NBA 4920/6921 Lecture 2

Data Exploration and Visualization

Murat Unal

Johnson Graduate School of Management

09/02/2021

Agenda

- Quiz 1
- Review
- Quick Intro to R Markdown
- Exploratory Data Analysis (EDA)
- Variation
- Co-variation
- Visualization
- Start Linear Regression

Load/install the following packages

```
rm(list=ls())
options("scipen"=100,"digits"=8)

library(tidyverse)
library(ISLR)
library(cowplot)
library(ggcorrplot)
library(stargazer)
library(corrr)
data <- data.frame(ggplot2::mpg)

#to get more info about the dataset type:
#?ggplot2::mpg
```

Exploratory Data Analysis (EDA)

Before we start building models we need to understand the data.

EDA refers to the process of constructing a preliminary understanding of the data before running models.

EDA is an important part of any data analysis. Use EDA to:

1. Generate questions about your data
2. Search for answers by visualizing, transforming, and/or modeling your data
3. Use what you learn to refine your questions and/or generate new questions

Start with the structure of the data and some basic descriptives.

```
str(data)
```

```
'data.frame':   234 obs. of  11 variables:
 $ manufacturer: chr  "audi" "audi" "audi" "audi" ...
 $ model       : chr  "a4" "a4" "a4" "a4" ...
 $ displ       : num  1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ..
 $ year        : int  1999 1999 2008 2008 1999 1999 2008 19
 $ cyl         : int  4 4 4 4 6 6 6 4 4 4 ...
 $ trans       : chr  "auto(l5)" "manual(m5)" "manual(m6)"
 $ drv         : chr  "f" "f" "f" "f" ...
 $ cty         : int  18 21 20 21 16 18 18 18 16 20 ...
 $ hwy         : int  29 29 31 30 26 26 27 26 25 28 ...
 $ fl          : chr  "p" "p" "p" "p" ...
 $ class       : chr  "compact" "compact" "compact" "compac
```

```r
names(data)
```

```
 [1] "manufacturer" "model"        "displ"        "year"
 [6] "trans"        "drv"          "cty"          "hwy"
[11] "class"
```

```r
ncol(data)
```

```
[1] 11
```

```r
nrow(data)
```

```
[1] 234
```

```r
head(data, n=3)
```

manufacturer	model	displ	year	cyl	trans	drv	cty	hw
audi	a4	1.8	1999	4	auto(l5)	f	18	2
audi	a4	1.8	1999	4	manual(m5)	f	21	2
audi	a4	2.0	2008	4	manual(m6)	f	20	3

```
tail(data)
```

	manufacturer	model	displ	year	cyl	trans	drv	ct
229	volkswagen	passat	1.8	1999	4	auto(l5)	f	1
230	volkswagen	passat	2.0	2008	4	auto(s6)	f	1
231	volkswagen	passat	2.0	2008	4	manual(m6)	f	2
232	volkswagen	passat	2.8	1999	6	auto(l5)	f	1
233	volkswagen	passat	2.8	1999	6	manual(m5)	f	1
234	volkswagen	passat	3.6	2008	6	auto(s6)	f	1

```
summary(data)[,c(1:3)]
```

```
 manufacturer          model               displ       
 Length:234         Length:234         Min.   :1.6000  
 Class :character   Class :character   1st Qu.:2.4000  
 Mode  :character   Mode  :character   Median :3.3000  
                                       Mean   :3.4718  
                                       3rd Qu.:4.6000  
                                       Max.   :7.0000  
```

We can also use the `stargazer()` function to produce easy to read summary statistics tables.

```r
stargazer(data, summary = TRUE, type = "text")
```

```
==============================================================================
Statistic  N    Mean    St. Dev.  Min  Pctl(25) Pctl(75)  M
------------------------------------------------------------------------------
displ     234   3.472    1.292   1.600  2.400    4.600   7.
year      234 2,003.500  4.510   1,999  1,999    2,008   2,
cyl       234   5.889    1.612     4      4        8
cty       234  16.859    4.256     9      14       19     3
hwy       234  23.440    5.955    12      18       27     4
------------------------------------------------------------------------------
```

