DATA VISUALISATION WITH PYTHON

DR. ALVIN ANG

CONTENTS

INTRODUCTION

- There are multiple .CSVs / Data frames that we will visualize:
    a. A Simple Random Data Frame
    b. Boston Housing Data.csv
    c. Mtcars.csv
    d. College.csv
    e. Titanic.csv
    f. Iris.csv
    g. Fmri.csv

STEP 1: IMPORT ALL LIBRARIES

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

plt.style.use('dark_background')

print(plt.style.available)

#%matplotlib inline --> do not need to use this anymore (for older panda versions)
```

- Output:
    - 'Solarize_Light2',
    - '_classic_test_patch',
    - 'bmh',
    - 'classic',
    - 'dark_background',
    - 'fast',
    - 'fivethirtyeight',

- 'ggplot',
- 'grayscale',
- 'seaborn',
- 'seaborn-bright',
- 'seaborn-colorblind',
- 'seaborn-dark',
- 'seaborn-dark-palette',
- 'seaborn-darkgrid',
- 'seaborn-deep',
- 'seaborn-muted',
- 'seaborn-notebook',
- 'seaborn-paper',
- 'seaborn-pastel',
- 'seaborn-poster',
- 'seaborn-talk',
- 'seaborn-ticks',
- 'seaborn-white',
- 'seaborn-whitegrid',
- 'tableau-colorblind10']

- Helpful Visualization Links:

    a. https://python-tricks.com/styling-graphs-in-matplotlib/

    b. https://www.tutorialspoint.com/how-to-change-the-text-color-of-font-in-the-legend-using-matplotlib

    c. https://tonysyu.github.io/raw_content/matplotlib-style-gallery/gallery.html

    d. https://matplotlib.org/2.0.2/api/colors_api.html

A SIMPLE RANDOM DATA FRAME

STEP 1: CREATE THE RANDOM DATA FRAME

```
df = pd.DataFrame(np.random.randn(10, 4),
                  columns=['col1', 'col2', 'col3', 'col4'],
                  index=['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j'])
```

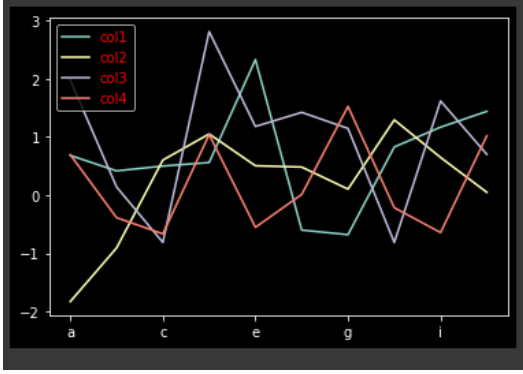
```
df
```

	col1	col2	col3	col4
a	0.682929	-1.831765	1.979614	0.694650
b	0.416908	-0.909262	0.140248	-0.388065
c	0.500005	0.602898	-0.814113	-0.666459
d	0.560322	1.051715	2.813758	1.041801
e	2.331652	0.505058	1.180666	-0.553576
f	-0.601580	0.482549	1.423800	0.011406
g	-0.680515	0.102793	1.146497	1.525807
h	0.828052	1.294050	-0.814635	-0.216515
i	1.168756	0.647984	1.619707	-0.644145
j	1.441174	0.047578	0.700822	1.020817

STEP 2: LINE PLOT

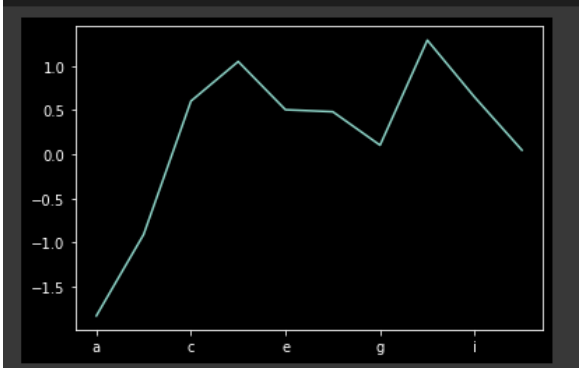
ALL COLUMNS

```
df.plot()

leg = plt.legend(loc='upper left')
for text in leg.get_texts():
    text.set_color("red")
```



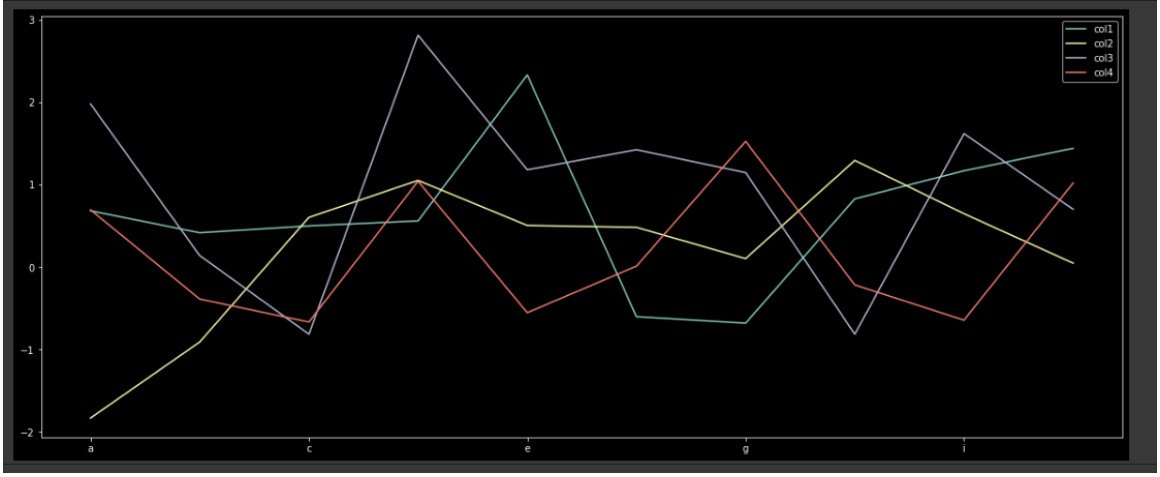
PARTICULAR COLUMN

```
df['col2'].plot();
```



CHANGING THE SIZE

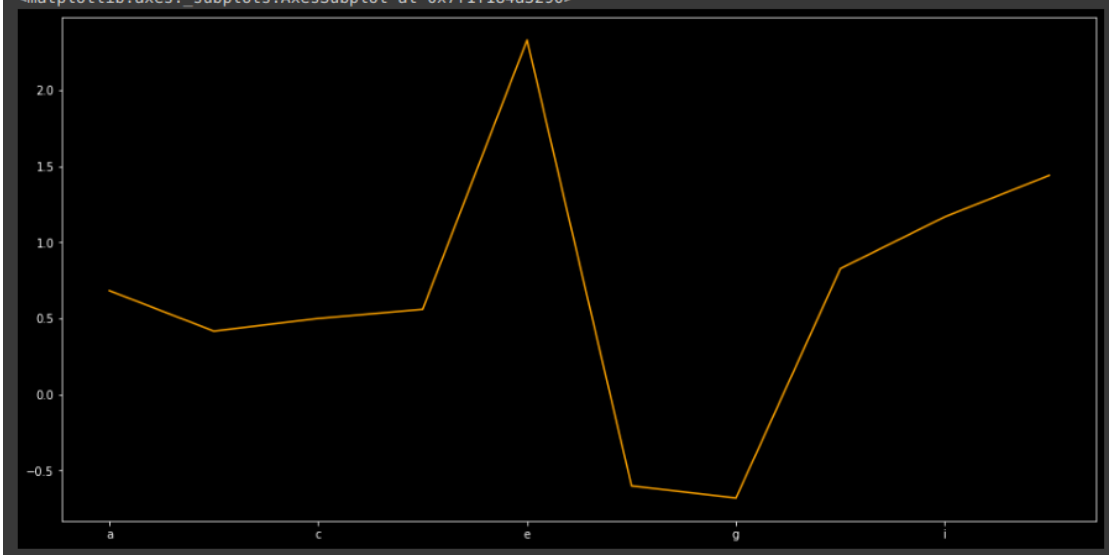
```
df.plot(figsize=(20,8));

# width = 20, height = 8
```



CHANGING THE COLOR

```
df['col1'].plot(color='orange', figsize=(16,8))
```

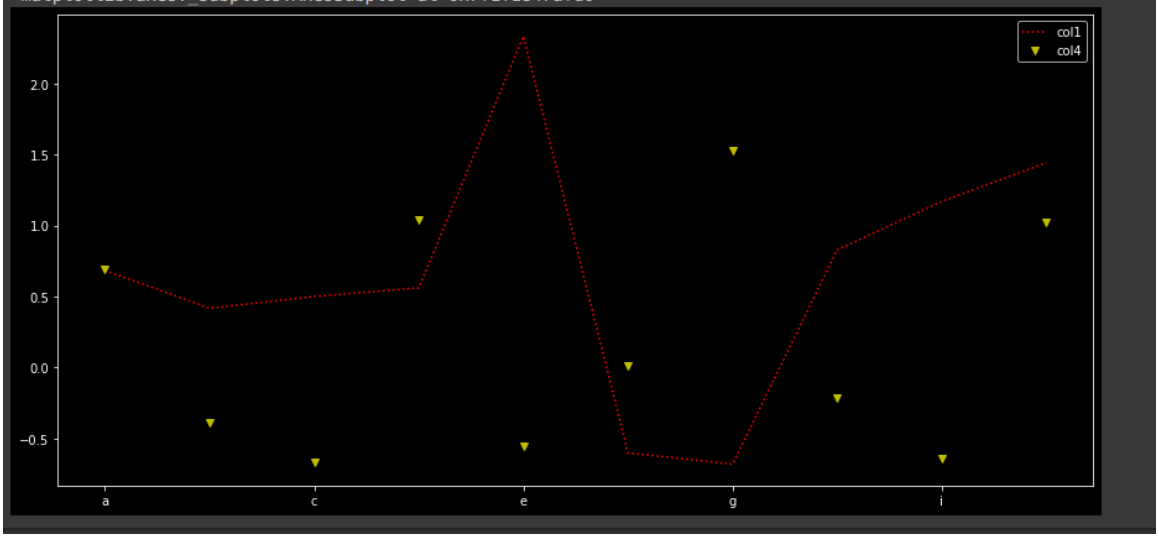
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1f184a3290>
```



CHANGING THE STYLE OF EACH LINE

```
df[['col1', 'col4']].plot(figsize=(15,7), style={'col1': ':r', 'col4': 'vy'})
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1f1847afd0>
```

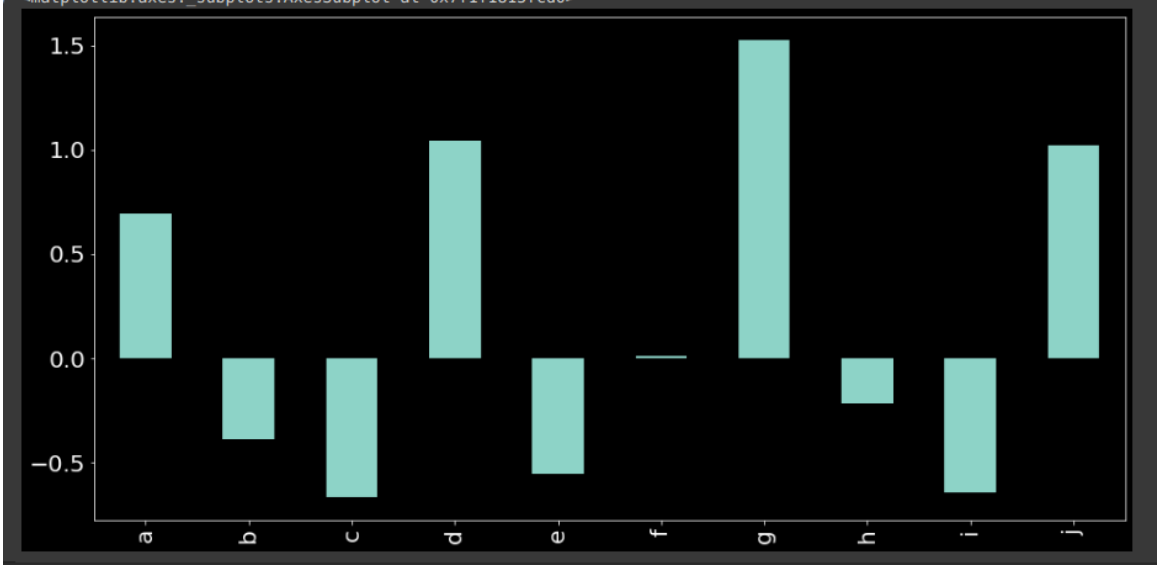


STEP 3: BAR PLOT

VERTICAL BAR PLOT

```
df['col4'].plot(kind='bar', figsize=(16,8), fontsize=20)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1f1815fcd0>
```

