CSE 6242 Assignment 2

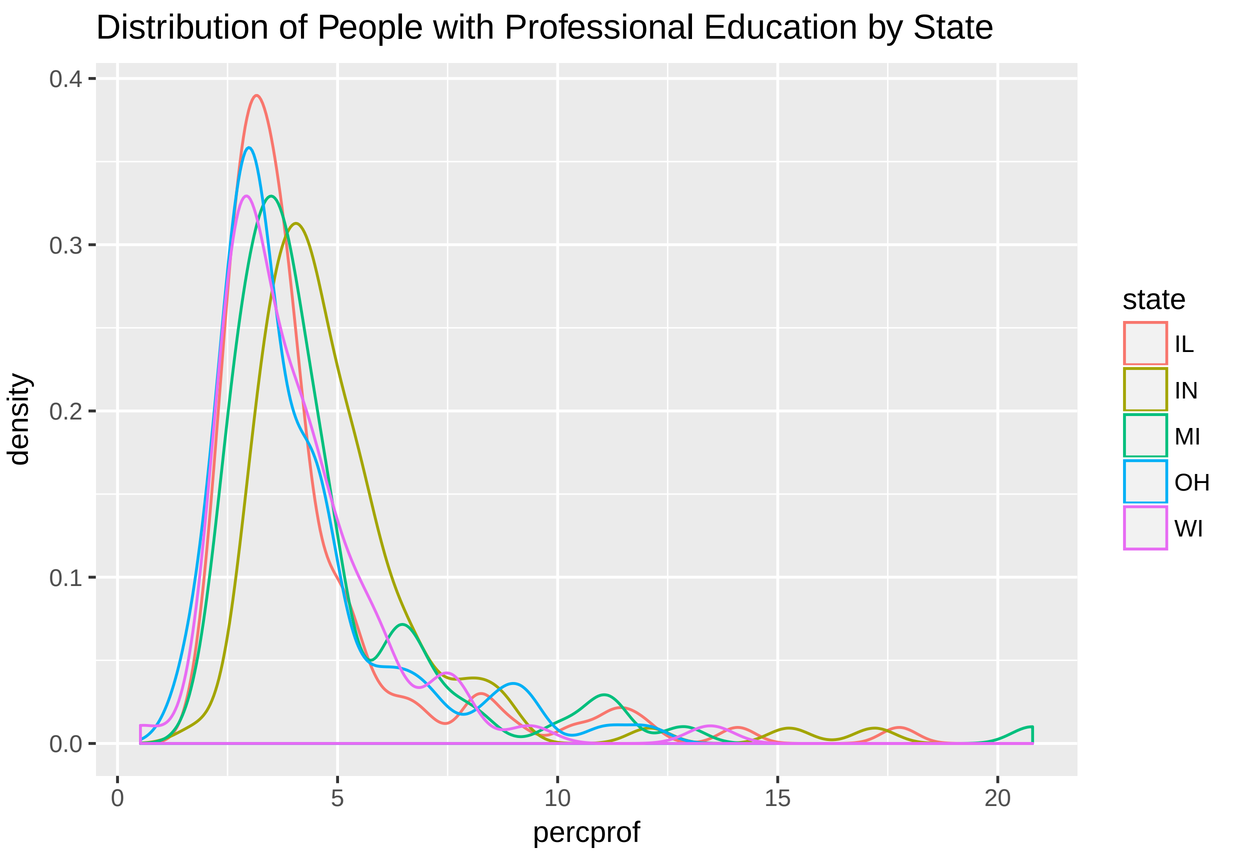
Vincent La (Georgia Tech ID - vla6)

September 12, 2017

Question 1: Professional Education by State

We first create a density plot grouped by state to show the distribution of `percprof`, people with a professional education for each county, grouped by state.

```
ggplot(df, aes(x=percprof)) +
  geom_density(aes(color=state, group=state)) +
  ggtitle('Distribution of People with Professional Education by State')
```



We can see from the density above that for each state, the distribution of `percprof` is right-tailed. This makes sense as it's bounded at 0, and there are a few counties where the proportion of people with professional education is very high. However, the median percentage of people with professional education by county for each state is under 5. We calculate the median for each state explicitly below:

```
summary_stats = tapply(df$percprof, df$state, summary)
print(summary_stats)
```

```
## $IL
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.945   2.935   3.455   4.315   4.455  17.757
##
```

```
## $IN
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.793   3.796   4.440   5.045   5.524  17.201
##
## $MI
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.006   3.251   3.828   4.686   4.966  20.791
##
## $OH
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.565   2.824   3.328   4.080   4.567  12.045
##
## $WI
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.5203  2.8219  3.4951  4.0448  4.6516 13.4715
```

Notice that Indiana has the highest median at 4.4401272. Ohio has the lowest median at 3.3280118.

A second interesting note is that the range varies by state as well. While Michigan has a lowwer median than Indiana, it has the highest maximum and a large range. Wisconsin, on the other hand, has a relatively low range.

The state with the lowest and highest percentage of population with a professional education are Ohio and Indiana, respectively, looking strictly at medians. We can also revisit this by comparing distributions directly.

Question 2: School and College Education by State

To compare `perchsd` and `percollege`, percentage of population with high school diploma and college education, respectively with state, we will again use density plots.