We want to have a clear idea about the missing values in the data.

```
colSums(is.na(data))
```

```
manufacturer         model         displ          year 
           0             0             0             0 
         drv           cty           hwy            fl 
           0             0             0             0 
```

We can also use `sapply()` for this

```
sapply(data, function(y) sum(is.na(y)))
```

```
manufacturer        model        displ         year
           0            0            0            0
         drv          cty          hwy           fl
           0            0            0            0
```

If there are missing observations you can remove them using the `na.omit()` function

The following questions will help us in understanding the data:

1. What type of variation occurs within my variables?
2. What type of covariation occurs between my variables?

Variation

Variation is the tendency of the values of a variable to change from measurement to measurement.

You can see variation easily in real life; if you measure any continuous variable twice—and precisely enough—you will get two different results.

Variation can be summarized in different ways, each providing you unique understanding of how the values are spread out.

```r
# Range
range(data$hwy, na.rm = TRUE)
```

```
[1] 12 44
```

```r
# Percentiles
# default quantile() percentiles are 0%, 25%, 50%,
# 75%, and 100%
quantile(data$hwy, na.rm = TRUE)
```

```
  0%  25%  50%  75% 100% 
  12   18   24   27   44 
```

```r
# we can customize quantile() for specific percentiles
quantile(data$hwy,
         probs = seq(from = 0, to = 1, by = .1),
         na.rm = TRUE)
```

```
  0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100% 
12.0 16.3 17.0 19.0 22.0 24.0 26.0 26.0 29.0 30.0 44.0 
```

Use `group_by()` to compute summary statistics by one or multiple categorical variables

```
data %>% group_by(class) %>% summarize(
                                 n = n(),
                                 mean_hwy = mean(hwy),
                                 mean_displ = mean(displ))
```

class	n	mean_hwy	mean_displ
2seater	5	24.800000	6.1600000
compact	47	28.297872	2.3255319
midsize	41	27.292683	2.9219512
minivan	11	22.363636	3.3909091
pickup	33	16.878788	4.4181818
subcompact	35	28.142857	2.6600000
suv	62	18.129032	4.4564516

```r
data %>% group_by(class,drv) %>% summarize(
                                   n = n(),
                                   mean_hwy = mean(hwy),
```

class	drv	n	mean_hwy	mean_displ
2seater	r	5	24.800000	6.1600000
compact	4	12	25.833333	2.4500000
compact	f	35	29.142857	2.2828571
midsize	4	3	24.000000	3.3666667
midsize	f	38	27.552632	2.8868421
minivan	f	11	22.363636	3.3909091
pickup	4	33	16.878788	4.4181818
subcompact	4	4	26.000000	2.3500000
subcompact	f	22	30.545455	2.0136364
subcompact	r	9	23.222222	4.3777778
suv	4	51	18.274510	4.2568627
suv	r	11	17.454545	5.3818182

Co-variation

Variation describes the behavior within a variable, co-variation describes the behavior between variables.

Co-variation is the tendency for the values of two or more variables to vary together in a related way.

We can summarize the linear dependence between two quantities using the **correlation coefficient**.

