Exploration Diamonds Data Set

Finn Burgemeister und Tobias Machnitzki

1.

First we want to load the data and get an overview of what we are dealing with.

```
library(ggplot2)
library(scales)
library(dplyr)
```

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

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

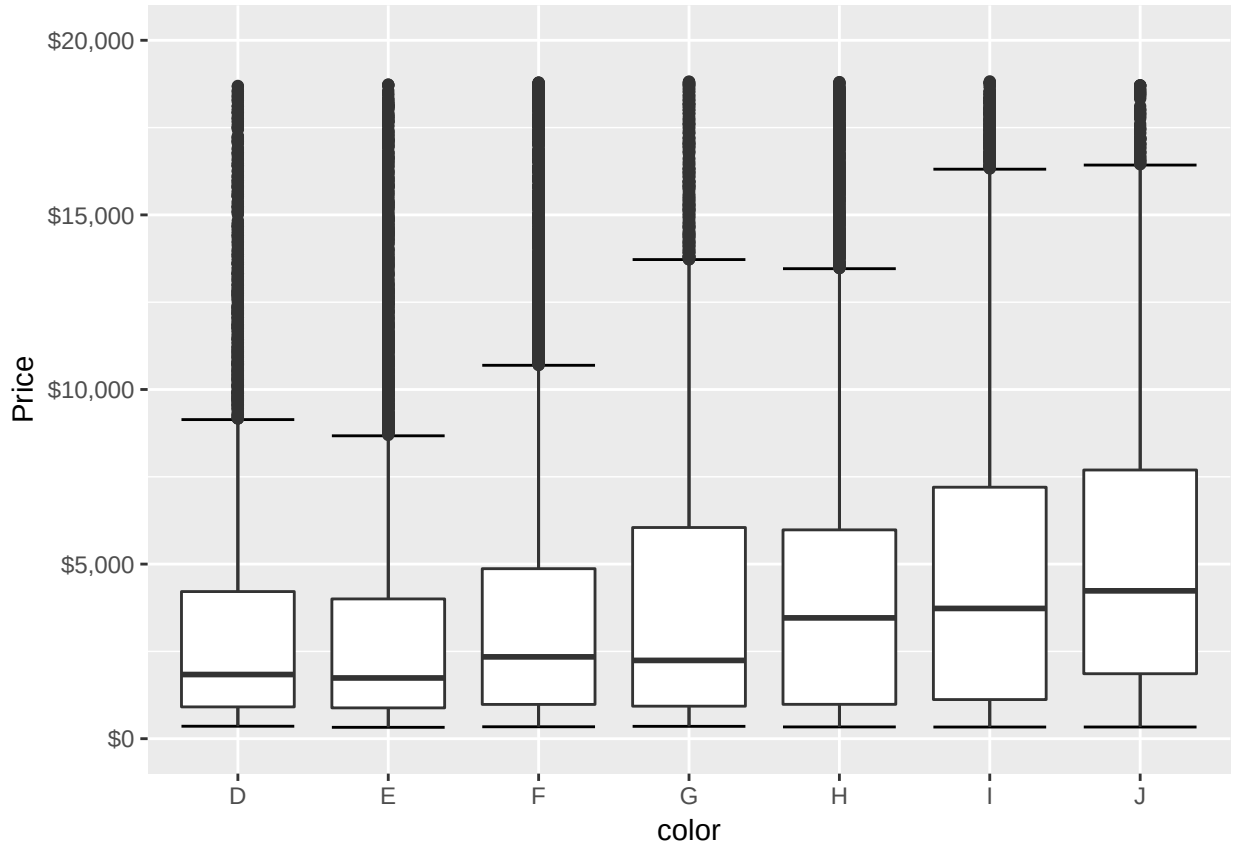
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(knitr)
library(markdown)
data(diamonds)
summary(diamonds)
```

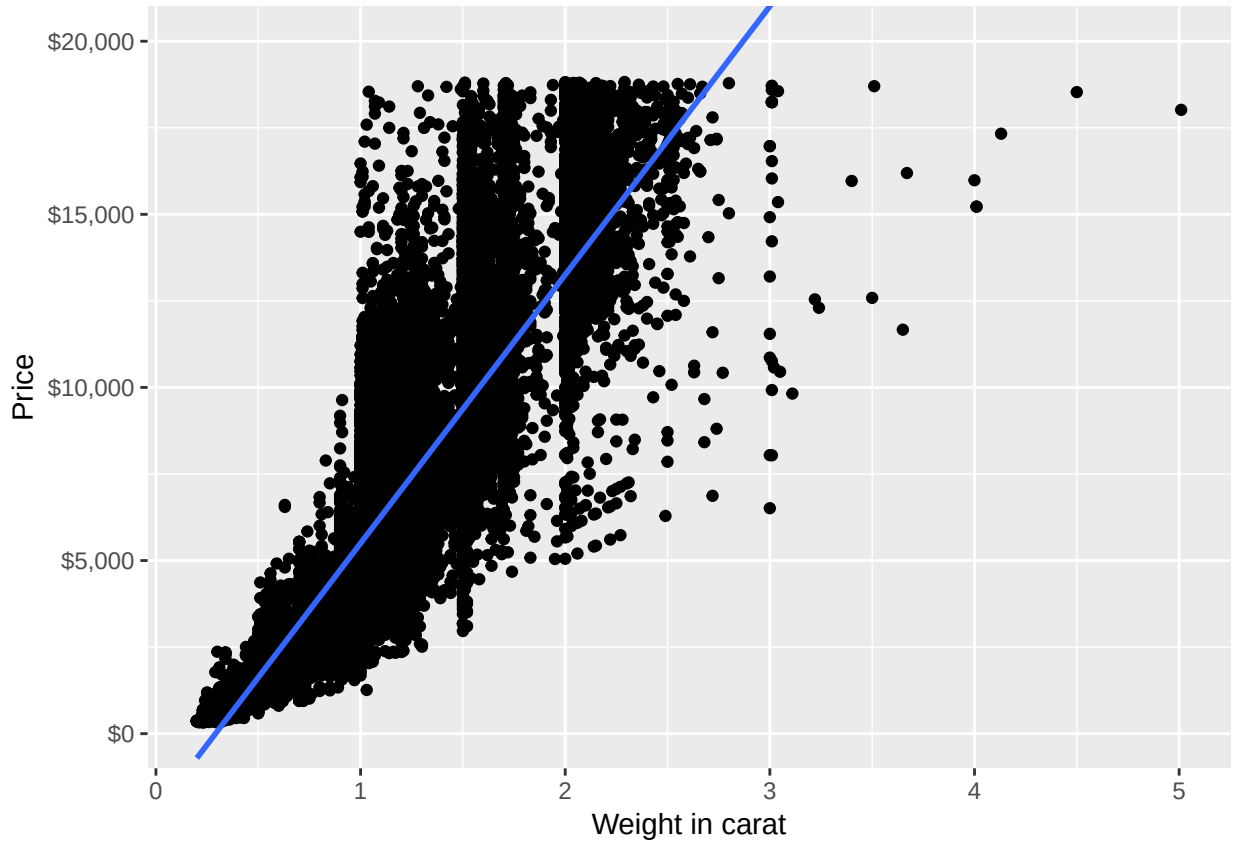
```
##      carat               cut        color        clarity
##  Min.   :0.2000   Fair      : 1610   D: 6775   SI1    :13065
##  1st Qu.:0.4000   