Week 3
Exploratory Data Analysis

1. Descriptive stats
2. GroupBy
3. ANOVA
4. Correlation

---

1. Desc. statistics describe the basic features of the data (summary)

- df.describe() sum's stats (~~inc.~~ NaN's)
  - shows count, mean, stdev, min, 25%, 50%, 75%., and max of data
- For categorical vars, df[var].value_counts() value counts can describe categorical data *
- To create a box plot, use sns.boxplot(x = var_1, y = var_2, data = df)
- To create a scatterplot, first def the vars and use the matplotlib function plt.scatter(x, y)
  - plt.title("Title"); plt.xlabel("var_1"); plt.ylabel("var_2")

2. GroupBy

With a scatterplot, you can find correlations b/c both vars are #s. What if you want to see how a categorical var affects a numerical var?

1. Assign a var to encompass any indep + dep vars:
   var_0 = df['var_I', 'var_D']
2. Group categorical indep vars with dep var:
   var_1 = var_0.groupby(['var_{I1}', 'var_{I2}'] as_index = False).mean()
3. You can make a pivot table, which is easier to visualize:
   var_2 = var_1.pivot(index = 'var_{I1}', columns = 'var_{I2}')

* To convert to a dataframe:
df[var].valuecounts().to_frame()

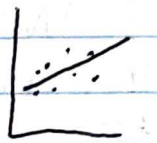
* To just find correlation, use:

df[['var$_1$', 'var$_2$']].corr()

3+4. Correlation

import matplotlib.pyplot as plt
import seaborn as sns

↘ To plot a regression line with a scatterplot, use:



```
sns.regplot(x = var1, y = var2, data = df)
Plt.ylim(0,)
```

Stat Methods

- To take the p-value and 'r' (certainty), use:

```
Pearson-coef, p-value = stats.pearsonr[['var1'], df['var2']]
```

- ANOVA
  - Correlation bet diff groups of a categorical var
  - F-test: $\frac{\text{var of means}}{\text{var of group}}$
  - 3 lines of code:

1. df_anova = df[['make', 'price']] ← initialize cat's
2. grouped_anova = df_anova.groupby(['make']) ← group by dep. var.
3. result = stats.f_oneway(grouped_anova.get_group('honda')['price'],
   grouped_anova.get_group('subaru')['price'])

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5 path = ____
6 df = pd.read_csv(path)
7 ...
:
```