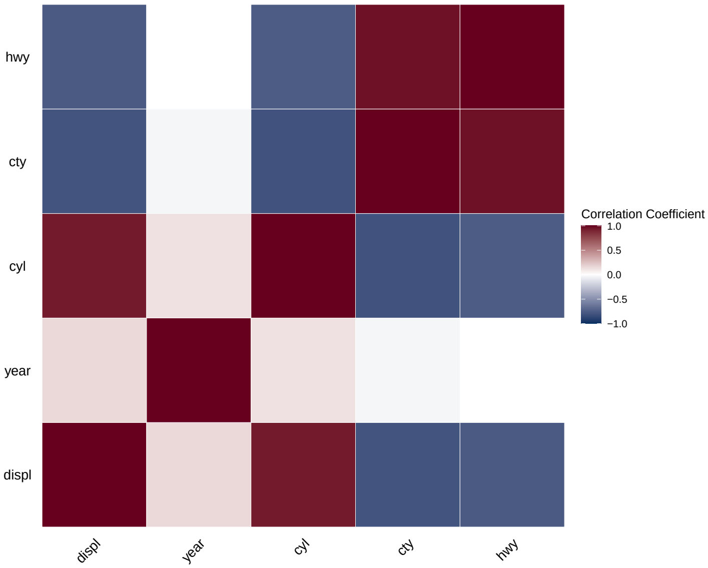
Let's select the numeric variables in the data and compute their correlations using the `cor()` function.

```
# Find the numeric columns
num_cols =  unlist(lapply(data, is.numeric))
# Create the correlation matrix
corr = cor(data[,num_cols])
corr
```

```
            displ          year         cyl          cty
displ  1.00000000  0.1478428165  0.93022710 -0.798523969 -0
year   0.14784282  1.0000000000  0.12224535 -0.037232291  0
cyl    0.93022710  0.1222453474  1.00000000 -0.805771408 -0
cty   -0.79852397 -0.0372322909 -0.80577141  1.000000000  0
hwy   -0.76602002  0.0021576431 -0.76191235  0.955915914  1
```

Let's also visualize the correlations using ggcorrplot().

```
ggcorrplot(corr,
     type = "full",lab = FALSE,
    legend.title = "Correlation Coefficient",
    colors = c("#053061", "white", "#67001f"),
    ggtheme = ggplot2::theme_void,
    outline.col = "white")
```



hwy
cty
cyl
year
displ
displ
year
cyl
cty
hwy
Correlation Coefficient
1.0
0.5
0.0
−0.5
−1.0

Let's create a data frame that has the absolute values of the correlations between `hwy` and other variables and sort them in descending order.

We'll use the `corrr()` package for this.

```r
# Convert correlation matrix to data frame
corr_df =   as_cordf(corr) %>%
# Focus on the hwy variable
  focus(hwy) %>%
# Get the absolute value of the correlation
# coefficient
  mutate(hwy = abs(hwy)) %>%
# Sort variables by absolute value of correlation
# coefficient
  arrange(desc(hwy)) %>%
# Clean up headers
  rename(`correlation with hwy` = term ) %>%
  rename(corr_coef = hwy)
corr_df
```

correlation with hwy	corr_coef
cty	0.95591591
displ	0.76602002
cyl	0.76191235
year	0.00215764

Exercise:

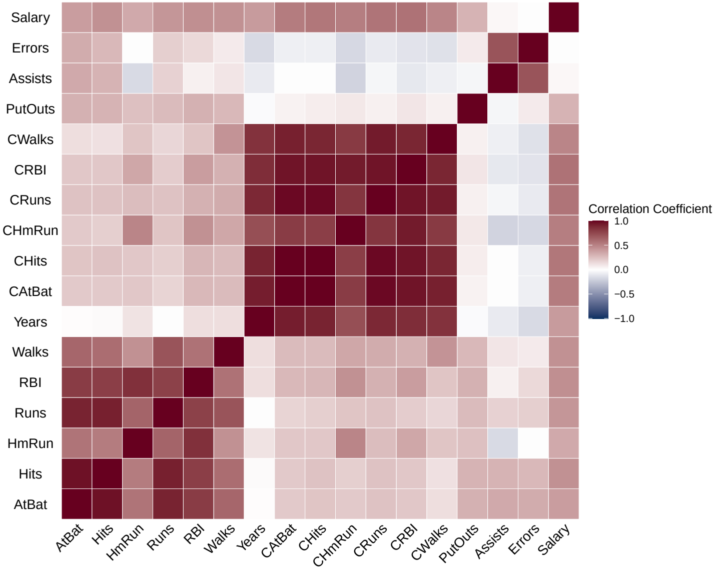
1. Read in the `Hitters` data from the `ISLR` package.
2. Remove observations with missing values.
3. Find the numeric variables.
4. Create the correlation matrix
5. Create the the correlation plot.
6. Display the first 3 variables that have the **lowest** absolute correlations with the `Salary`.

```r
Hitters <- ISLR::Hitters
Hitters <- na.omit(Hitters)
# Find the numeric columns
num_cols =  unlist(lapply(Hitters, is.numeric))
```

```r
# Create the correlation matrix
corr = cor(Hitters[,num_cols])

corr[1:4,1:4]
```

```
          AtBat      Hits     HmRun      Runs
AtBat 1.00000000 0.96396913 0.55510215 0.89982910
Hits  0.96396913 1.00000000 0.53062736 0.91063014
HmRun 0.55510215 0.53062736 1.00000000 0.63107588
Runs  0.89982910 0.91063014 0.63107588 1.00000000
```

```r
# Create the plot
ggcorrplot(corr,
  type = "full",
  lab = FALSE,
  legend.title = "Correlation Coefficient",
  colors = c("#053061", "white", "#67001f"),
  ggtheme = ggplot2::theme_void,
  outline.col = "white"
)
```