Good      : 4906   E: 9797   VS2    :12258
##  Median :0.7000   Very Good:12082   F: 9542   SI2    : 9194
##  Mean   :0.7979   Premium   :13791   G:11292   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
##
```

The first thing we notice is that cut, color and clarity do not contain numbers for evaluation, but some kind of tags. So lets look at some combinations:

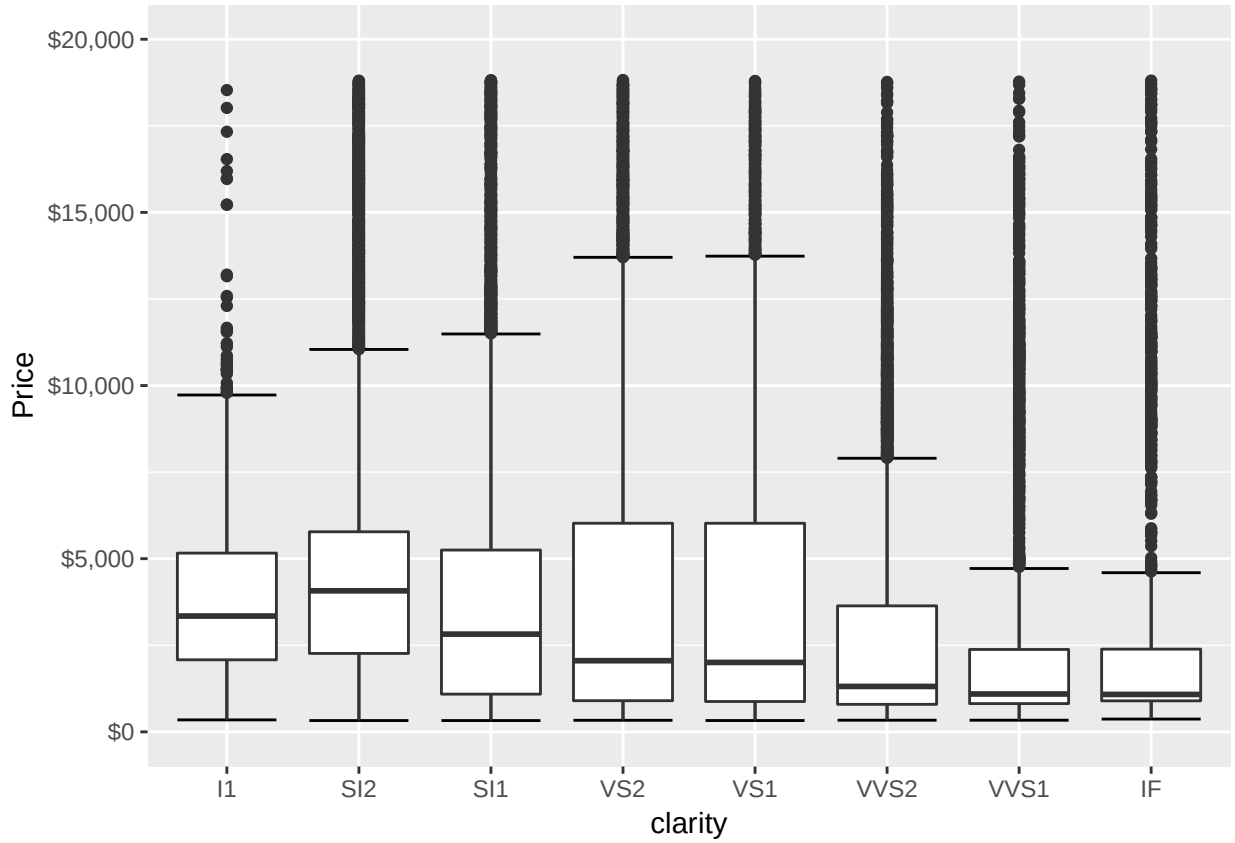
```
ggplot(diamonds, aes(x = color, y = price)) +
  stat_boxplot(geom="errorbar") +
  geom_boxplot() +
  coord_cartesian(ylim=c(0, 20000)) +
  scale_y_continuous(labels=dollar) +
  xlab("color") + ylab("Price")
```



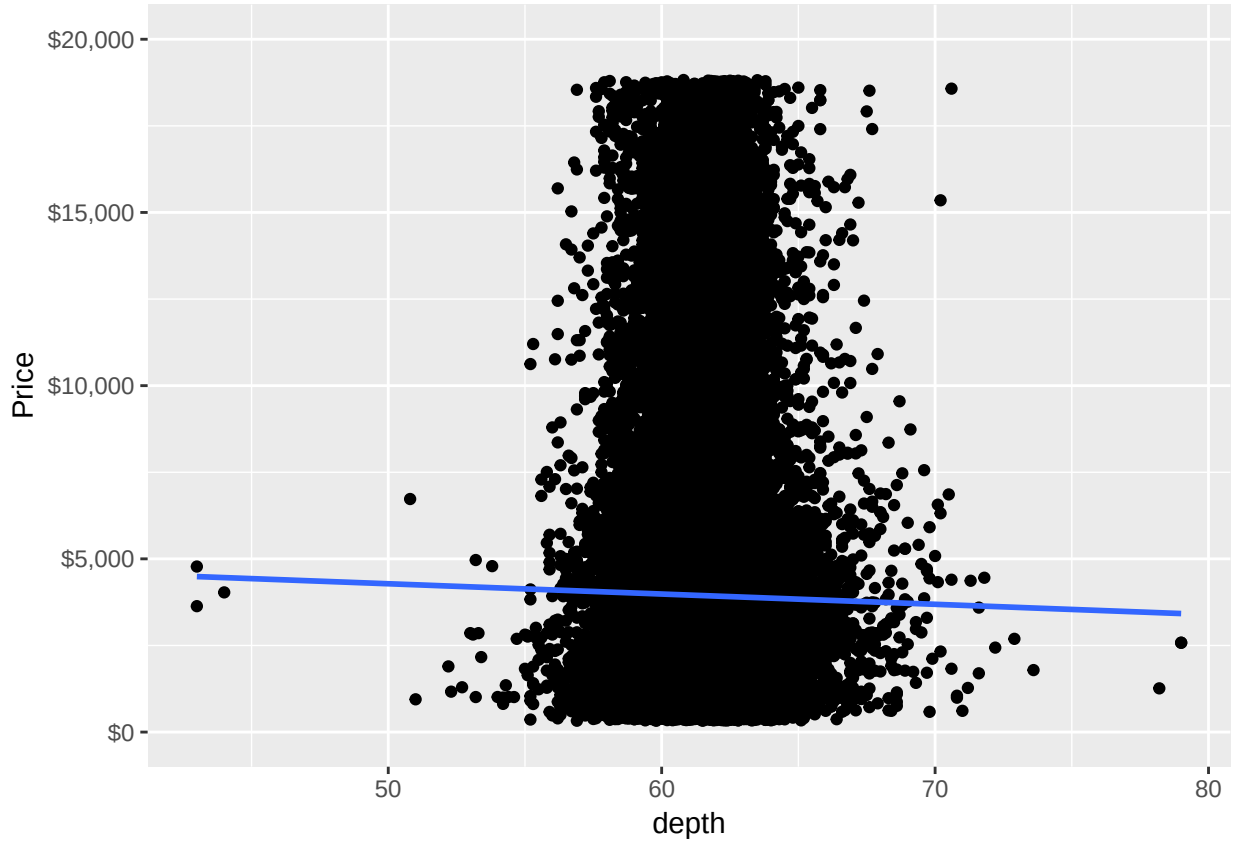

