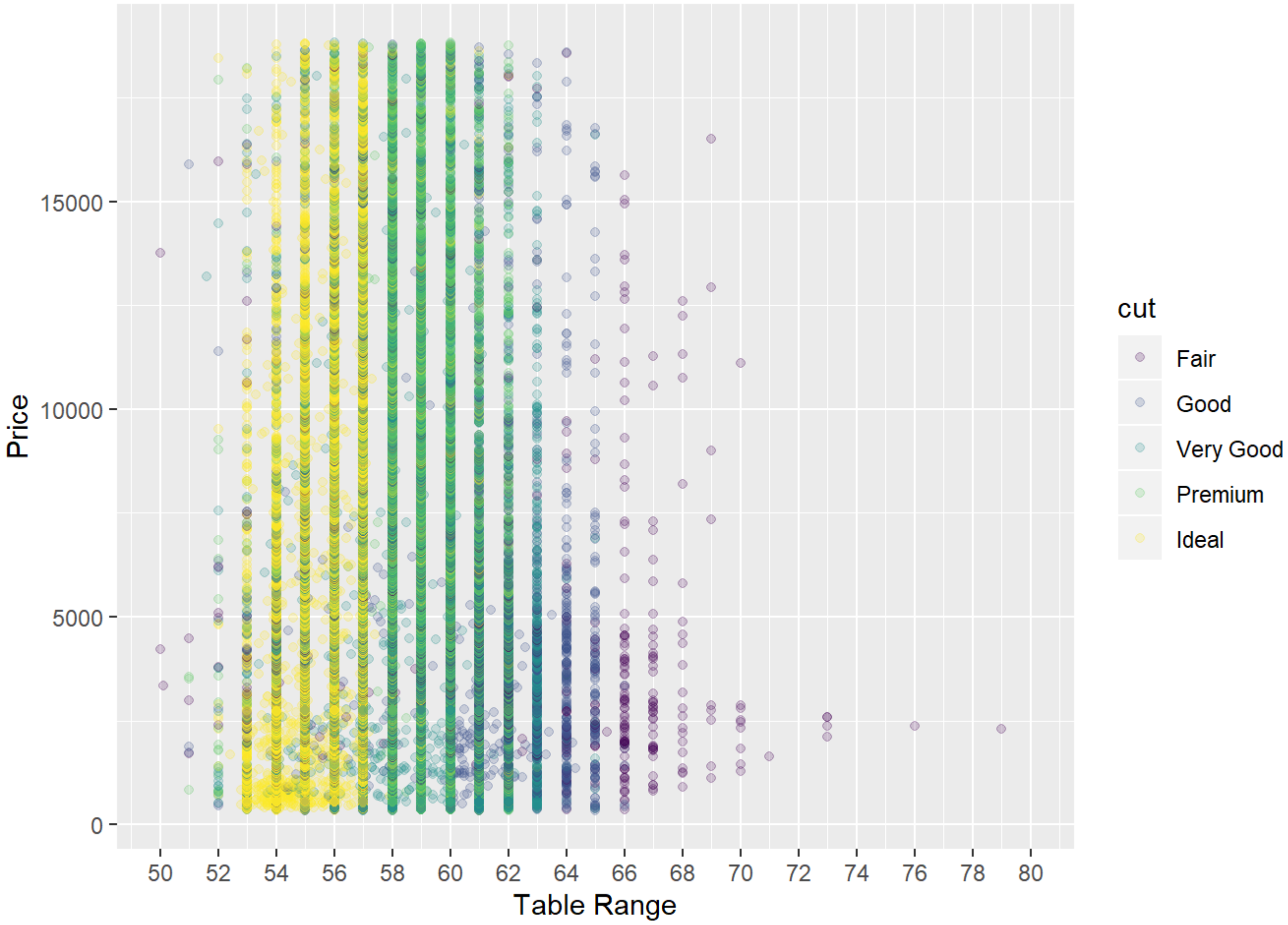
```
ggplot(aes(x = price, color = cut), data = diamonds) +
  facet_wrap(~color, ncol = 3) +
  geom_histogram() +
  scale_x_log10() +
  scale_fill_brewer(type = 'qual') +
  ggtitle("Diamond price distribution by cut")
```

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



```
ggplot(data = diamonds, aes(x = table, y = price, color = cut)) +
  geom_point(alpha = 1/5) +
  scale_x_continuous(limits = c(50, 80), breaks = seq(50, 80, 2)) +
  xlab("Table Range") +
  ylab("Price")
```

```
## Warning: Removed 5 rows containing missing values (geom_point).
```