Salary
Errors
Assists
PutOuts
CWalks
CRBI
CRuns
CHmRun
CHits
CAtBat
Years
Walks
RBI
Runs
HmRun
Hits
AtBat
AtBat
Hits
HmRun
Runs
RBI
Walks
Years
CAtBat
CHits
CHmRun
CRuns
CRBI
CWalks
PutOuts
Assists
Errors
Salary
Correlation Coefficient
1.0
0.5
0.0
−0.5
−1.0

```
# Convert correlation matrix to data frame
corr_df =   as_cordf(corr) %>%
# Focus on the Salary variable
  focus(Salary) %>%
# Get the absolute value of the correlation
# coefficient
  mutate(Salary = abs(Salary)) %>%
# Sort variables by absolute value of correlation
# coefficient
  arrange(Salary) %>%
# Clean up headers
  rename(`correlation with Salary` = term ) %>%
  rename(corr_coef = Salary)
```

```r
head(corr_df,n=3)
```

correlation with Salary	corr_coef
Errors	0.00540070
Assists	0.02543614
PutOuts	0.30048036

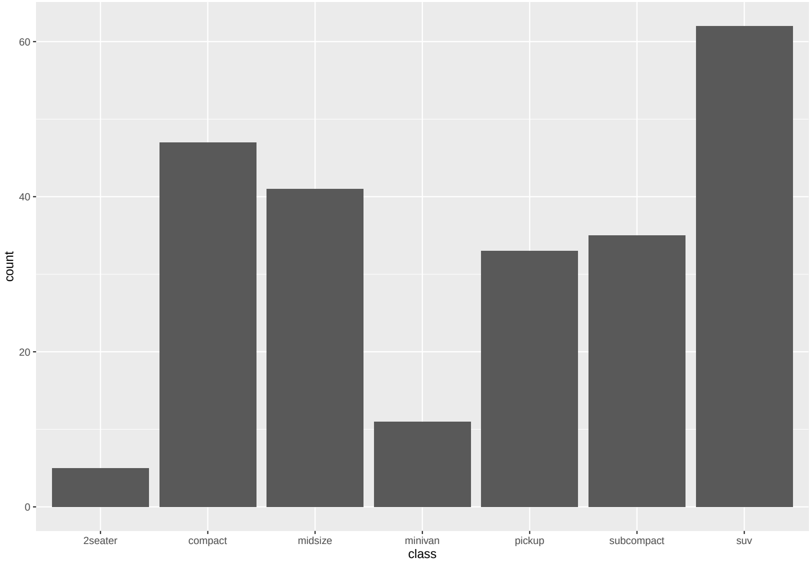
Visualization

Summary statistics and correlations are not enough for understanding the data.

The best way to understand a variable's pattern of variation is to visualize the distribution of the variable's values.