```
ggplot(diamonds, aes(x = carat, y = price)) +
  geom_point() +
  geom_smooth(method=lm,   # Add linear regression line
              se=FALSE) +   # Don't add shaded confidence region
  scale_y_continuous(labels=dollar) +
  coord_cartesian(ylim=c(0, 20000)) +
  xlab("Weight in carat") + ylab("Price")
```




```
ggplot(diamonds, aes(x = clarity, y = price)) +
  stat_boxplot(geom="errorbar") +
  geom_boxplot() +
  coord_cartesian(ylim=c(0, 20000)) +
  scale_y_continuous(labels=dollar) +
  xlab("clarity") + ylab("Price")
```




```
ggplot(diamonds, aes(x = depth, y = price)) +
  geom_point() +
  geom_smooth(method=lm,   # Add linear regression line
              se=FALSE) +   # Don't add shaded confidence region
  scale_y_continuous(labels=dollar) +
  coord_cartesian(ylim=c(0, 20000)) +
  xlab("depth") + ylab("Price")
```

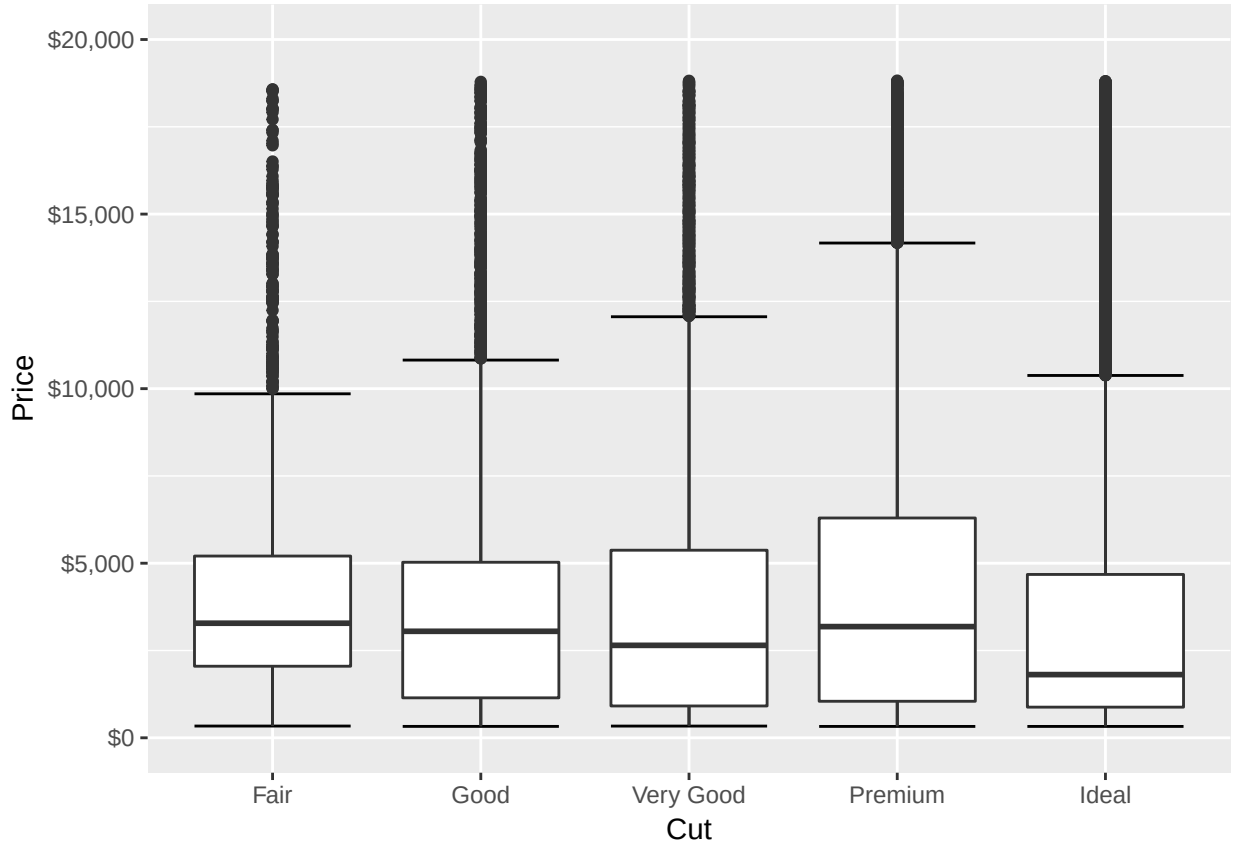


The conclusion of the plots above is, that the price does correlates very well with the weight and the color. On the other hand it does not correlate at all with the size (depth) and the clarity.

2.

Furthermore we want to know, how the Price correlates with the cut.