To compare `percsd` to `percollege` directly, we will use a scatter plot, grouped at the state level to see if there are any state level trends.

```
df = midwest[, c('PID', 'county', 'state', 'perchsd', 'percollege')]
print(df)
```

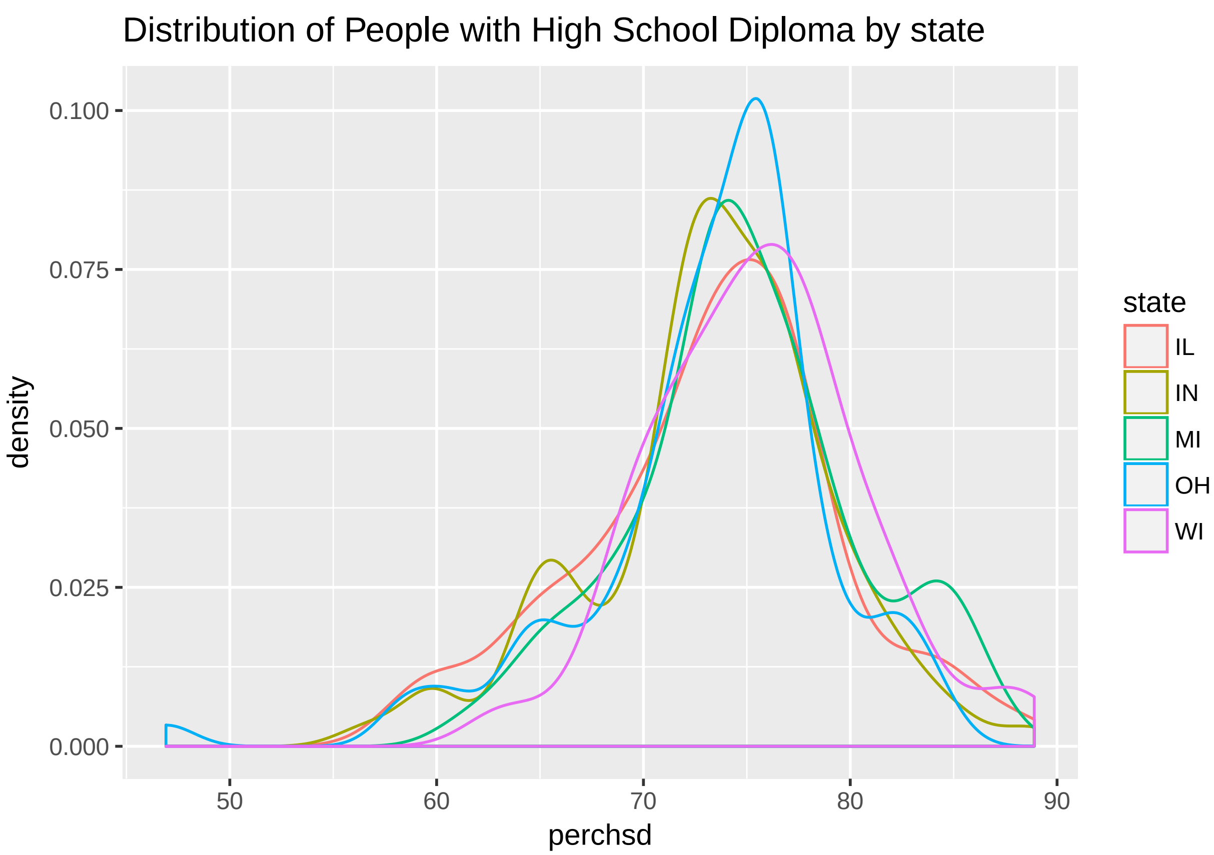
```
## # A tibble: 437 x 5
##      PID    county state  perchsd percollege
##    <int>     <chr> <chr>    <dbl>      <dbl>
##  1   561     ADAMS    IL 75.10740   19.63139
##  2   562 ALEXANDER    IL 59.72635   11.24331
##  3   563      BOND    IL 69.33499   17.03382
##  4   564     BOONE    IL 75.47219   17.27895
##  5   565     BROWN    IL 68.86152   14.47600
##  6   566    BUREAU    IL 76.62941   18.90462
##  7   567   CALHOUN    IL 62.82445   11.91739
##  8   568   CARROLL    IL 75.95160   16.19712
##  9   569      CASS    IL 72.27195   14.10765
## 10   570 CHAMPAIGN    IL 87.49935   41.29581
## # ... with 427 more rows
```

```
summary(df)
```

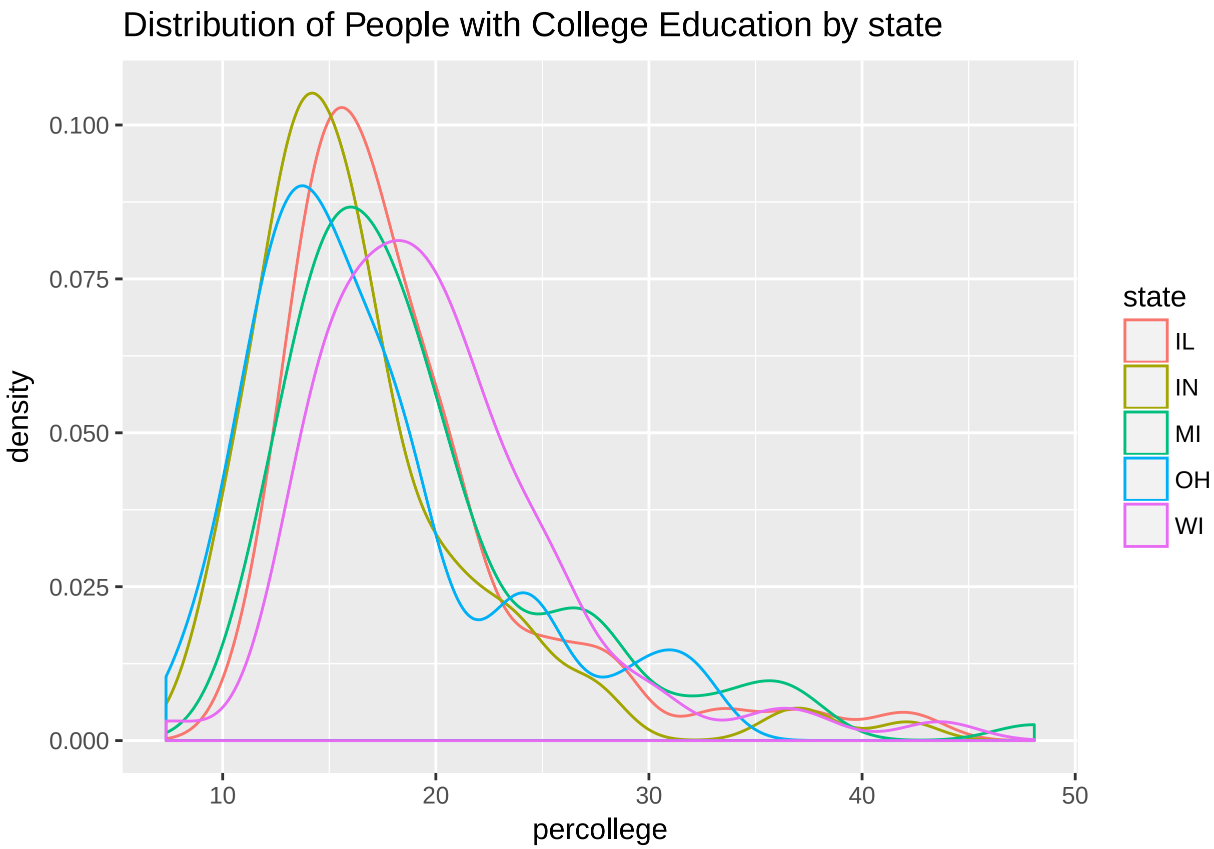
```
##       PID          county              state              perchsd
##  Min.   : 561   Length:437         Length:437         Min.   :46.91
##  1st Qu.: 670   Class :character   Class :character   1st Qu.:71.33
```

```
##  Median :1221   Mode  :character   Mode  :character   Median :74.25
##  Mean   :1437                                         Mean   :73.97
##  3rd Qu.:2059                                         3rd Qu.:77.20
##  Max.   :3052                                         Max.   :88.90
##    percollege
##  Min.   : 7.336
##  1st Qu.:14.114
##  Median :16.798
##  Mean   :18.273
##  3rd Qu.:20.550
##  Max.   :48.079
```