To examine the distribution of a categorical variable, use a bar chart.

```
ggplot(data = mpg) +
  geom_bar(mapping = aes(x = class))
```



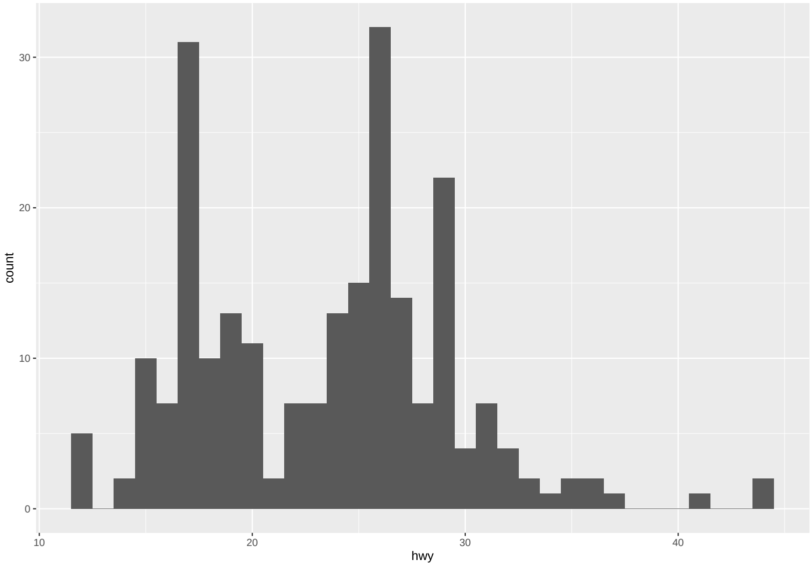
The height of the bars displays how many observations occurred with each x value. You can compute these values manually with `dplyr::count()`:

```
mpg %>% count(class)
```

class	n
2seater	5
compact	47
midsize	41
minivan	11
pickup	33
subcompact	35
suv	62

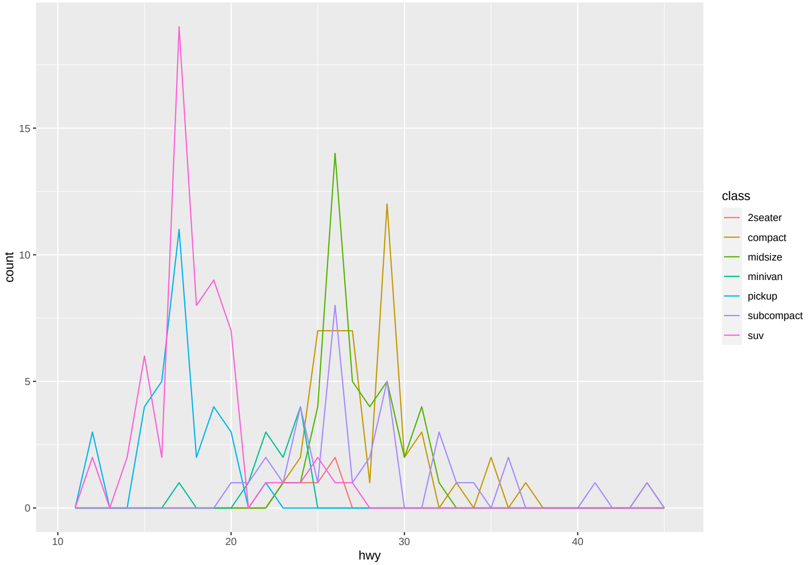
To examine the distribution of a continuous variable, use a hist:

```
ggplot(data = data) +
  geom_histogram(mapping = aes(x = hwy), binwidth = 1)
```



Overlaying multiple histograms in the same plot can be useful in discerning differences between categorical variables.

```
ggplot(data = data,
       mapping = aes(x = hwy, colour = class)) +
  geom_freqpoly(binwidth = 1)
```


count
15
10
5
0
10
20
30
40
hwy
class
2seater
compact
midsize
minivan
pickup
subcompact
suv

Frequencies

In both bar charts and histograms, tall bars show the common values of a variable, i.e. the values that appear frequently.

Look for anything unexpected:

- Which values are the most common? Why?
- Which values are rare? Why? Does that match your expectations?
- Can you see any unusual patterns? What might explain them?
- Are there any outliers?