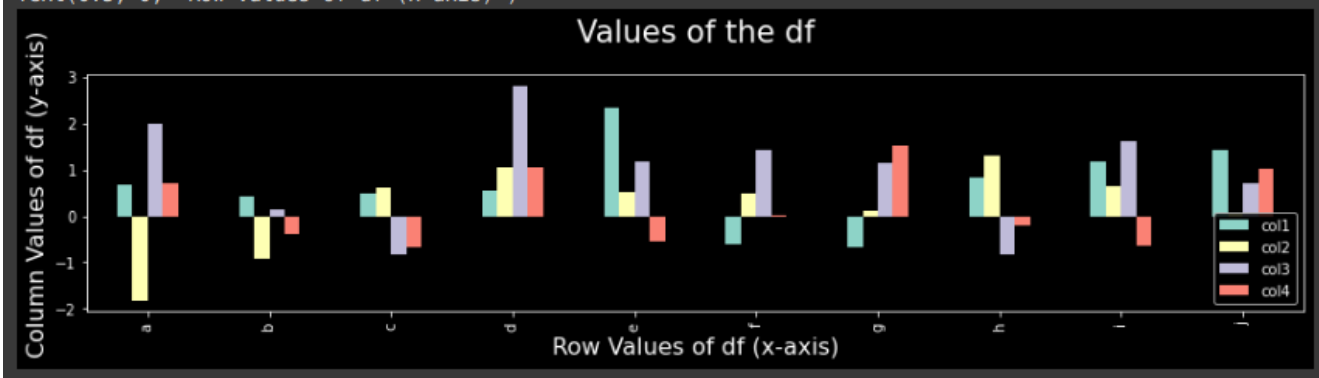


ADJUSTING THE BAR PLOT

- Assigning the df.plot to "adj"
- Setting the title
- Moving the legend
- Labeling the y axis
- Labeling the x axis

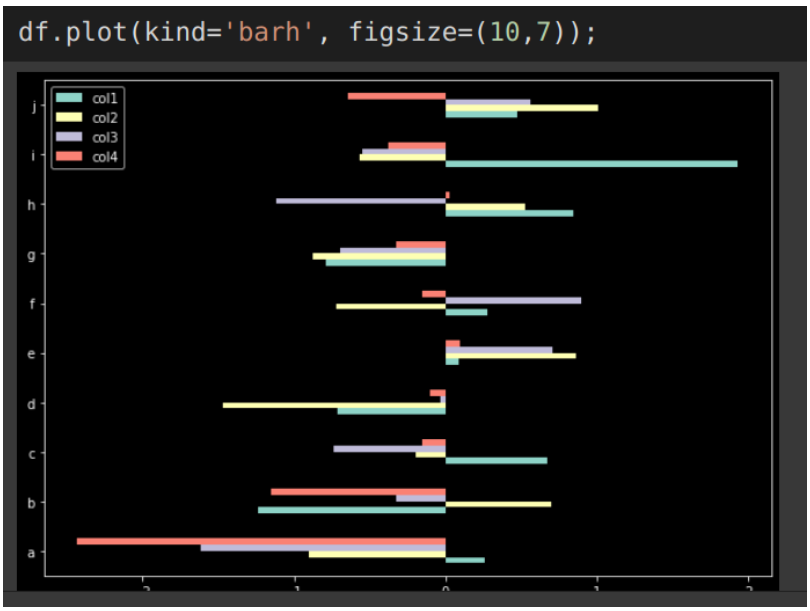
```
adj = df.plot(kind='bar', figsize=(15,3))

adj.set_title('Values of the df', fontsize=21, y=1.1)

adj.legend(loc=4) #1,2,3,4

adj.set_ylabel('Column Values of df (y-axis)', fontsize=16)

adj.set_xlabel('Row Values of df (x-axis)', fontsize=16)
```

```
Text(0.5, 0, 'Row Values of df (x-axis)')
```



HORIZONTAL BAR PLOT

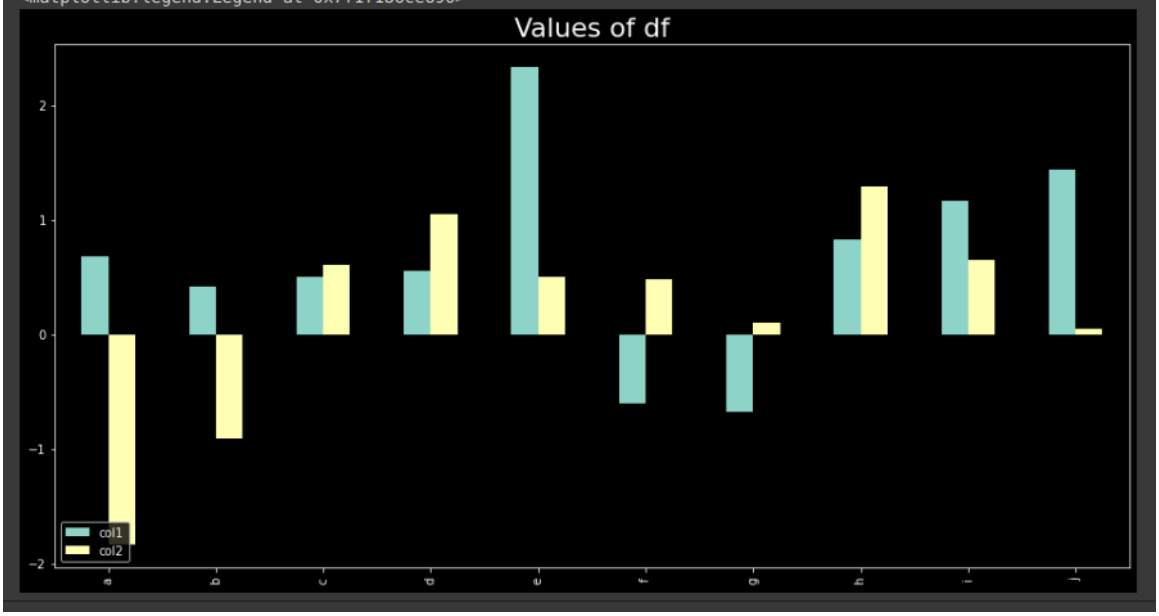
```
df.plot(kind='barh', figsize=(10,7));
```



ADJUSTING THE LEGEND

```
move_legend_1 = df[['col1','col2']].plot(kind='bar', figsize=(16,8))
move_legend_1.set_title('Values of df', fontsize=22)
move_legend_1.legend(loc=3)
```

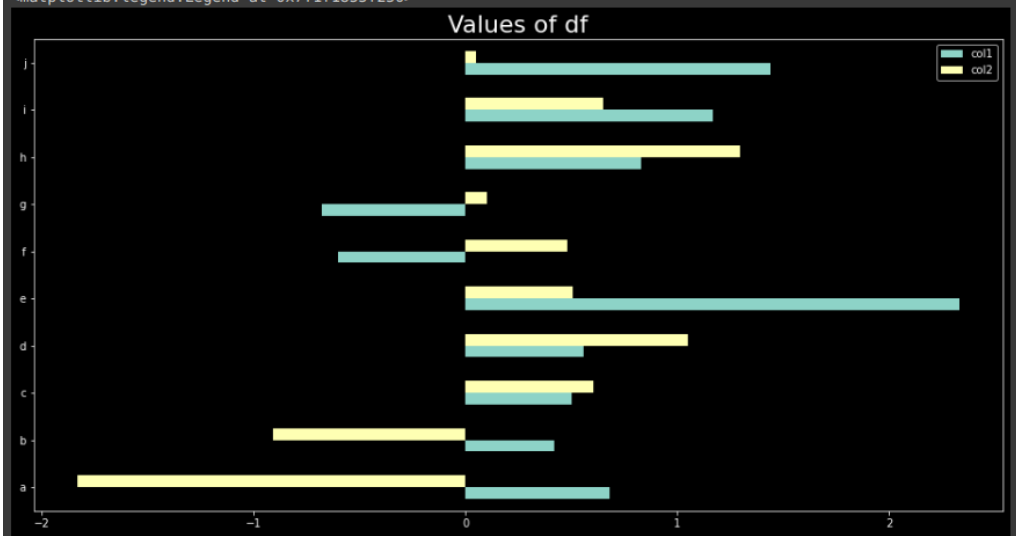
```
<matplotlib.legend.Legend at 0x7f1f186ce690>
```



MOVING LEGEND TO UPPPER RIGHT

```
move_legend_2 = df[['col1','col2']].plot(kind='barh', figsize=(16,8))
move_legend_2.set_title('Values of df', fontsize=22)
move_legend_2.legend(loc=1)
```

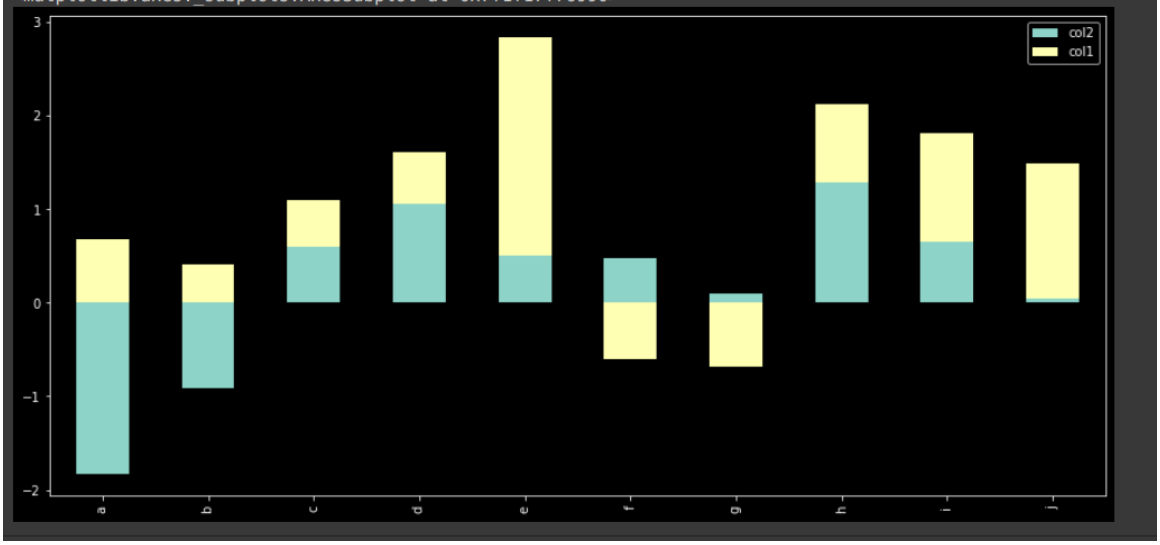
```
<matplotlib.legend.Legend at 0x7f1f1855f250>
```



VERTICAL STACKED BARS

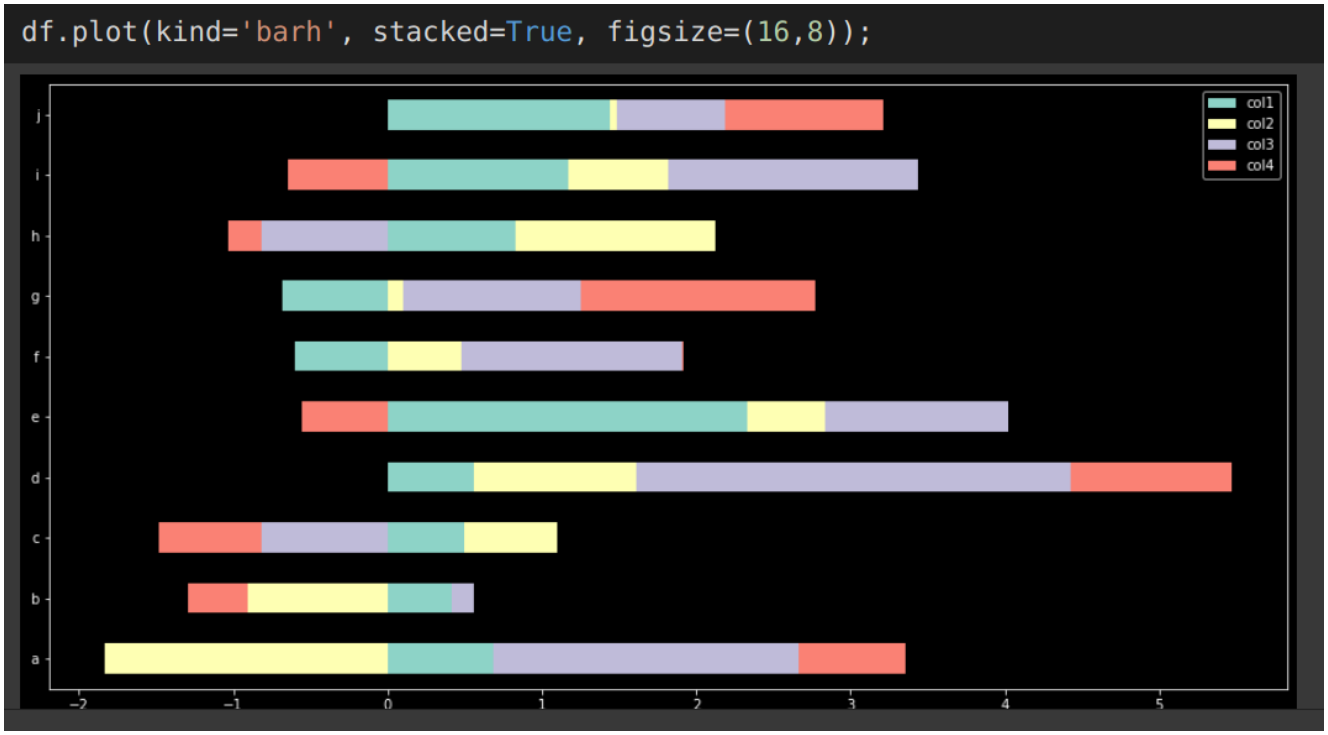
```
df[['col2', 'col1']].plot(kind='bar', stacked=True, figsize=(15,7))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1f17f76990>
```