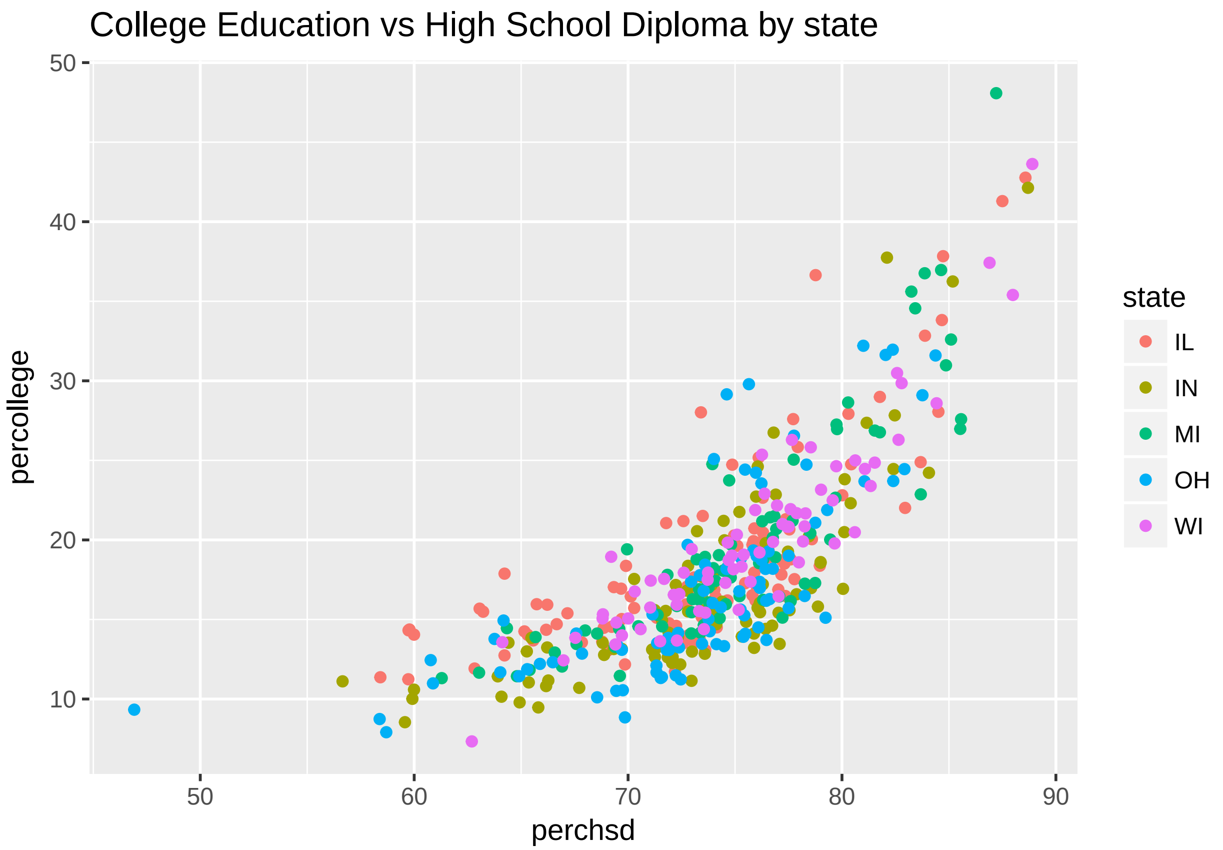
```
# Plotting perchsd density plots by state
ggplot(df, aes(x=perchsd)) +
  geom_density(aes(color=state, group=state)) +
  ggtitle('Distribution of People with High School Diploma by state')
```




```
# Plotting percollege density plots by state
ggplot(df, aes(x=percollege)) +
  geom_density(aes(color=state, group=state)) +
  ggtitle('Distribution of People with College Education by state')
```




```r
# Plotting scatter plot of perchsd vs percollege
# Plotting percollege density plots by state
ggplot(df, aes(x=perchsd, y=percollege)) +
  geom_point(aes(color=state, group=state)) +
  ggtitle('College Education vs High School Diploma by state')
```



In looking at the relationship between `perchsd` and `percollege` there seems to be very strong correlation between the two. This makes sense as counties with high proportion of people have a high school diploma may value education and thus those people will be more likely to complete a college education as well.

Another interesting note is that the distribution of people with high school diploma, `perchsd` almost seems somewhat normally distribution, with perhaps a slight left tail. On the other hand, the distribution of people with college education, `percollege` is more right skewed. This also makes sense since college education is more difficult to obtain.