```
ggplot(diamonds, aes(x = cut, y = price)) +
  stat_boxplot(geom="errorbar") +
  geom_boxplot() +
  coord_cartesian(ylim=c(0, 20000)) +
  scale_y_continuous(labels=dollar) +
  xlab("Cut") + ylab("Price")
```



This shows quite easaly, that the Premium cut in average has a much higher price than an ideal cutted diamond.

Lets have a look at some numbers:

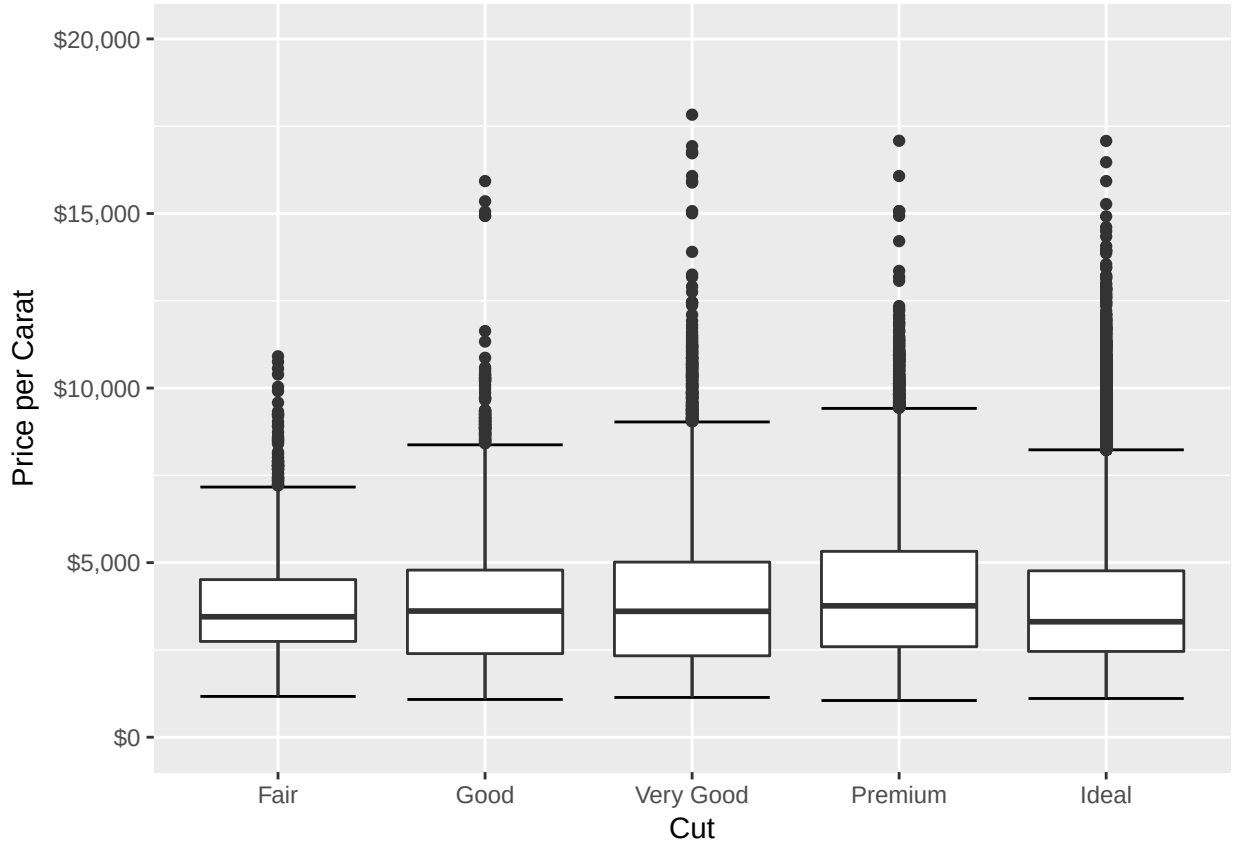
```
diamonds %>%
  group_by(cut) %>%
  summarise(max_price = max(price),
            min_price = min(price),
            median_price = median(price))
```

```
## # A tibble: 5 x 4
##         cut max_price min_price median_price
##       <ord>     <dbl>     <dbl>        <dbl>
## 1      Fair     18574       337       3282.0
## 2      Good     18788       327       3050.5
## 3 Very Good     18818       336       2648.0
## 4   Premium     18823       326       3185.0
## 5     Ideal     18806       326       1810.0
```

The maximum price of each cut category is very similar,as well as the minimum price. This is due to the fact, that we look at comlete diamonds here, which price depends most on size. So even if the cut is ideal, if it is a very small diamond it still will be cheap. To see the real effect of the shape on the price we need to use a new category: price/carat.

```
ggplot(diamonds, aes(x = cut, y = price/carat)) +
  stat_boxplot(geom="errorbar") +
  geom_boxplot() +
  coord_cartesian(ylim=c(0, 20000)) +
```

```
  scale_y_continuous(labels=dollar) +
  xlab("Cut") + ylab("Price per Carat")
```




```
diamonds %>%
  group_by(cut) %>%
  summarise(max_price_per_carat = max(price/carat),
            min_price_per_carat = min(price/carat),
            median_price_per_carat = median(price/carat))
```

```
## # A tibble: 5 x 4
##         cut max_price_per_carat min_price_per_carat median_price_per_carat
##       <ord>               <dbl>               <dbl>                  <dbl>
## 1      Fair            10909.33            1168.000               3449.444
## 2      Good            15928.00            1080.645               3613.250
## 3 Very Good            17828.85            1138.710               3605.826
## 4   Premium            17083.18            1051.163               3763.333
## 5     Ideal            17077.67            1109.091               3307.143
```

Now we can see, that a fair cut drives down the maximum price you can get, but better than very good does not really makes a huge difference on th price. In fact the Very Good cut category has not only the highest priced diamond, but as well higher median, than the Ideal cutted diamonds.