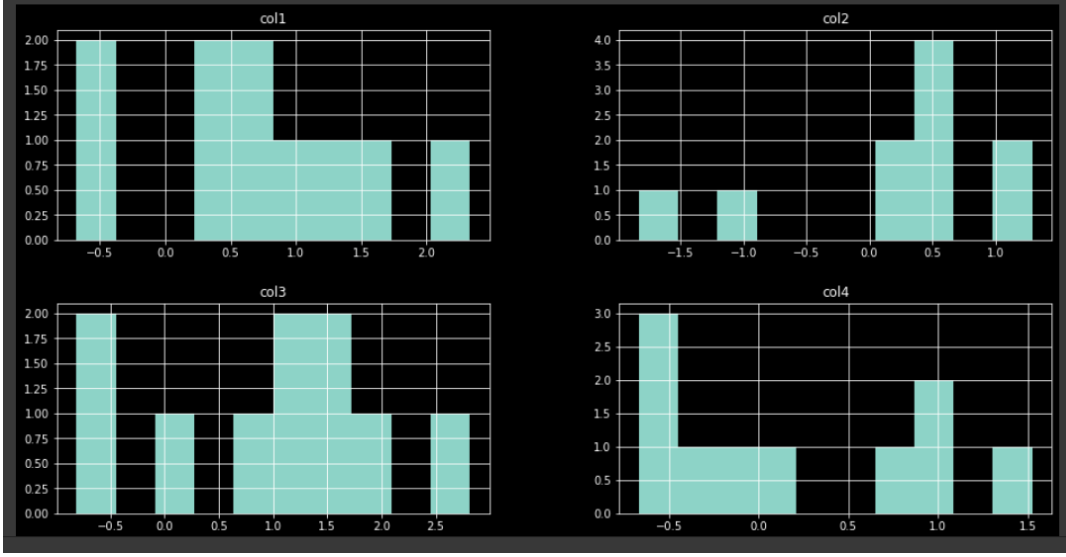
HORIZONTAL STACKED BARS

```
df.plot(kind='barh', stacked=True, figsize=(16,8));
```



STEP 4: HISTOGRAM

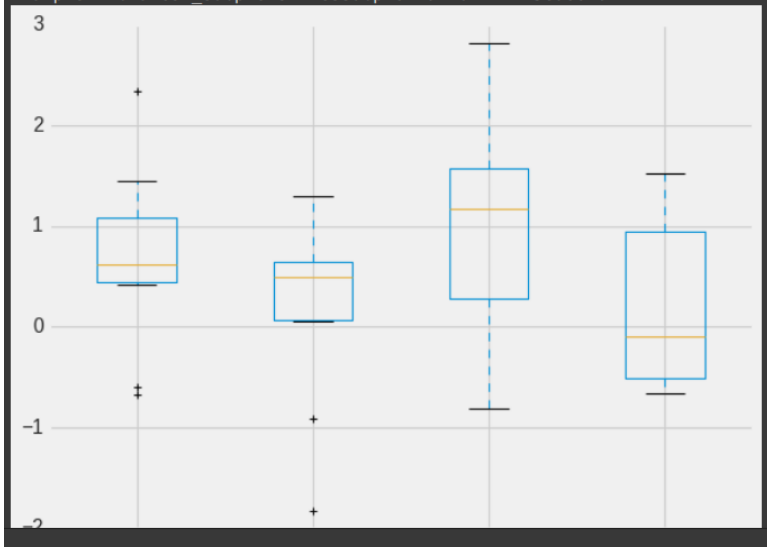
```
df.hist(figsize=(16,8));
```



STEP 5: BOX PLOT

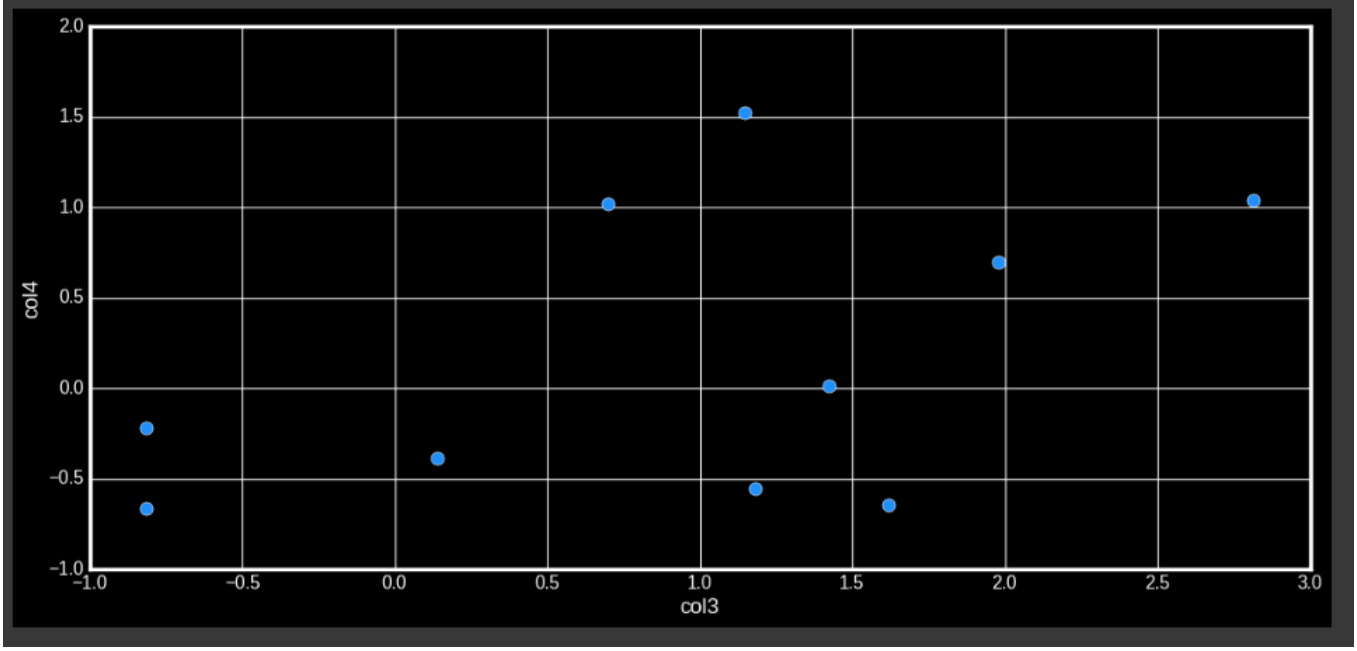
```
plt.style.use('fivethirtyeight')

df.boxplot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1f15ed6610>
```



STEP 6: SCATTER PLOT

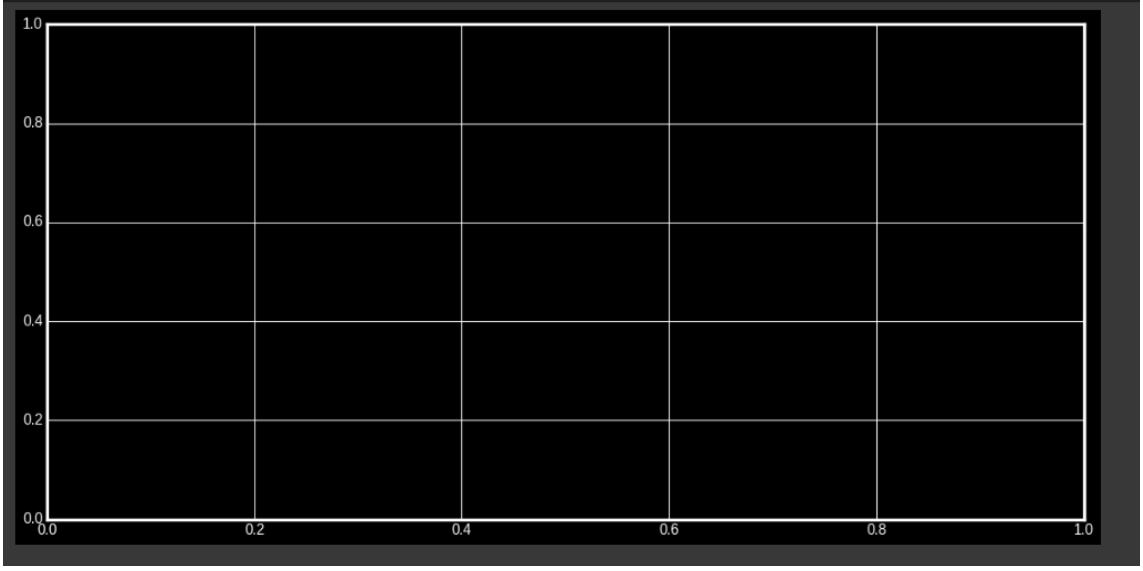
```
plt.style.use('dark_background')

df.plot(x='col3', y='col4', kind='scatter', color='dodgerblue',\
        figsize=(15,7), s=100);
```



STEP 7: SUB PLOT

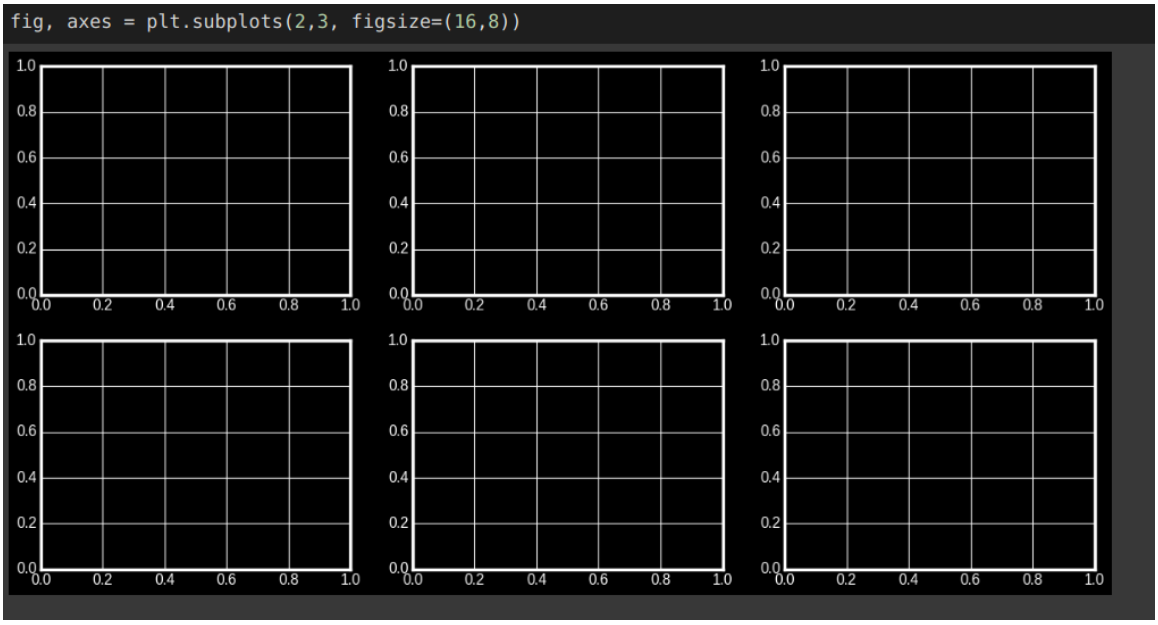
SINGLE SUB PLOT

```
fig = plt.subplots(1,1, figsize=(16,8))
```



MANY SUB PLOTS

```
fig, axes = plt.subplots(2,3, figsize=(16,8))
```

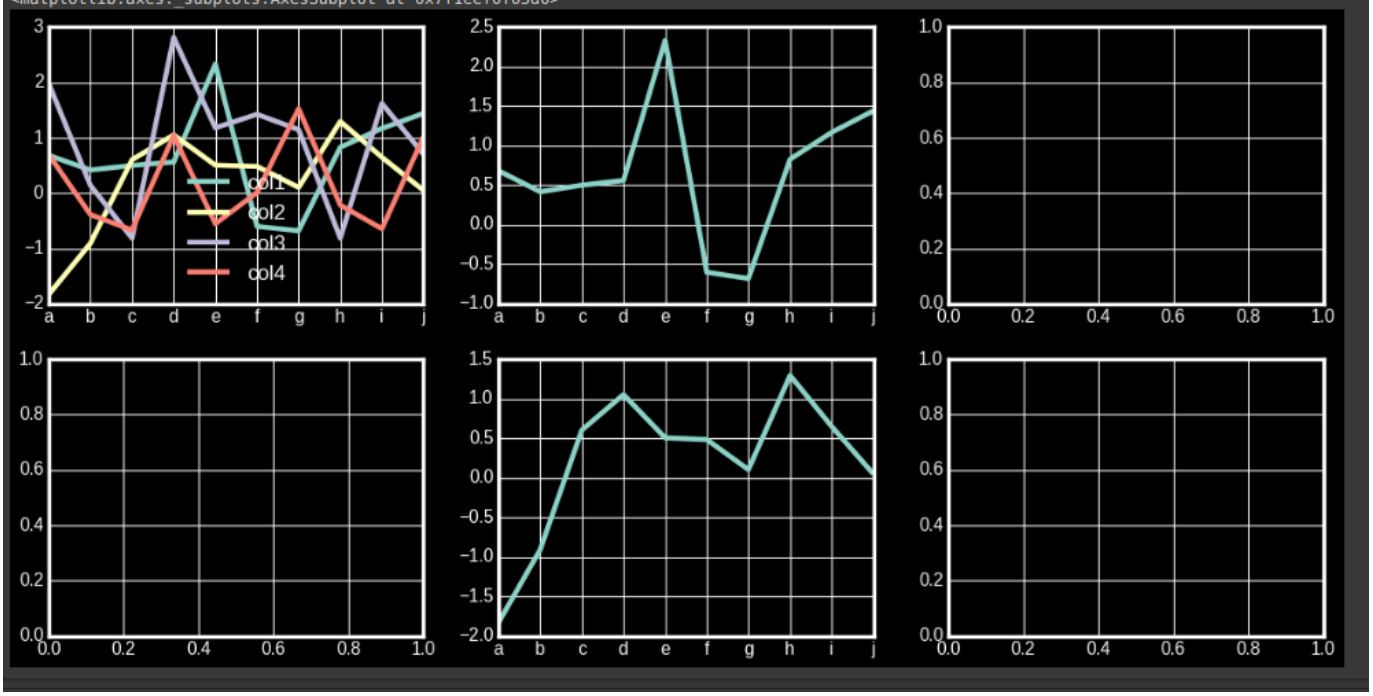


PASSING DATA FRAME TO THE SUB PLOT

- Pandas allows us to plot to a specified axes if we pass the object to the ax paramter.
-

```
fig, axes = plt.subplots(2,3, figsize=(16,8))
df.plot(ax=axes[0][0])
df['col1'].plot(ax=axes[0][1])
df['col2'].plot(ax=axes[1][1])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1eef0f05d0>
```



CUSTOMIZING SUB PLOTS

First, create a Subplot with 4 figures = [0][0] to [1][1]

- [0][0] = 1st Figure
    - Plot only the 'col2'
    - Changing the line color to purple
    - Changing the Font Size of the Axis

- [0][1] = 2nd Figure
    - For "df", the index column is from a – j
    - if we enumerate(df['col2'].index, we would obtain...
    - a: 0
    - b: 1
    - c: 2
    - ...