Question 3: Comparison of Visualization Techniques

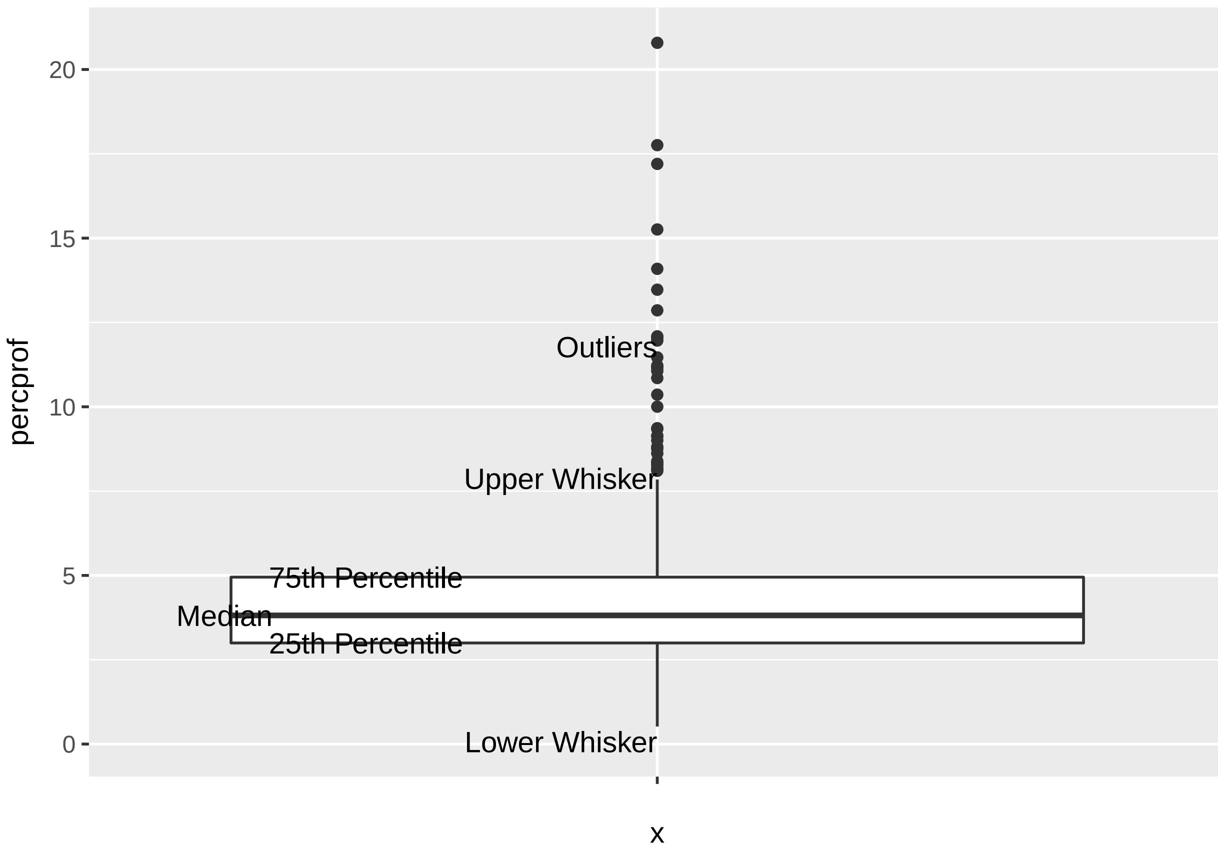
Box plots are like histograms in the sense that they also give a sense of the distribution about the data. However, box plots are considered more "lossy" in the sense that it doesn't provide as much information; but displays the key points, such as median, and other "important" quantiles and outliers that histograms are not as useful at displaying.

Histograms are also useful in the sense that they provide more detail than box plots. With a histogram, one can more easily see the "tails" of distributions. Furthermore, with a histogram, one can see exactly how many values fall into a specific bin.

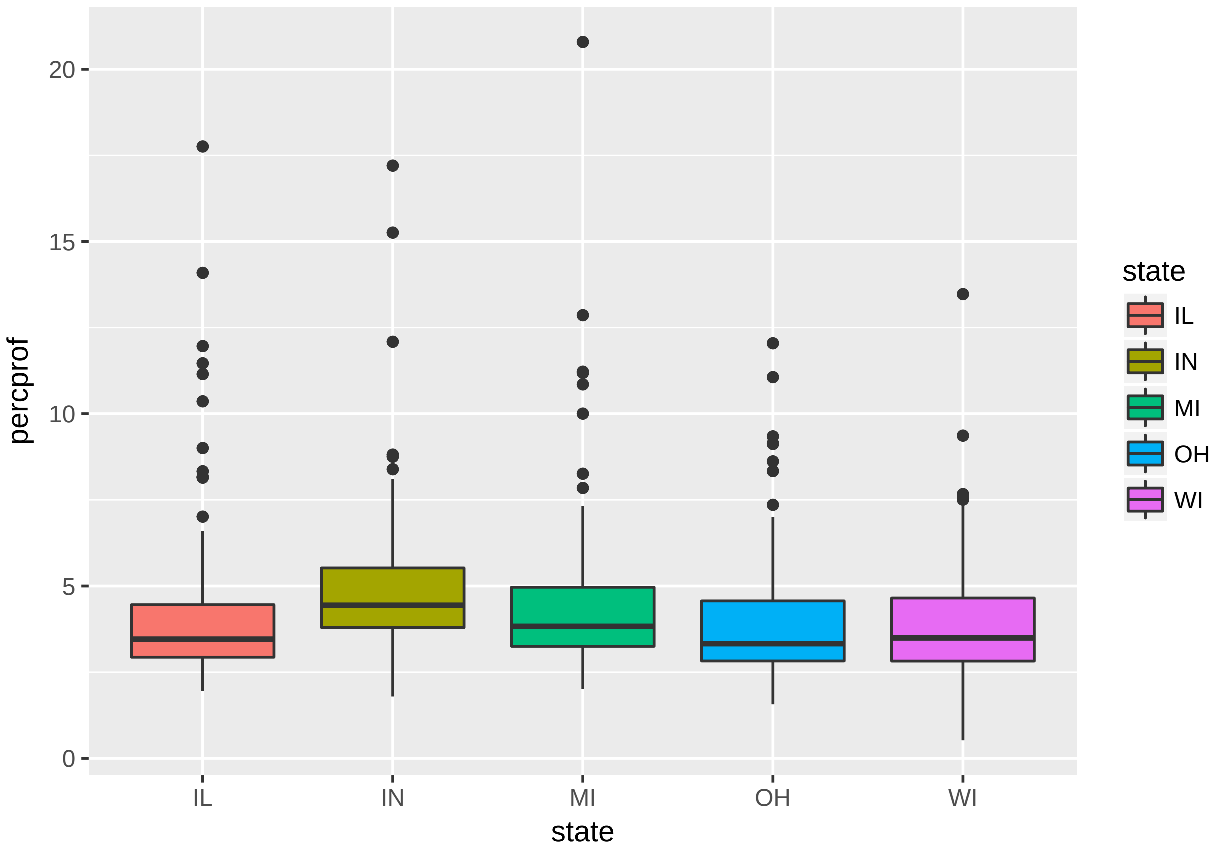
Finally, a QQ plot is a scatter plot of quantiles of one data set on the x axis and quantiles of another data set on the y axis. Sometimes, the x or y axis describes quantiles coming from theoretical distribution as opposed to specific data set. One specific use case of QQ plot using quanties coming from theoretical distribution is to compare to a normal distribution to see if a dataset is normal.

