Comparing Two Variables.

Today we will continue to discuss Exploratory Data Analysis (EDA).

1. Two categorical variables.
2. One categorical variable and one numeric variable.
3. Two numeric variables.

```
library(tidyverse)
```

Two categorical variables.

```
diamonds %>% ggplot(aes(x = cut, y = color)) + geom_count()
```
```r
diamonds %>%
  count(color, cut) %>%
  ggplot(mapping = aes(x = cut, y = color)) +
  geom_tile(mapping = aes(fill = n))
```
```r
diamonds %>% count(color, cut)

## # A tibble: 35 x 3
## # Groups: color [7]
##     color cut    n
##   <ord> <ord> <int>
## 1    D    Fair  163
## 2    D    Good  662
## 3    D  Very Good 1513
## 4    D   Premium 1603
## 5    D   Ideal  2834
## 6    E    Fair  224
## 7    E    Good  933
## 8    E  Very Good 2400
## 9    E   Premium 2337
## 10   E   Ideal  3903
## # ... with 25 more rows
```

```r
diamonds %>% group_by(color, cut) %>% summarise(n=n())

## # A tibble: 35 x 3
## # Groups: color [7]
##     color cut    n
##   <ord> <ord> <int>
## 1    D    Fair  163
## 2    D    Good  662
## 3    D  Very Good 1513
## 4    D   Premium 1603
```
Contingency table.

```r
diamonds %>%
group_by(color, cut) %>%
summarise(n=n()) %>%
spread(cut, n)
```

Using the new `pivot_wider()` function, that replaces the `spread()`. You will need to update the `tidyr` package to version 1.0. The new function has a name that makes more sense and is more memorable.

```r
diamonds %>%
group_by(color, cut) %>%
pivot_wider(
  names_from = cut,
  values_from = n
)
```

Export the data to an Excel file and try making this Pivot Table.

```r
write.csv(diamonds, file="~/diamonds.csv")
```
One categorical variables and one numeric.

```r
ggplot(data = diamonds, mapping = aes(x = price)) + 
  geom_freqpoly(mapping = aes(colour = cut), binwidth = 500)
```

```r
ggplot(data = diamonds, mapping = aes(x = price, y = ..density..)) + 
  geom_freqpoly(mapping = aes(colour = cut), binwidth = 500)
```
```
ggplot(data = diamonds, mapping = aes(x = cut, y = price)) +
  geom_boxplot()
```
Two numeric variables.

```r
ggplot(data = diamonds) + geom_point(mapping = aes(x = carat, y = price))```
ggplot(data = diamonds) +
  geom_point(mapping = aes(x = carat, y = price), alpha = 1 / 100)
smaller <- diamonds %>%
  filter(carat < 3)

ggplot(data = smaller) +
  geom_bin2d(mapping = aes(x = carat, y = price))
library(hexbin)

ggplot(data = smaller) +
  geom_hex(mapping = aes(x = carat, y = price))