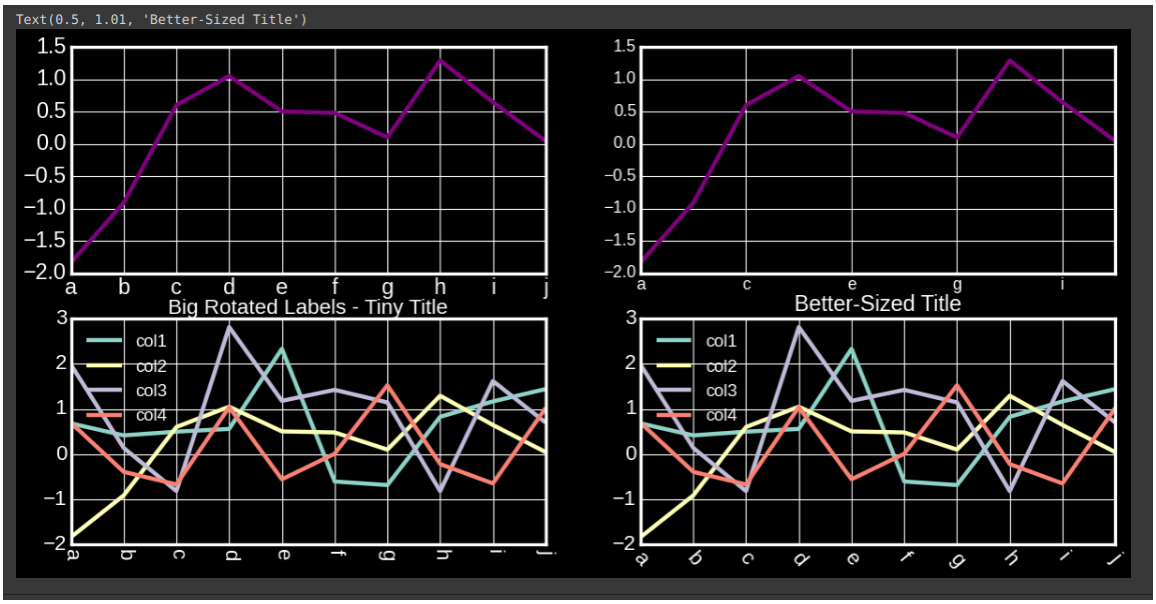
- Presume we only want to show a / c / e / g / i (the even numbered)
- idx % 2 == 0 will help us only obtain the even numbered
- We plot only the the rows a/c/e/g/i

- [1][0] = 3rd Figure
    - Plot all columns
    - Put the Axis Font Size = 20

- Rotate the Axis Font by -90 degrees

- [1][1] = 4th Figure
  - Plot all columns
  - Leave the Axis Font Size = 20
  - Put the Title Font Size = 21
  - Rotate the Axis Font by -50 degrees
  - We use .set_title() to change title size

```
#Create a Subplot
fig, axes = plt.subplots(2,2, figsize=(16,8))

# [0][0]
df['col2'].plot(figsize=(16,4), color='purple', fontsize=21, ax=axes[0][0])

# [0][1]
ticks_to_show = [idx for idx, _ in enumerate(df['col2'].index) if idx % 2 == 0] #even id

df['col2'].plot(figsize=(16,4), color='purple', xticks=ticks_to_show, fontsize=16, ax=axes[0][1])

# [1][0]
df.plot(figsize=(15,7), title='Tiny Title',
        fontsize=20, rot=-90, ax=axes[1][0])\

# [1][1]
df.plot(figsize=(16,8), fontsize=20, rot=-50, ax=axes[1][1])\
        .set_title('Better-Sized Title', fontsize=21, y=1.01)
```



The file can be found here: https://www.alvinang.sg/s/boston_housing_data.csv

STEP 1: LOAD THE DATA

```
housing_csv = 'https://www.alvinang.sg/s/boston_housing_data.csv'
housing = pd.read_csv(housing_csv)
```

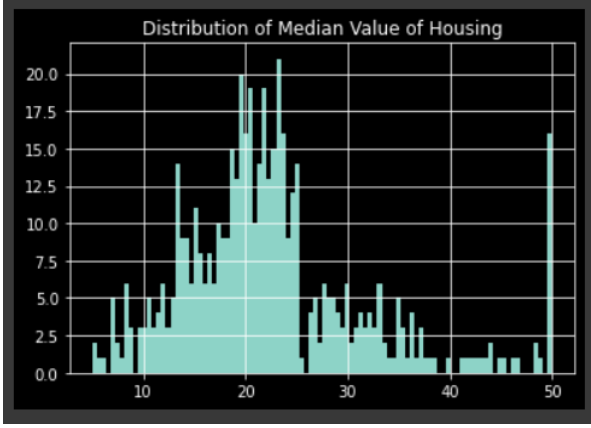
```
housing
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.90	5.33	36.2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
501	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273.0	21.0	391.99	9.67	22.4
502	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273.0	21.0	396.90	9.08	20.6
503	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273.0	21.0	396.90	5.64	23.9
504	0.10959	0.0	11.93	0	0.573	6.794	89.3	2.3889	1	273.0	21.0	393.45	6.48	22.0
505	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273.0	21.0	396.90	7.88	11.9

506 rows × 14 columns

STEP 2: HISTOGRAM

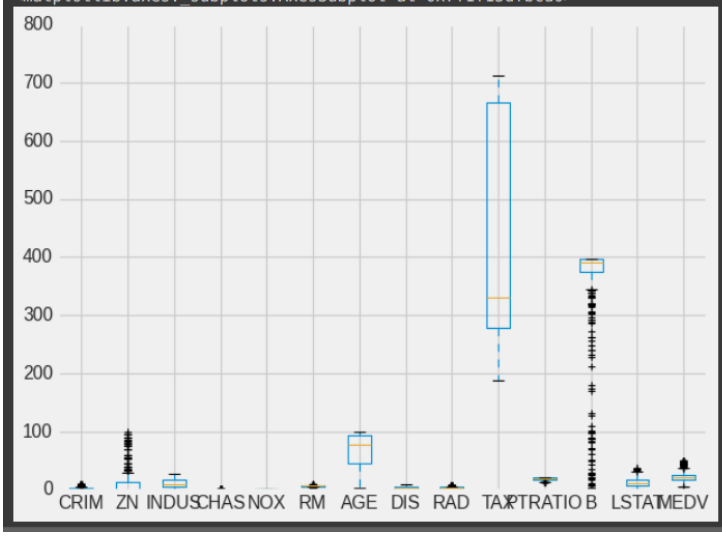
```
housing_MEDV = housing['MEDV'].hist(bins=100)

housing_MEDV.set_title('Distribution of Median Value of Housing');
```



STEP 3: BOXPLOT

```
housing.boxplot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1f15dfbcd0>
```

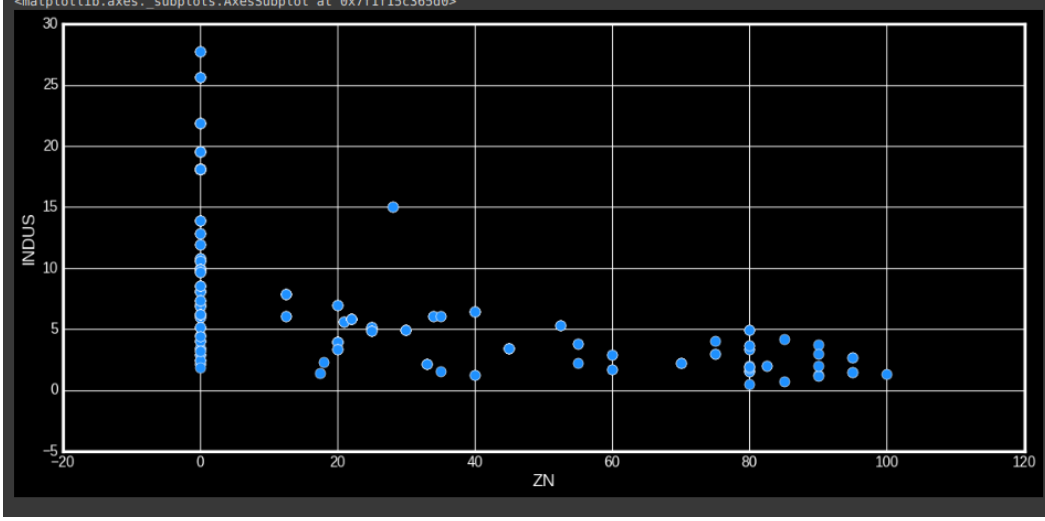


STEP 4: SCATTER PLOT

NORMAL SCATTER PLOT

```
housing.plot(x='ZN', y='INDUS', kind='scatter',
             color='dodgerblue', figsize=(15,7), s=100)
```

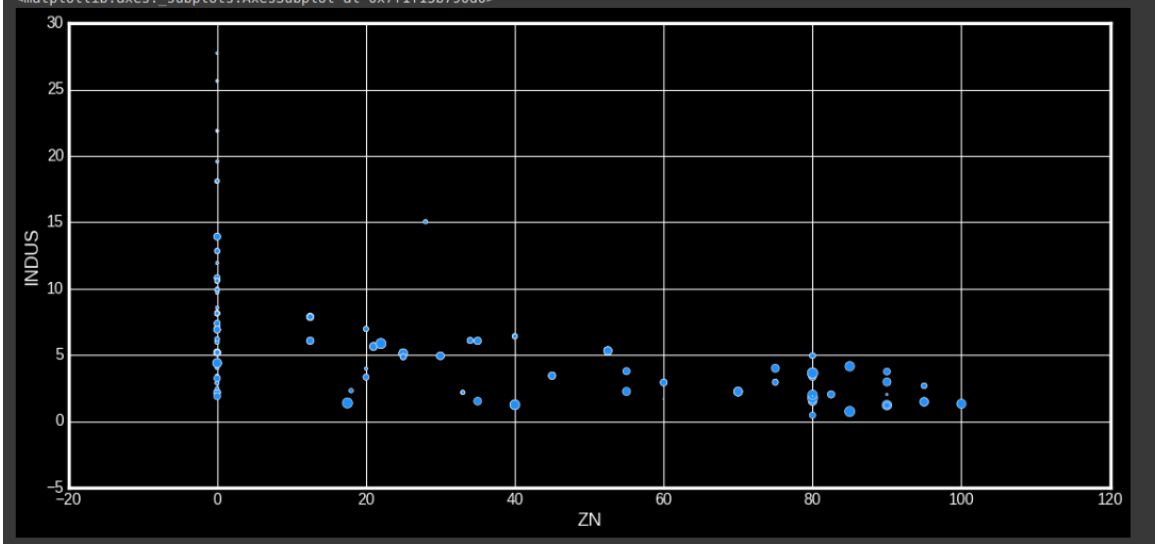
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1f15c365d0>
```



ADJUSTED SCATTER PLOT WITH DIFFERENT CIRCLE SIZES

```
housing.plot(x='ZN', y='INDUS', kind='scatter',
             color='dodgerblue', figsize=(15,7), s=[x**2 for x in housing['DIS']])
```