To illustrate a box plot, let's revisit the example from question 1, professional education by state, or `percprof`.

```r
df = midwest[, c('PID', 'county', 'state', 'percprof')]
quants = quantile(df$percprof, c(.25, .5, .75))
iqr = quants[3] - quants[1]

# Aggregate across all states
ggplot(midwest, aes(x='', y=percprof)) +
    geom_boxplot() +
  annotate("text", x = '', y = quants[1] - 1.5 * iqr, label = "Lower Whisker", hjust = 1) +
  annotate("text", x = '', y = quants[1], label = "25th Percentile", hjust = 2) +
  annotate("text", x = '', y = quants[2], label = "Median", hjust = 5) +
  annotate("text", x = '', y = quants[3], label = "75th Percentile", hjust = 2) +
  annotate("text", x = '', y = quants[3] + 1.5 * iqr, label = "Upper Whisker", hjust = 1) +
  annotate("text", x = '', y = quants[3] + 3.5 * iqr, label = "Outliers", hjust = 1)
```




```r
# Now group by each state, similar to question 1
ggplot(midwest, aes(x=state, y=percprof, fill=state)) +
  geom_boxplot()
```



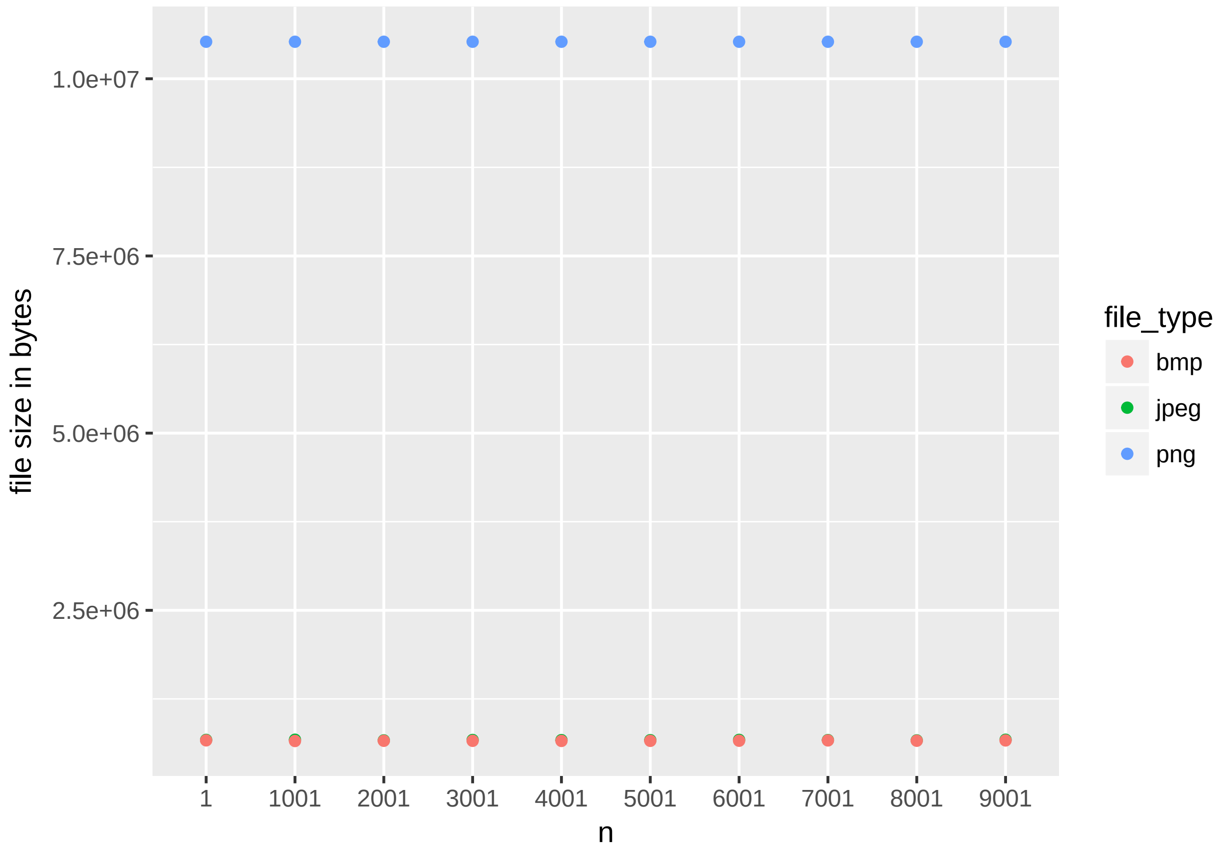
You can see for the box plots by each state, we see the 1st quartile (bottom of box), median (solid line in middle of box), and 3rd quartile (top of box). Furthermore, you can see outliers that lie outside the IQR in dots. You can compare these with the summary statistics presented in question 1.