Why look at data?

Good visualization methods offer extremely valuable tools that we can use to better understand the relationship between two variables.

```
str(anscombe)
```

```
'data.frame':	11 obs. of  8 variables:
 $ x1: num  10 8 13 9 11 14 6 4 12 7 ...
 $ x2: num  10 8 13 9 11 14 6 4 12 7 ...
 $ x3: num  10 8 13 9 11 14 6 4 12 7 ...
 $ x4: num  8 8 8 8 8 8 8 19 8 8 ...
 $ y1: num  8.04 6.95 7.58 8.81 8.33 ...
 $ y2: num  9.14 8.14 8.74 8.77 9.26 8.1 6.13 3.1 9.13 7.26
 $ y3: num  7.46 6.77 12.74 7.11 7.81 ...
 $ y4: num  6.58 5.76 7.71 8.84 8.47 7.04 5.25 12.5 5.56 7.
```

```r
colMeans(anscombe)[1:4]
```

```
x1 x2 x3 x4
 9  9  9  9
```

```r
colMeans(anscombe)[5:8]
```

```
      y1       y2       y3       y4
7.5009091 7.5009091 7.5000000 7.5009091
```

```r
#Correlation between pairs of x and y
cor(anscombe)[5:8,1:4]
```

```
            x1          x2          x3          x4
y1  0.81642052  0.81642052  0.81642052 -0.52909274
y2  0.81623651  0.81623651  0.81623651 -0.71843653
y3  0.81628674  0.81628674  0.81628674 -0.34466100
y4 -0.31404671 -0.31404671 -0.31404671  0.81652144
```

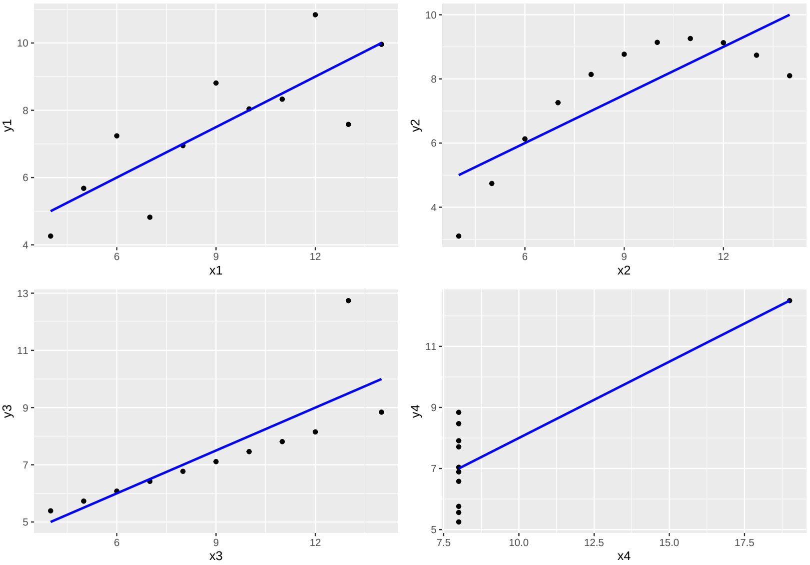
Exercise

Now let's create scatter plots for this data and fit a regression line for each pair

```r
p1 <- ggplot(anscombe, aes(x1,y1,)) +
  geom_point()+
  geom_smooth(method='lm', formula= y~x,se=FALSE,
                                colour = "blue")

p2 <- ggplot(anscombe, aes(x2,y2,)) +
  geom_point()+
  geom_smooth(method='lm', formula= y~x,se=FALSE,
                                colour = "blue")
```

```r
p3 <- ggplot(anscombe, aes(x3,y3,)) +
  geom_point()+
  geom_smooth(method='lm', formula= y~x,se=FALSE,
                           colour = "blue")

p4 <- ggplot(anscombe, aes(x4,y4,)) +
  geom_point()+
  geom_smooth(method='lm', formula= y~x,se=FALSE,
                           colour = "blue")

plot_grid(p1,p2,p3,p4)
```