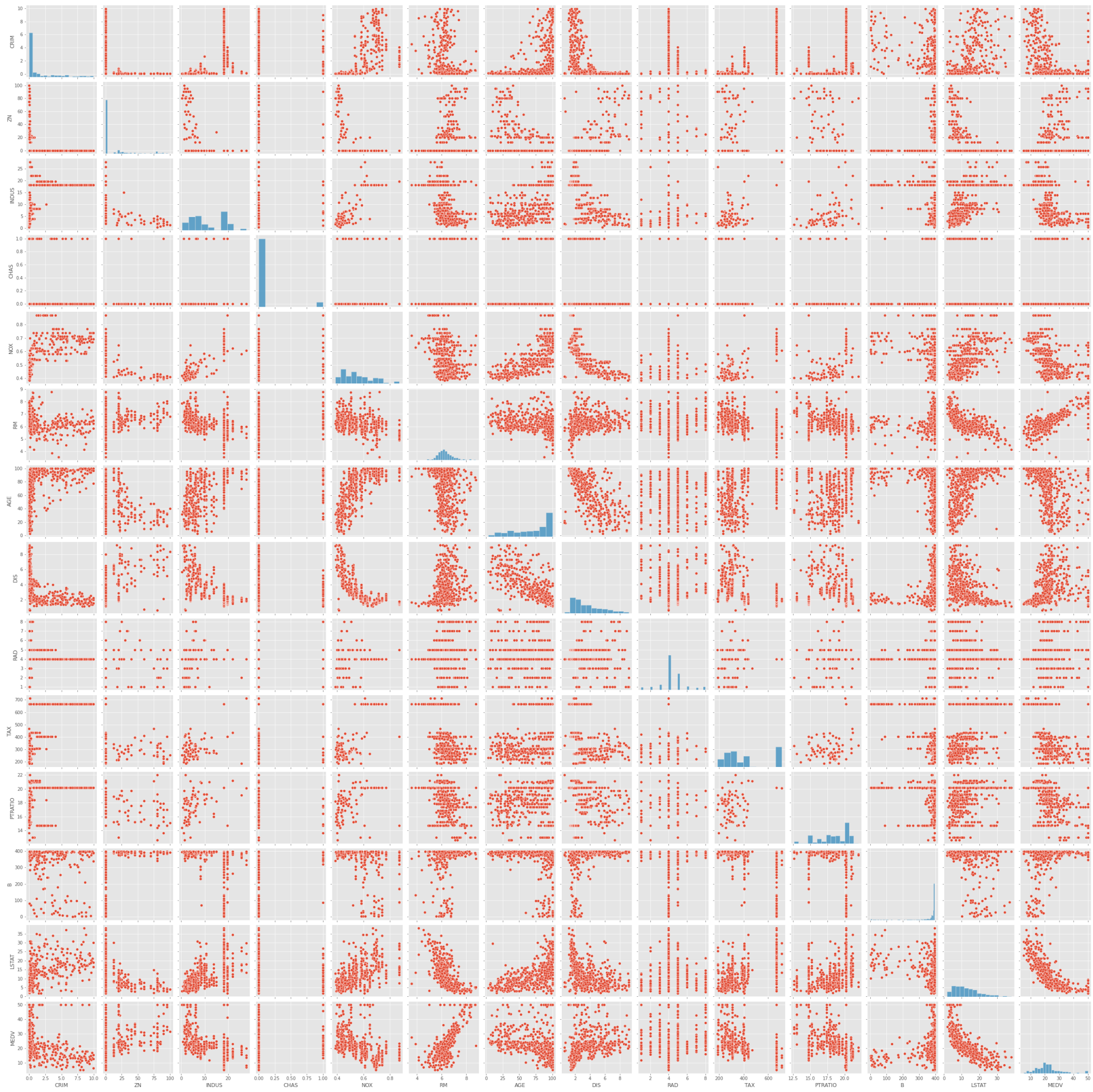
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1f15b790d0>
```

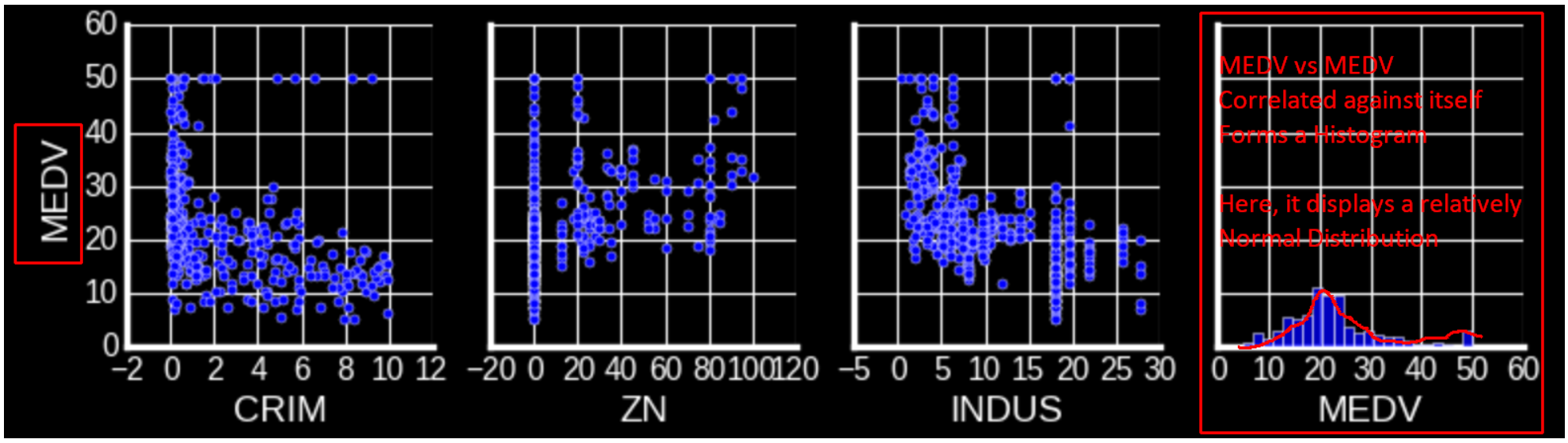


STEP 5: PAIR PLOT

ENTIRE PAIR PLOT

```
plt.style.use('ggplot')

sns.pairplot(housing)
```

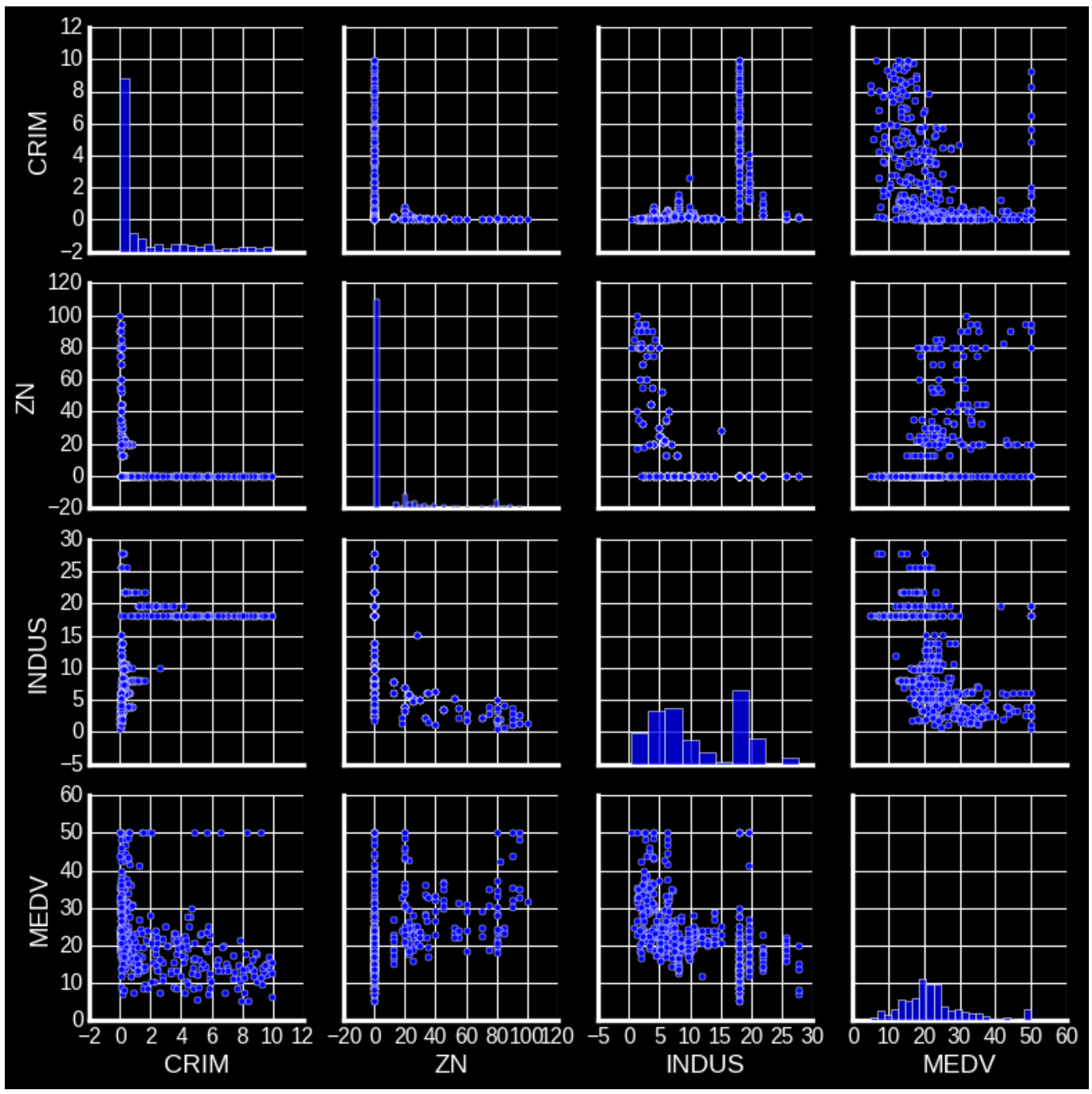




- Note that the Pair Plot, if the variable is plotted against itself
- Example MEDV vs MEDV, it shows instead a Histogram.
- In this case, it displays a Normal Distribution.

SELECTED PAIR PLOT

```
plt.style.use('dark_background')
sns.pairplot(housing, vars=['CRIM', 'ZN', 'INDUS', 'MEDV'])
#sns.pairplot(housing, vars=['CRIM', 'ZN', 'INDUS', 'MEDV'],
#hue = "Outcome", markers=["o", "s"])
```

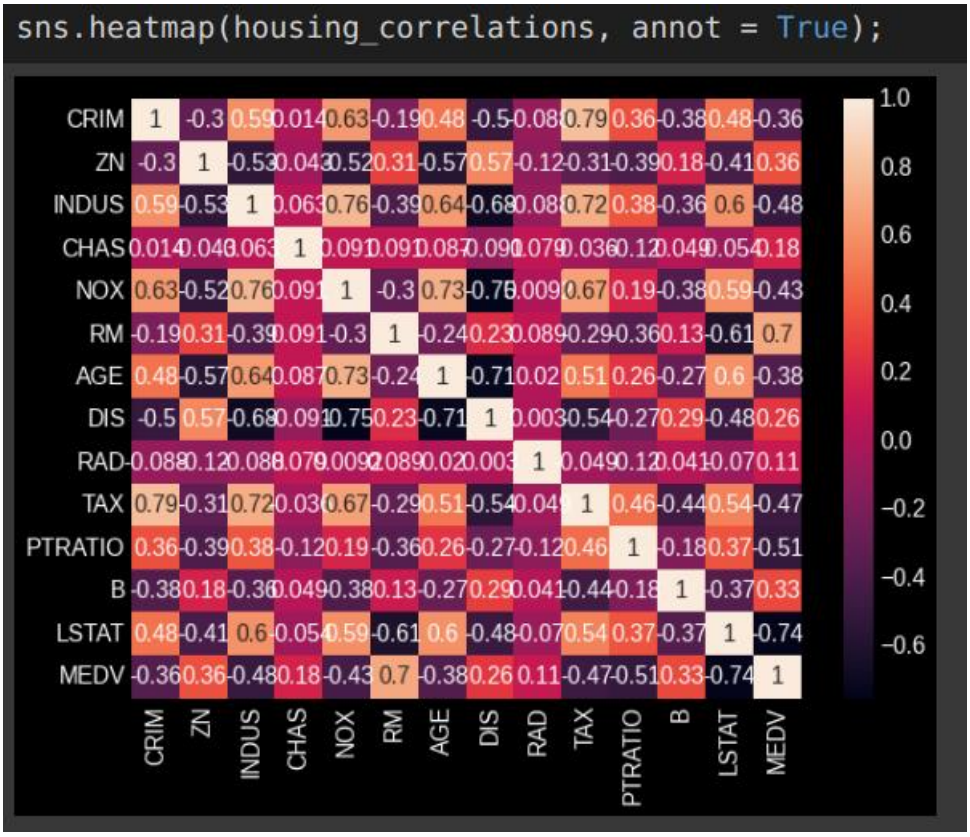


STEP 6: HEAT MAP

```
housing_correlations = housing.corr()
housing_correlations
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
CRIM	1.000000	-0.300774	0.590822	0.013922	0.634679	-0.190197	0.482013	-0.495148	-0.088451	0.793392	0.362615	-0.377013	0.481907	-0.362077
ZN	-0.300774	1.000000	-0.533828	-0.042697	-0.516604	0.311991	-0.569537	0.566660	-0.119290	-0.314563	-0.391679	0.175520	-0.412995	0.360445
INDUS	0.590822	-0.533828	1.000000	0.062938	0.763651	-0.391676	0.644779	-0.678498	-0.087615	0.720760	0.383248	-0.356977	0.603800	-0.483725
CHAS	0.013922	-0.042697	0.062938	1.000000	0.091203	0.091251	0.086518	-0.090950	0.079105	-0.035587	-0.121515	0.048788	-0.053929	0.175260
NOX	0.634679	-0.516604	0.763651	0.091203	1.000000	-0.302188	0.731470	-0.748872	0.009217	0.668023	0.188933	-0.380051	0.590879	-0.427321
RM	-0.190197	0.311991	-0.391676	0.091251	-0.302188	1.000000	-0.240265	0.225052	0.088753	-0.292048	-0.355501	0.128069	-0.613808	0.695360
AGE	0.482013	-0.569537	0.644779	0.086518	0.731470	-0.240265	1.000000	-0.713313	0.019658	0.506456	0.261515	-0.273534	0.602339	-0.376955
DIS	-0.495148	0.566660	-0.678498	-0.090950	-0.748872	0.225052	-0.713313	1.000000	0.003030	-0.541369	-0.269140	0.293621	-0.479158	0.264325
RAD	-0.088451	-0.119290	-0.087615	0.079105	0.009217	0.088753	0.019658	0.003030	1.000000	-0.049221	-0.116969	0.040705	-0.069828	0.113519
TAX	0.793392	-0.314563	0.720760	-0.035587	0.668023	-0.292048	0.506456	-0.541369	-0.049221	1.000000	0.460853	-0.441808	0.543993	-0.468536
PTRATIO	0.362615	-0.391679	0.383248	-0.121515	0.188933	-0.355501	0.261515	-0.269140	-0.116969	0.460853	1.000000	-0.177383	0.374044	-0.507787
B	-0.377013	0.175520	-0.356977	0.048788	-0.380051	0.128069	-0.273534	0.293621	0.040705	-0.441808	-0.177383	1.000000	-0.366087	0.333461
LSTAT	0.481907	-0.412995	0.603800	-0.053929	0.590879	-0.613808	0.602339	-0.479158	-0.069828	0.543993	0.374044	-0.366087	1.000000	-0.737663
MEDV	-0.362077	0.360445	-0.483725	0.175260	-0.427321	0.695360	-0.376955	0.264325	0.113519	-0.468536	-0.507787	0.333461	-0.737663	1.000000

```
sns.heatmap(housing_correlations, annot = True);
```