Question 4: Random Scatterplots

```
plot_random_unif <- function(n){
#' Plots two vectors of size n drawn from a uniform distribution.
#' Saves image to './image' directory
#' Number of plots drawn equal to n, where loop from 1 to n
file_types = c('png', 'jpeg', 'bmp')
by_sequence = 1000
for(ft in file_types){
    for (i in seq(1, n, by = by_sequence)) {
        x = runif(n)
        y = runif(n)
        df = data.frame(x=x, y=y)

        ggplot(df, aes(x=x, y=y)) +
          geom_point()
          ggsave(paste0('./images/example_n', i, '.', ft), device=ft)
    }
}

# To extract file size use file.info() and then extract column size
```

```r
# file.info('filename.png')$size
file_type = rep(file_types, n/by_sequence)
file_name = c()
file_size = c()
num_points = c()
files = list.files(path='./images')

for (f in files){
    file_name = c(file_name, f)
    file_size = c(file_size, file.info(paste0('./images/', f))$size)
    num_points = c(num_points, sub(".*example_n *(.*?) *\\..*", "\\1", f)) # Extracts regex everything
}

df = data.frame(file_name=file_name, file_type=file_type, num_points=num_points, file_size=file_size)

# Print scatter plot of relationship between file size and num points grouped by file type
ggplot(df, aes(x=num_points, y=file_size)) +
  geom_point(aes(group=file_type, color=file_type)) +
  xlab('n') +
  ylab('file size in bytes')
}

plot_random_unif(10000)
```

```
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
```



For this problem, we chose three file types: 1. png 2. bmp 3. jpeg

Immediately, we can see that png images are much much bigger than bmp and jpeg. This makes sense since png is a lossless file type, so it will generally be bigger than bmp or jpeg. Also, the file size does not seem to vary much between different values of n between 1 and 10,000.

Question 5: Diamonds

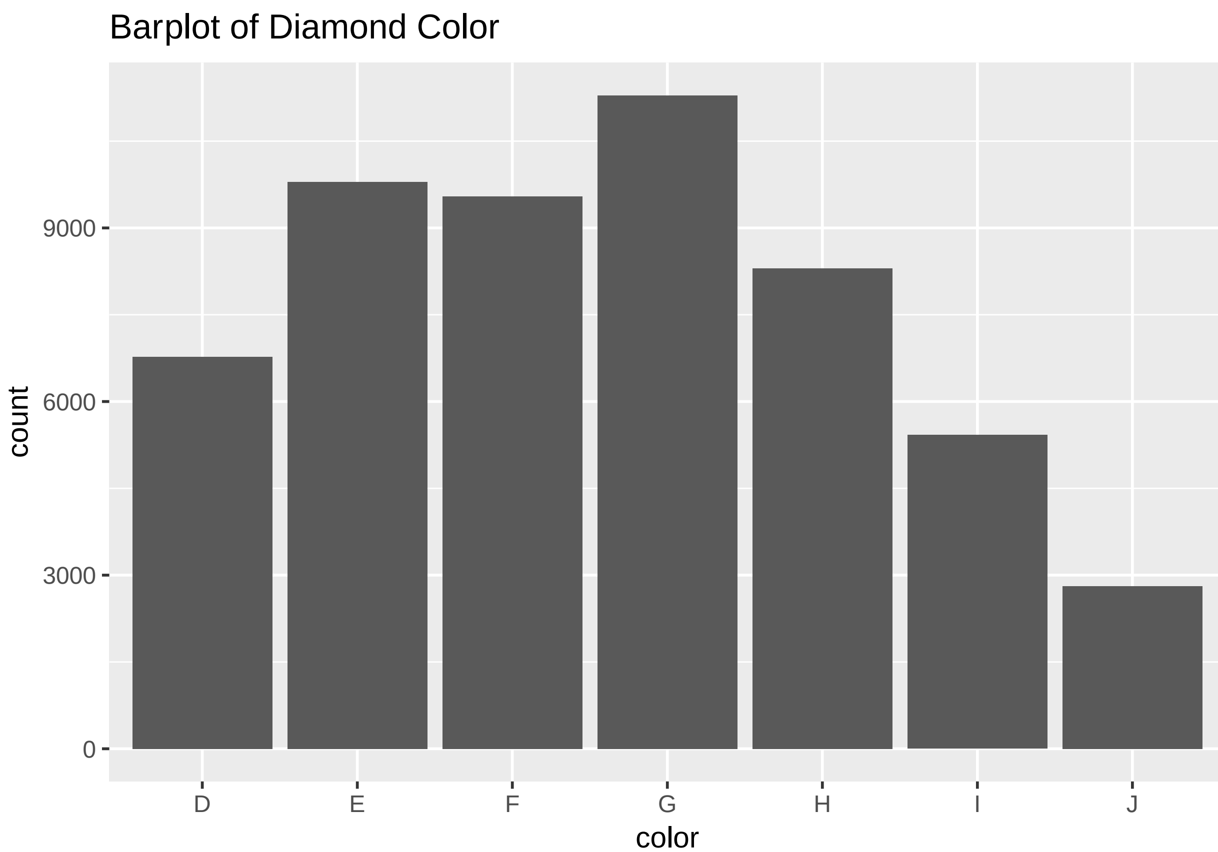
```
data(diamonds)
df = diamonds[, c('color', 'carat', 'price')]
print(df)
```

```
## # A tibble: 53,940 x 3
##    color carat price
##    <ord> <dbl> <int>
##  1     E  0.23   326
##  2     E  0.21   326
##  3     E  0.23   327
##  4     I  0.29   334
##  5     J  0.31   335
##  6     J  0.24   336
##  7     I  0.24   336
##  8     H  0.26   337
##  9     E  0.22   337
## 10     H  0.23   338
## # ... with 53,930 more rows
```

```r
print(summary(df))
```