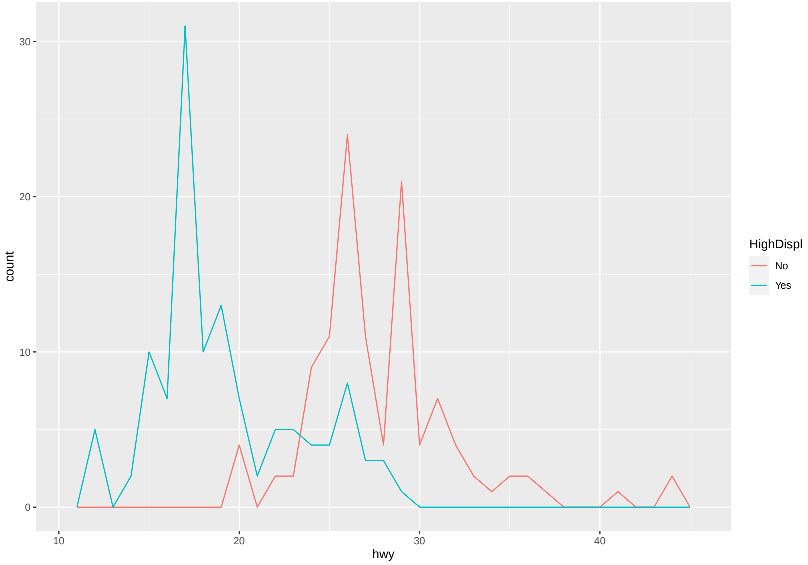
What is your interpretation of the relationship between each pair?



Exercise

Create a graph that shows the differences between the hwy distributions of two groups of cars: those that have `displ` below and greater or equal the median `displ`.

Solution:

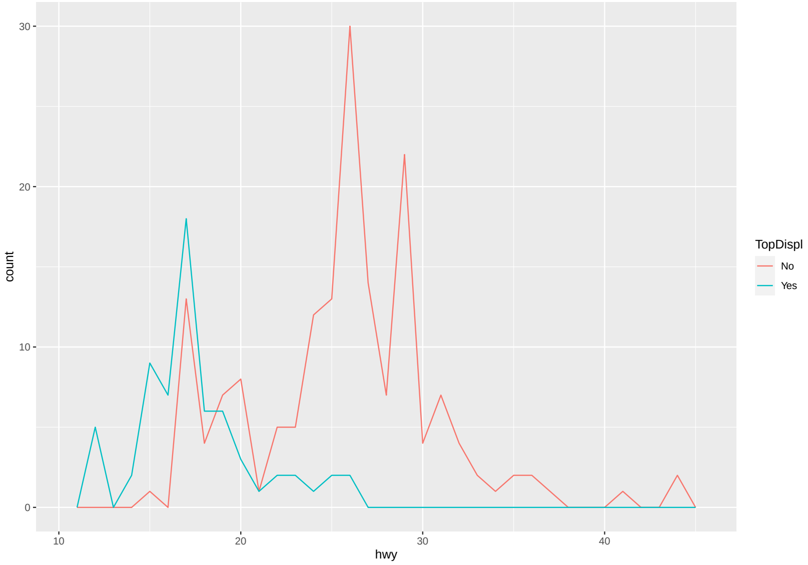


```r
data$HighDispl <- factor(
                  ifelse(data$displ>=median(data$displ),
                  "Yes","No"))
```

```r
ggplot(data = data,
       mapping = aes(x = hwy,colour = HighDispl)) +
  geom_freqpoly(binwidth = 1)
```

Repeat the same exercise for the cars in the top quartile and the rest.

Solution:



```
data$TopDispl <- factor(
                 ifelse(data$displ>=quantile(data$displ)[4]
                    "Yes","No"))
```

```
ggplot(data = data,
       mapping = aes(x = hwy, colour = TopDispl)) +
  geom_freqpoly(binwidth = 1)
```

What is wrong with the last figure?