MTCARS.CSV

The file can be found here: https://www.alvinang.sg/s/mtcars.csv

STEP 1: TAKE A PEEK AT THE DATA

	A	B	C	D	E	F	G	H	I	J	K	L
1	car_names	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
2	Mazda RX4	21	6	160	110	3.9	2.62	16.46	0	1	4	4
3	Mazda RX4 Wag	21	6	160	110	3.9	2.875	17.02	0	1	4	4
4	Datsun 710	22.8	4	108	93	3.85	2.32	18.61	1	1	4	1
5	Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
6	Hornet Sportabout	18.7	8	360	175	3.15	3.44	17.02	0	0	3	2
7	Valiant	18.1	6	225	105	2.76	3.46	20.22	1	0	3	1
8	Duster 360	14.3	8	360	245	3.21	3.57	15.84	0	0	3	4
9	Merc 240D	24.4	4	146.7	62	3.69	3.19	20	1	0	4	2
10	Merc 230	22.8	4	140.8	95	3.92	3.15	22.9	1	0	4	2

STEP 2: READING IN .CSV AND EXPORTING OUT

```
#Read in the .csv
mtcars_sample = pd.read_csv('https://www.alvinang.sg/s/mtcars.csv',
                            index_col='car_names',
                            usecols=['car_names','mpg','hp','cyl','am'])
mtcars_sample

#You can Export out to .csv
mtcars_sample.to_csv('cars_sample.csv')

#You can Export out to .xls
mtcars_sample.to_excel('cars_sample.xlsx', sheet_name='cars', index=False)

#You can also Read in the .xls
mtcars_sample2 = pd.read_excel('cars_sample.xlsx', sheet_name='cars')
```

DISPLAYING THE .CSV

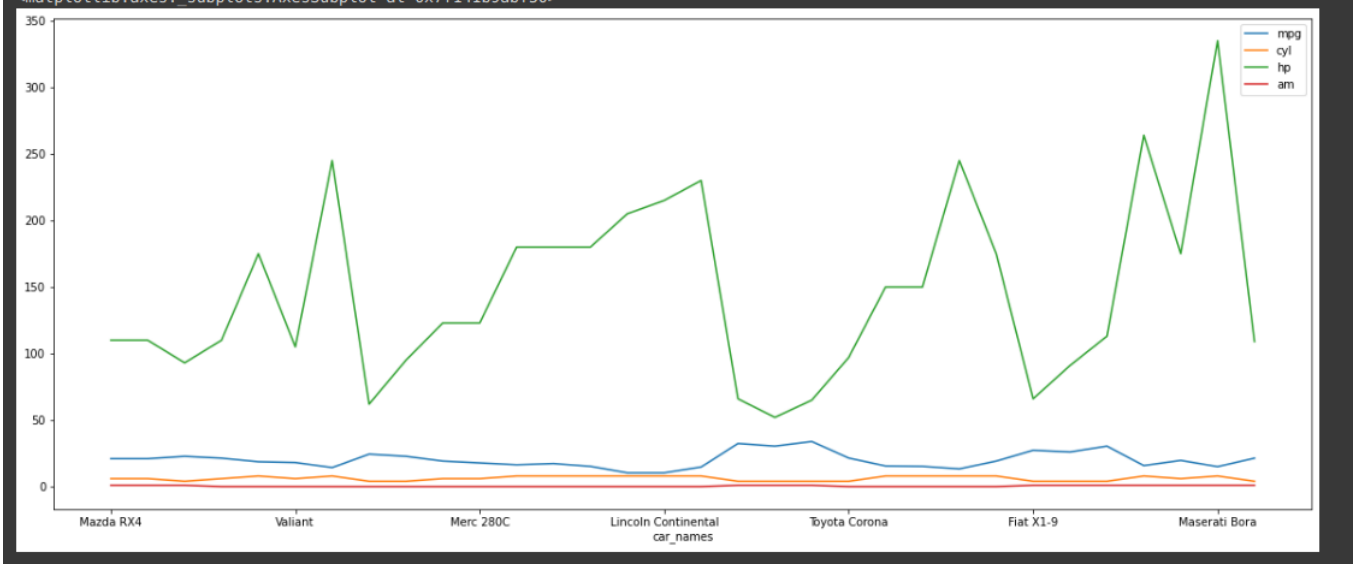
```
#Read in the .csv
mtcars_sample = pd.read_csv('https://www.alvinang.sg/s/mtcars.csv',
                            index_col='car_names',
                            usecols=['car_names','mpg','hp','cyl','am'])
mtcars_sample
```

car_names	mpg	cyl	hp	am
Mazda RX4	21.0	6	110	1
Mazda RX4 Wag	21.0	6	110	1
Datsun 710	22.8	4	93	1
Hornet 4 Drive	21.4	6	110	0
Hornet Sportabout	18.7	8	175	0
Valiant	18.1	6	105	0

STEP 3: SIMPLE LINE PLOT

```
mtcars_sample.plot(figsize=(20,8))
```

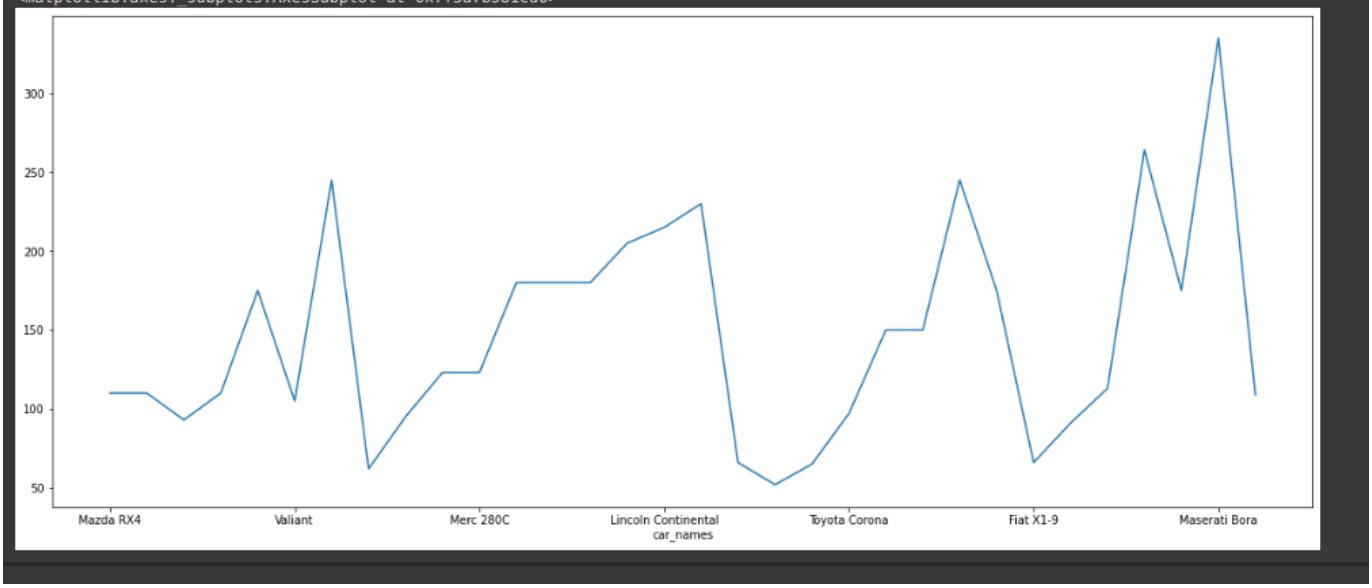
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f141b9abf50>
```



LINE PLOT FOR ONE COLUMN ONLY

```
mtcars_sample['hp'].plot(figsize=(20,8))
```

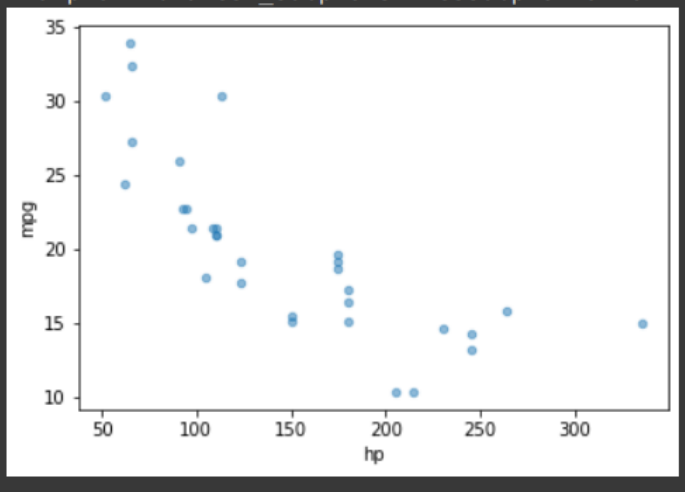
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5a7b581cd0>
```



STEP 4: SCATTER PLOT

```
mtcars_sample.plot.scatter(x="hp", y="mpg",alpha=0.5)
```

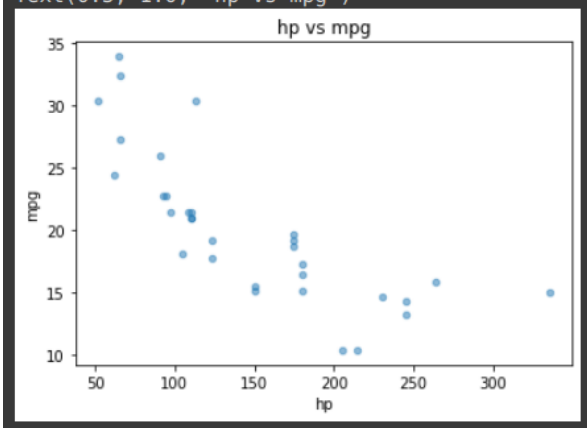
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fda56545e10>
```



PANDAS: SCATTER PLOT

```
mtcars_sample.plot.scatter(x="hp", y="mpg",alpha=0.5)
plt.xlabel("hp")
plt.ylabel("mpg")
plt.title("hp vs mpg")
```

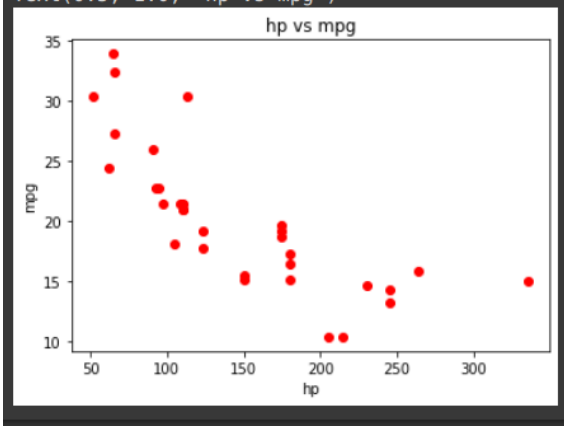
```
Text(0.5, 1.0, 'hp vs mpg')
```



MATPLOTLIB: SCATTER PLOT

```
plt.scatter(x="hp", y="mpg", data=mtcars_sample,color='red')
plt.xlabel('hp')
plt.ylabel('mpg')
plt.title('hp vs mpg')
```

```
Text(0.5, 1.0, 'hp vs mpg')
```

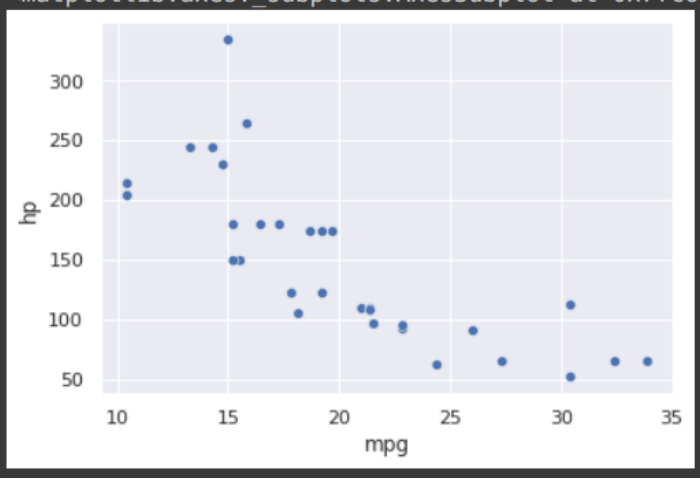


SEABORN: SCATTERPLOT

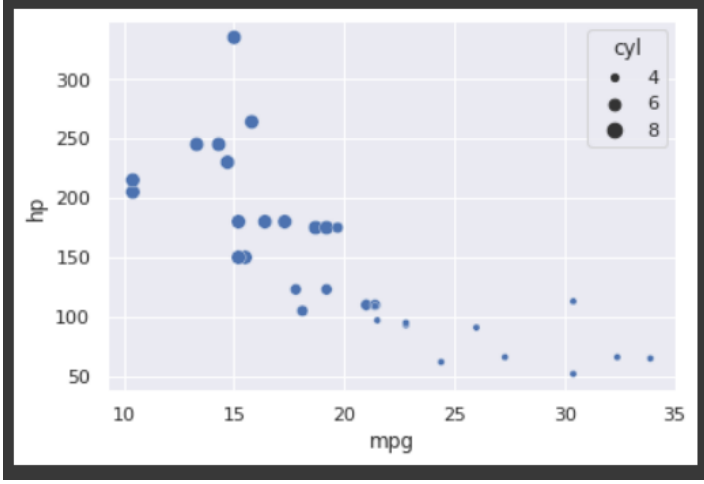
(note that now SNS is seaborn!)