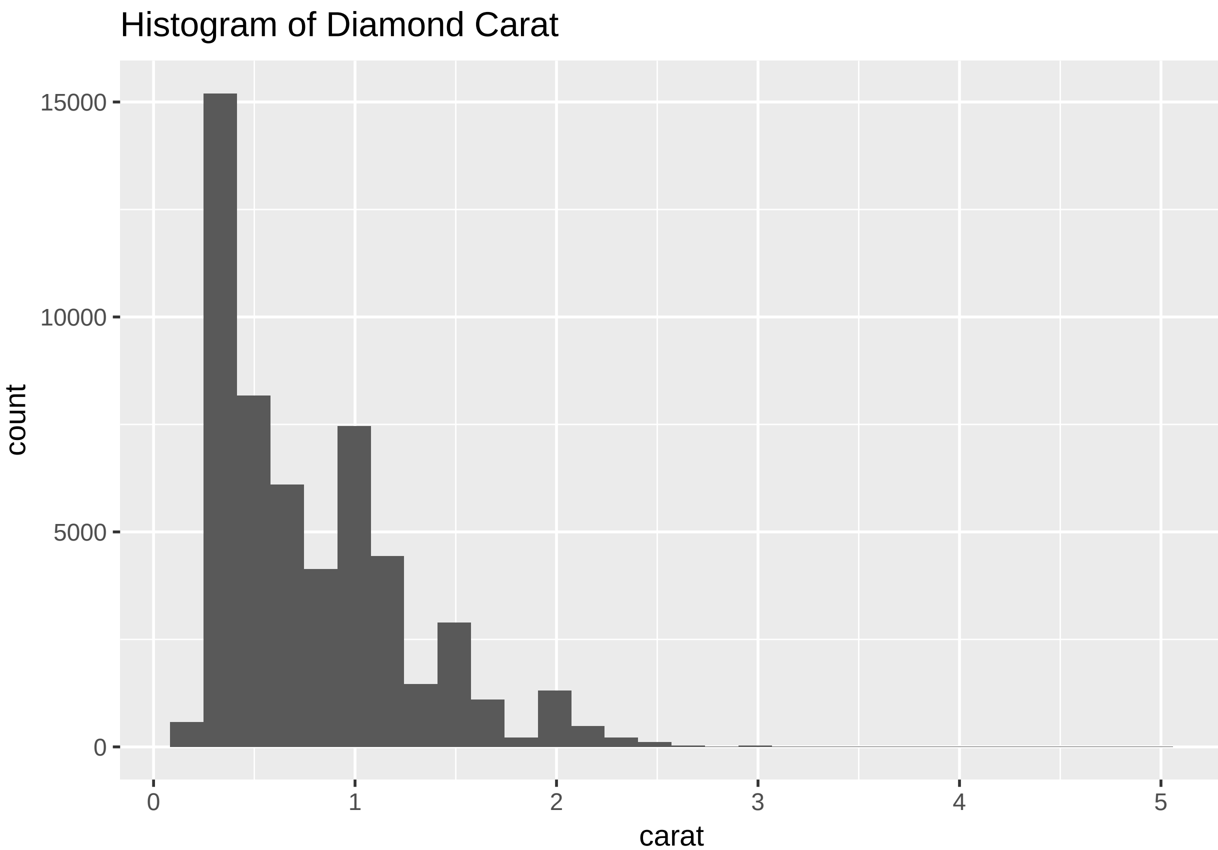
```
##  color        carat           price      
##  D: 6775   Min.   :0.2000   Min.   :  326  
##  E: 9797   1st Qu.:0.4000   1st Qu.:  950  
##  F: 9542   Median :0.7000   Median : 2401  
##  G:11292   Mean   :0.7979   Mean   : 3933  
##  H: 8304   3rd Qu.:1.0400   3rd Qu.: 5324  
##  I: 5422   Max.   :5.0100   Max.   :18823  
##  J: 2808
```

```r
ggplot(df, aes(x=color)) +
  geom_bar() +
  ggtitle('Barplot of Diamond Color')
```



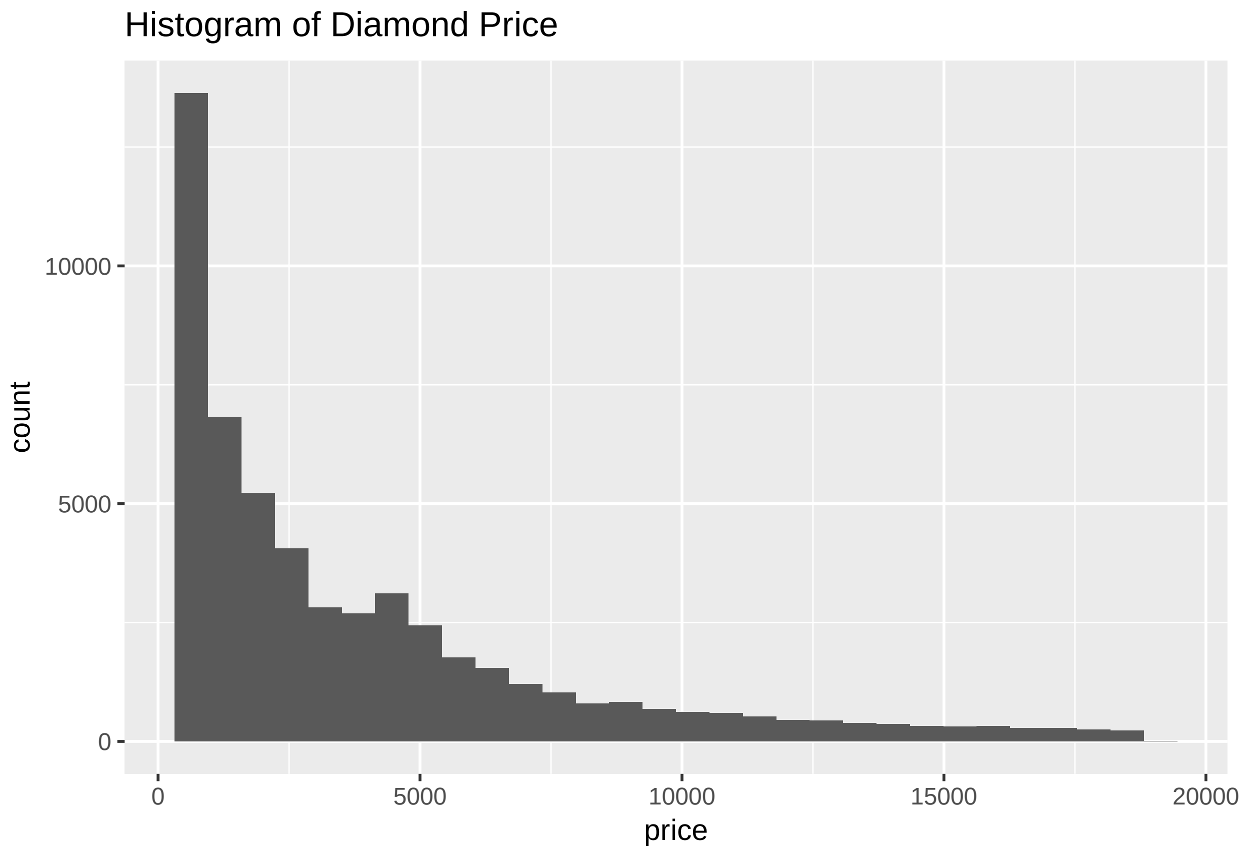

```r
ggplot(df, aes(x=carat)) +
  geom_histogram() +
  ggtitle('Histogram of Diamond Carat')
```