The two groups differ in the number of Hitters.

```
summary(data$TopDispl)
```

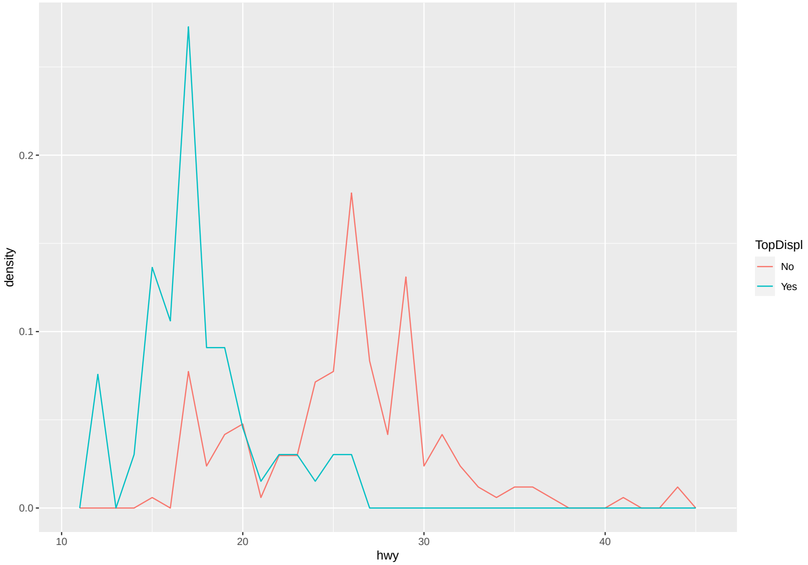
```
 No Yes
168  66
```

If one of the groups is much smaller than the others, the shapes can be misleading and it's hard to see the differences.

To make the comparison easier we need to swap what is displayed on the y-axis.

Instead of displaying `count`, we'll display `density`, which is the count standardized so that the area under each frequency polygon is one.

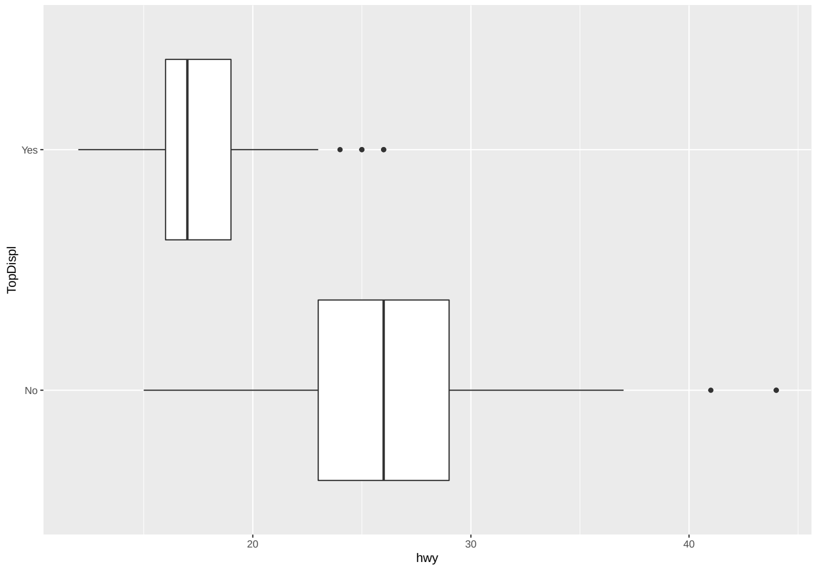
```r
ggplot(data = data,
       mapping = aes(x = hwy, y = ..density..)) +
  geom_freqpoly(mapping = aes(colour = TopDispl),
          binwidth = 1)
```



density
0.0
0.1
0.2
10
20
30
40
hwy
TopDispl
No
Yes

Let's take a look at the distribution of `hwy` by `displ` status using `geom_boxplot()`:

```
ggplot(data = data,
       mapping = aes(x = hwy, y = TopDispl)) +
  geom_boxplot()
```



Yes
No
TopDispl
20
30
40
hwy