```
import seaborn as sns
sns.scatterplot(x="mpg", y="hp", data=mtcars_sample)
```

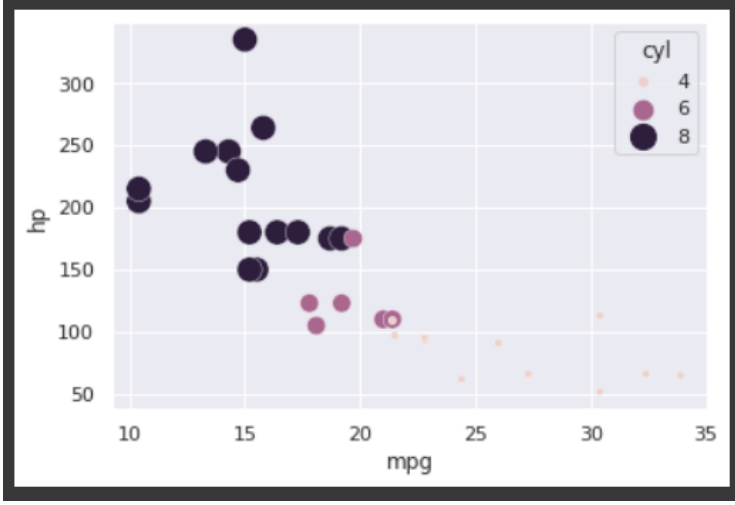
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc6499bfe10>
```



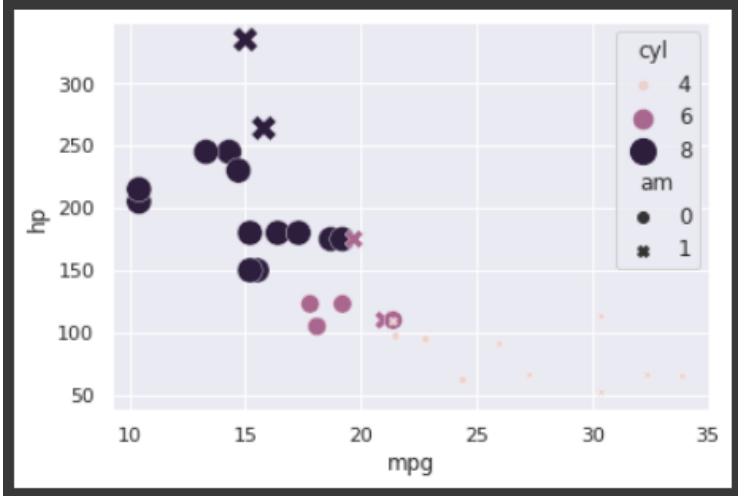
```
sns.scatterplot(x="mpg", y="hp", size="cyl",data=mtcars_sample);
```




```
sns.scatterplot(x="mpg", y="hp",
                hue="cyl", size="cyl",
                sizes=(20, 200),data=mtcars_sample);
```

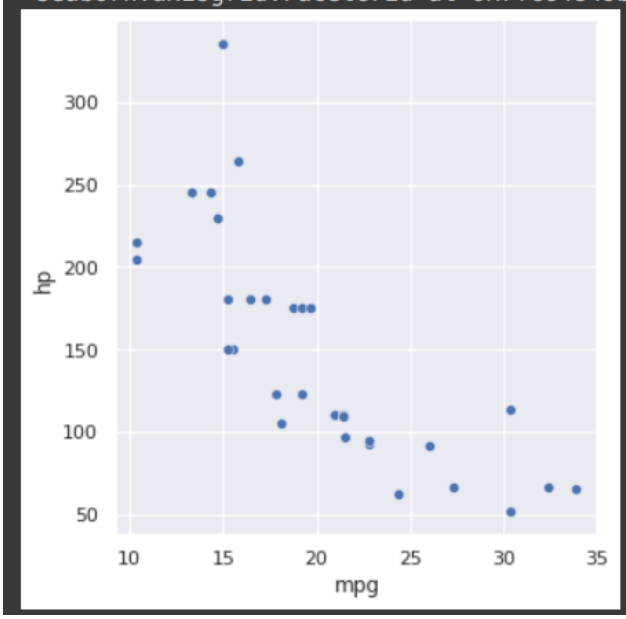


```
cmap = sns.cubehelix_palette(dark=.5, light=0.9, as_cmap=True)
sns.scatterplot(x="mpg", y="hp",
                hue="cyl", size="cyl",sizes=(20, 200),
                style='am',data=mtcars_sample);
```

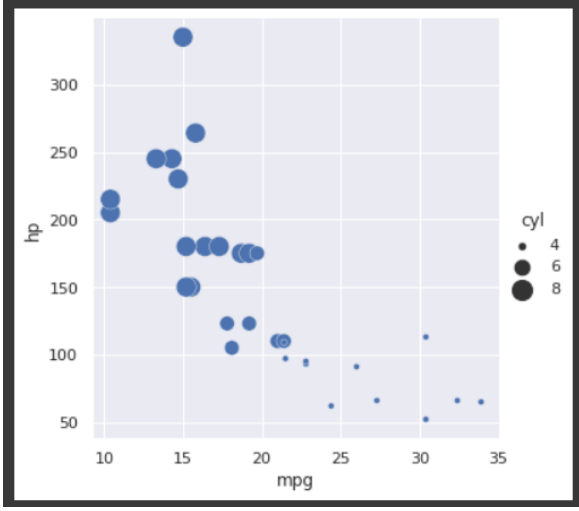


```
sns.relplot(x="mpg", y="hp", data=mtcars_sample)
```

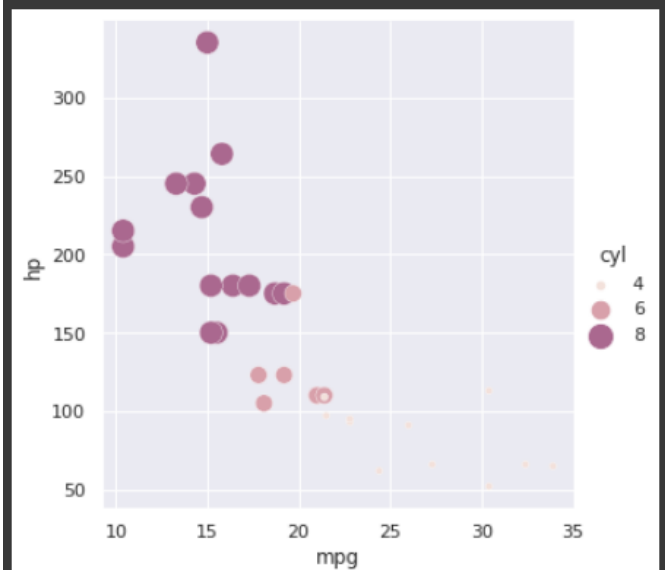
```
<seaborn.axisgrid.FacetGrid at 0x7fc64849b490>
```



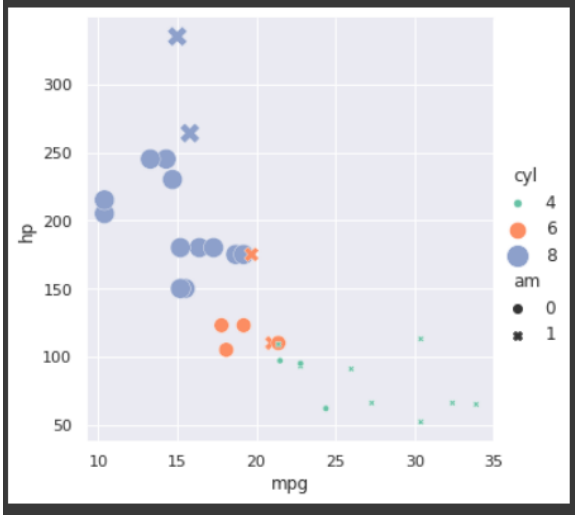

```
sns.relplot(x="mpg", y="hp", size="cyl",sizes=(20, 200),
            data=mtcars_sample);
```



```
cmap = sns.cubehelix_palette(dark=.5, light=0.9, as_cmap=True)
sns.relplot(x="mpg", y="hp", hue="cyl",
            size="cyl",sizes=(20, 200),
            palette=cmap,data=mtcars_sample);
```




```
sns.relplot(x="mpg", y="hp", hue="cyl", size="cyl",sizes=(20, 200),style='am',
            palette='Set2',data=mtcars_sample);
```

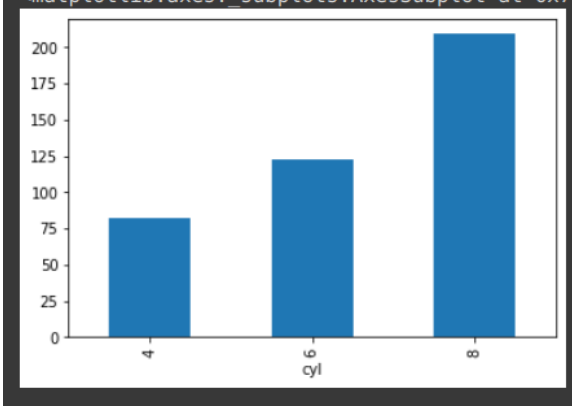


STEP 5: BAR PLOT

BAR PLOT USING PANDAS

```
mtcars_sample.pivot(columns='cyl',values='hp').mean().plot.bar()
```

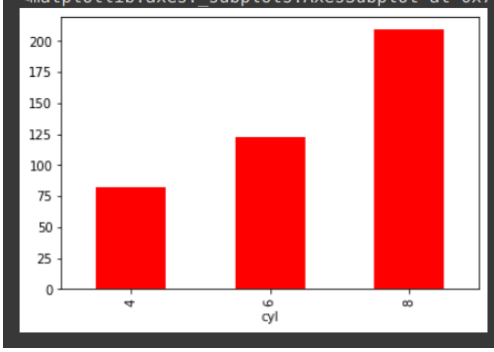
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5a79801e90>
```



Pandas: Bar Plot Change Color

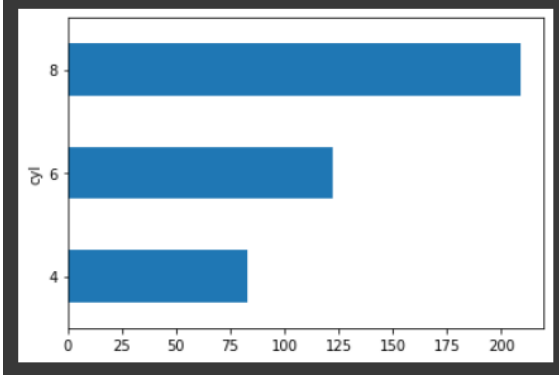
```
mtcars_sample.pivot(columns='cyl',values='hp').mean().plot.bar(color='red')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5a796d5990>
```



Pandas: Horizontal bar Plot

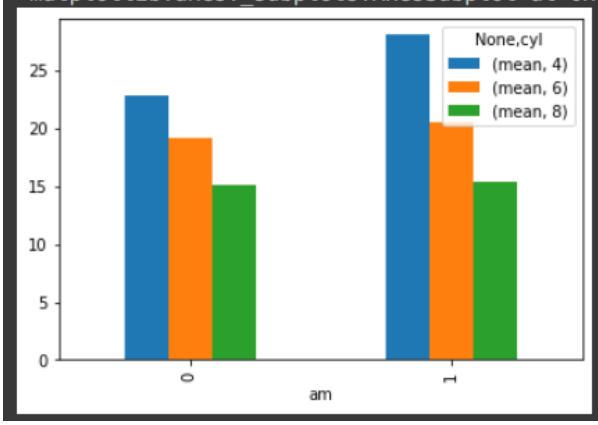
```
mtcars_cyl= mtcars_sample.pivot(columns='cyl',values='hp').mean().plot.barh()
```



Pandas: Double Bar Plots

```
mtcars_sample.pivot_table(
    index='am',columns='cyl',values='mpg',aggfunc=['mean']).plot.bar()
```

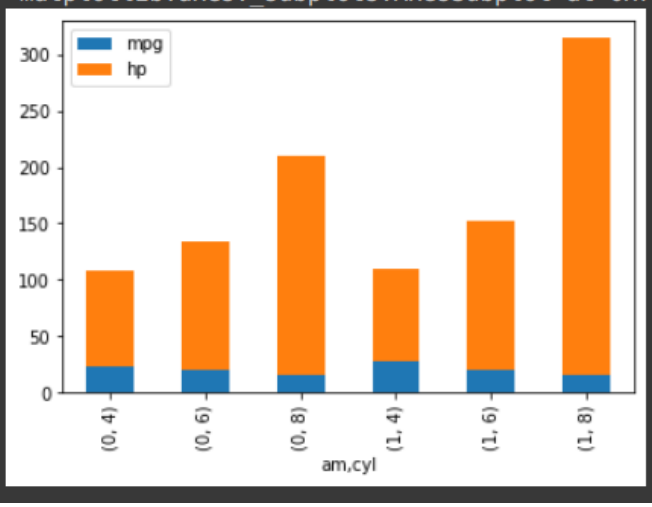
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5a79584f10>
```