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




```r
ggplot(df, aes(x=price)) +
  geom_histogram() +
  ggtitle('Histogram of Diamond Price')
```

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



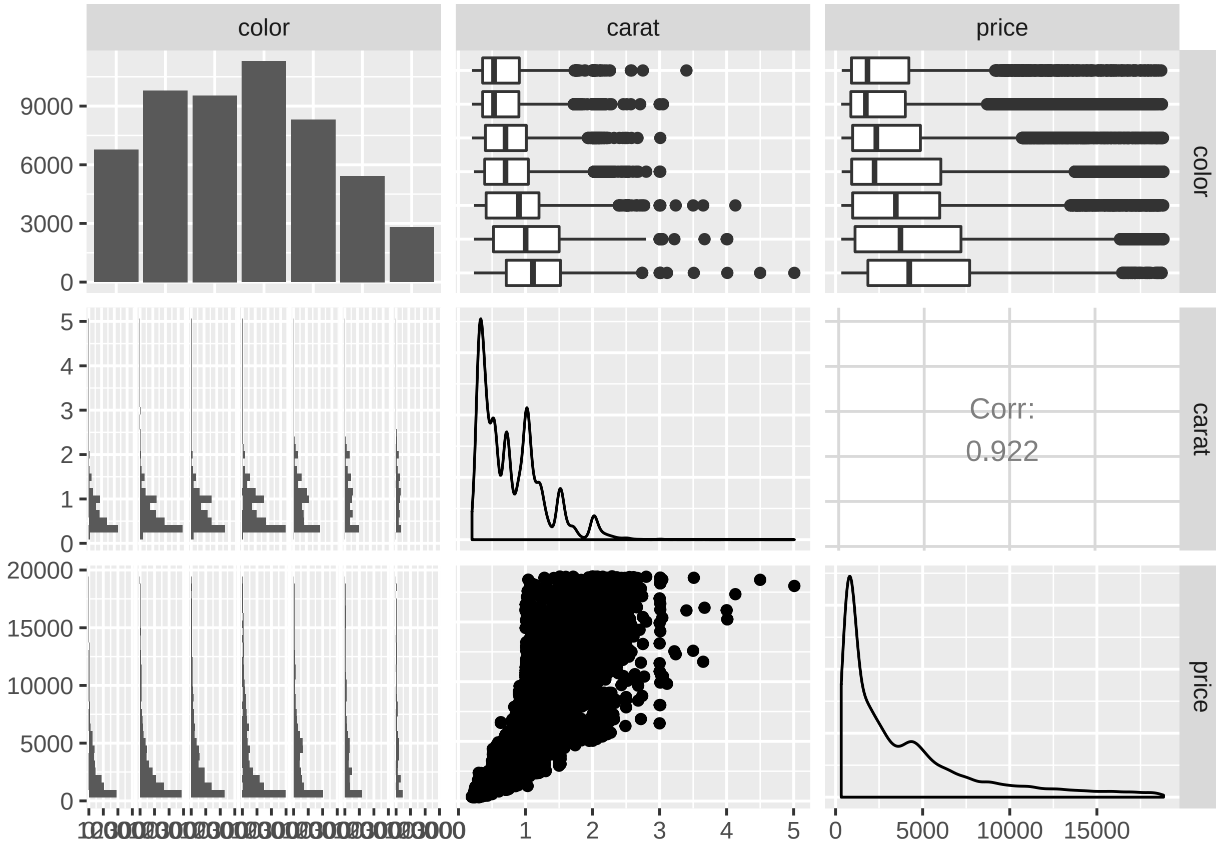
Looking first at the barplot of diamond color, we know from looking at `?diamond` that the colors are arranged such that J is the worst, and D is the best. Unsurprisingly, we see relatively low number of "J" colors. There is a bit of a right skew in the colors.

For both carats and price, there is a strong right skew to the data. With carats, the distribution is not as smooth, which makes sense because there are probably relatively "common" values for carats. However, if you look at price, notice that the distribution is much smoother and right skewed. This also makes sense because we know some diamonds can be extremely pricey, but the majority are not as expensive.

Now, we go ahead and plot the pairs

```
ggpairs(df, aes())
```

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



Notice that from the box plots of both `carat` and `price` with color, the distribution of `carat` and `price` increase with better quality of `color`. One can notice this by looking at how the IQR shifts to the right for each successive "increase" in color.

Another interesting point is the strong correlation between `carat` and `price`. Again this should be expected since `carat` is one of the primary determinants of `price` but you can seee the correlation is 0.922, and the scatter plot shows a strong positive relationship.