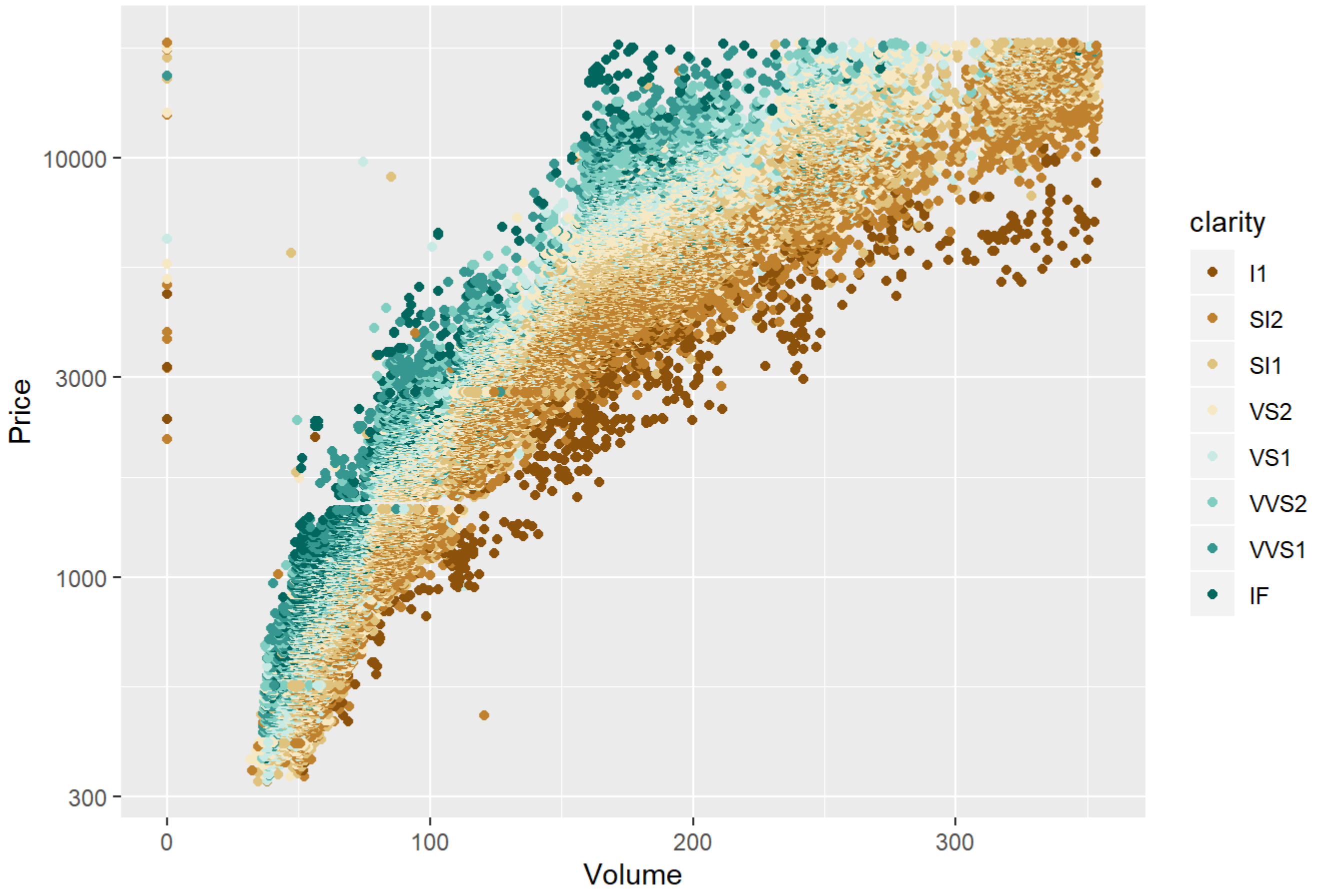


```
diamonds$volume <- with(diamonds, x * y * z)

ggplot(data = diamonds, aes(x = volume, y = price, color = clarity)) +
  geom_point() +
  scale_color_brewer(type = 'div') +
  scale_y_log10() +
  scale_x_continuous(limits = c(0, quantile(diamonds$volume, 0.99)))  +
  ggtitle("Diamond price by volume grouped by clarity") +
  xlab("Volume") +
  ylab("Price")
```

```
## Warning: Removed 540 rows containing missing values (geom_point).
```

Diamond price by volume grouped by clarity


Price
Volume
clarity
● I1
● SI2
● SI1
● VS2
● VS1
● VVS2
● VVS1
● IF