Pandas: Stacked Bar Plot

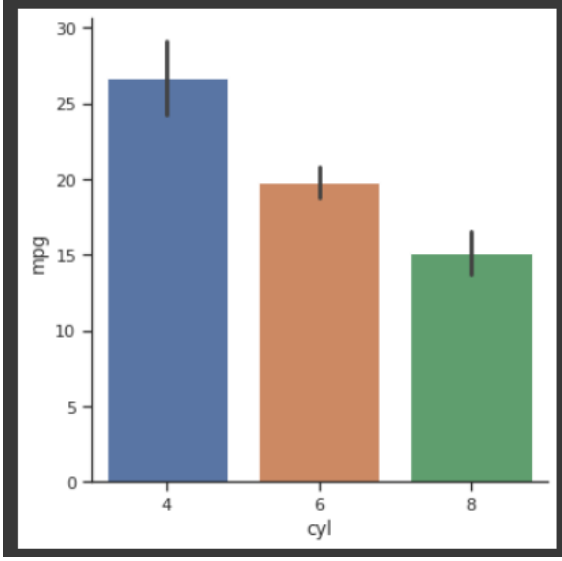
```
mtcars_sample.groupby(['am','cyl']).mean().plot.bar(stacked=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5a783e9510>
```



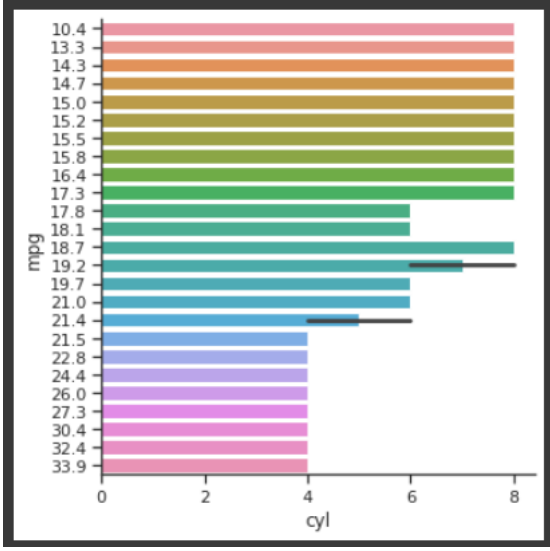
BAR PLOT USING SEABORN CATEGORY PLOT

```
sns.catplot(x="cyl", y="mpg", kind='bar',data=mtcars_sample);
```



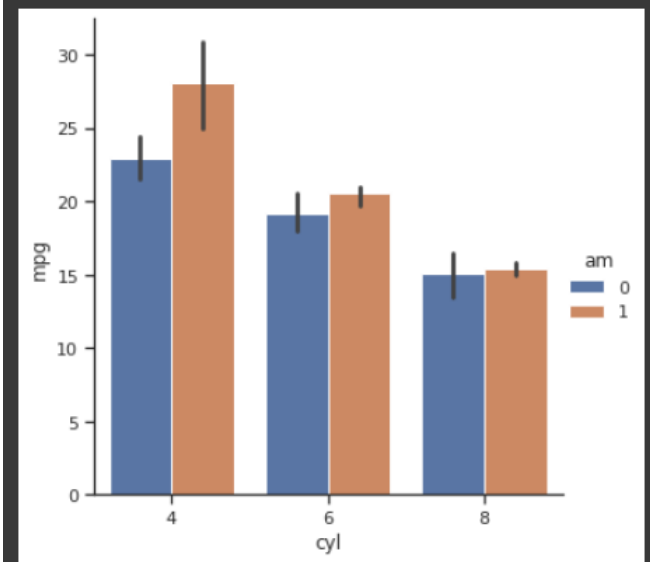
Seaborn: Horizontal Bar Plot

```
sns.catplot(x="cyl", y="mpg", orient='h',kind='bar',data=mtcars_sample);
```



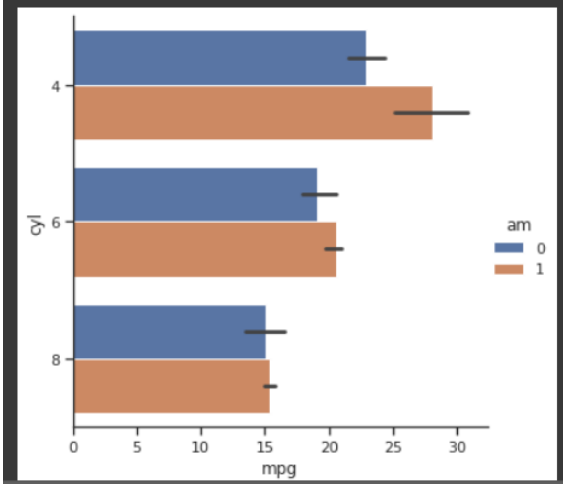
Seaborn: Bar Plot with 3 Variables

```
sns.catplot(x="cyl", y="mpg", kind='bar',hue='am',data=mtcars_sample);
```



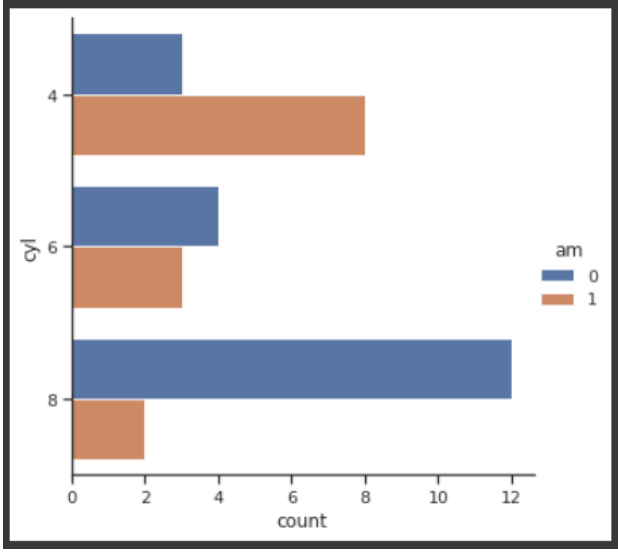
Seaborn: Horizontal Bar Plot with 3 Variables

```
sns.catplot(y="cyl", x="mpg", orient='h',kind='bar',hue='am',data=mtcars_sample);
```



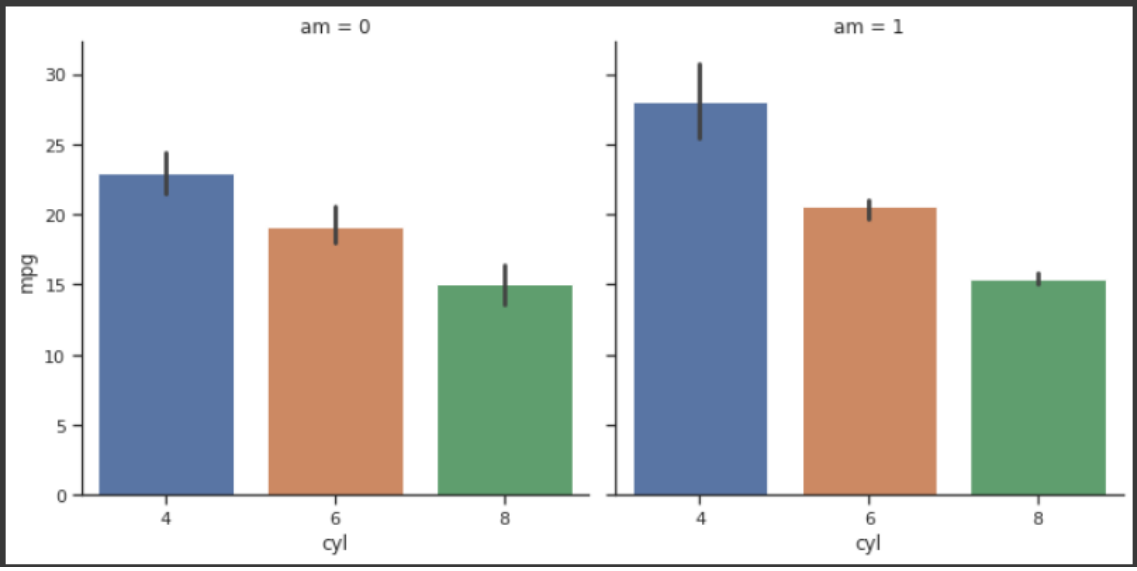
Seaborn: Horizontal Bar Plot of Counts

```
sns.catplot(y="cyl", kind='count',hue='am',data=mtcars_sample);
```



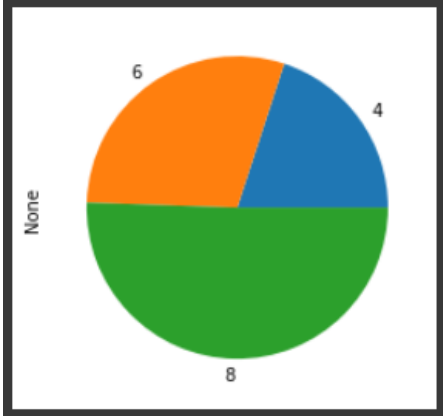
Seaborn: Double Bar Plots

```
sns.catplot(x="cyl", y='mpg', kind='bar',col='am',data=mtcars_sample);
```



STEP 6: PIE CHART

```
mtcars_cyl=
mtcars_sample.pivot(columns='cyl',values='hp').mean().plot.pie()
```

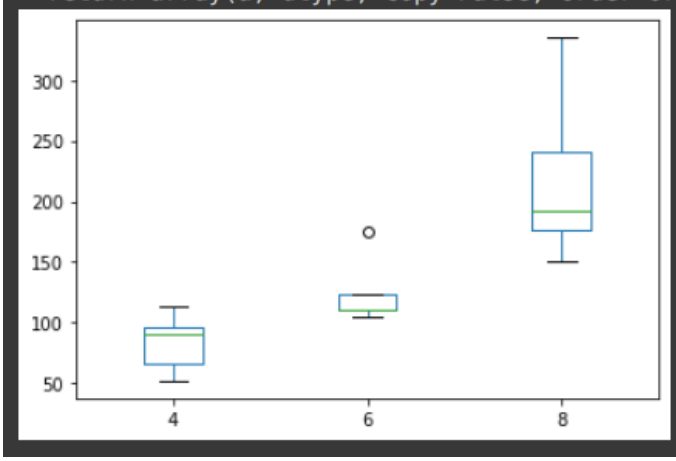


STEP 7: BOX PLOT

PANDAS: BOX PLOT

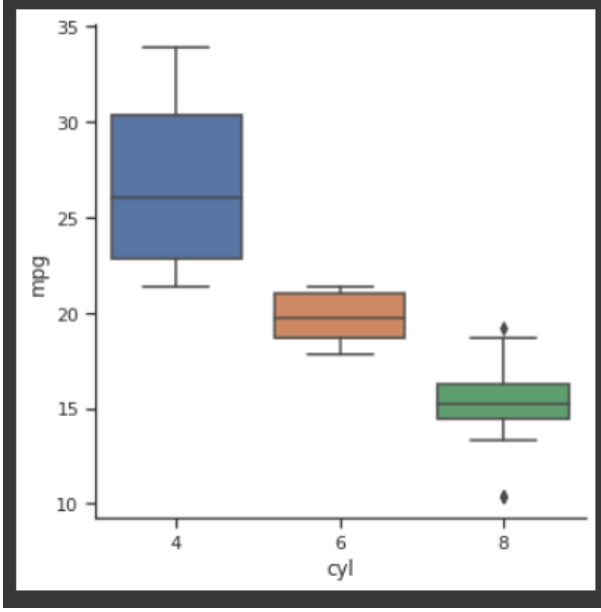
```
mtcars_cyl=
mtcars_sample.pivot(columns='cyl',values='hp').plot.box()
```

```
/usr/local/lib/python3.7/dist-packages/numpy/core/_asarray.py:83: VisibleDeprecation
  return array(a, dtype, copy=False, order=order)
```



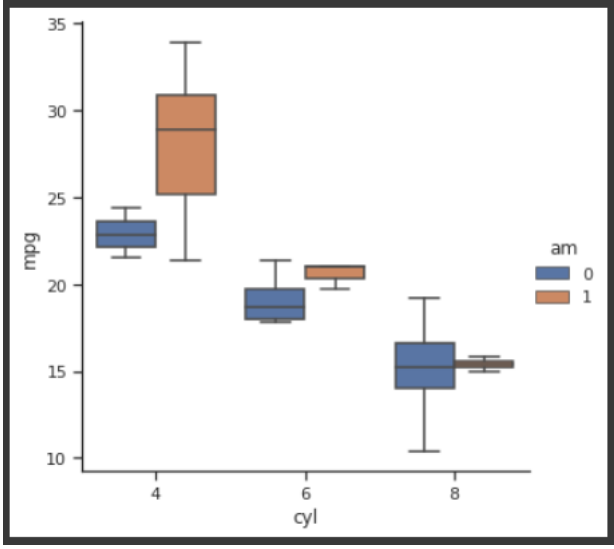
SEABORN: BOX PLOT (2 VARIABLES)

```
sns.catplot(x="cyl", y="mpg", kind='box',data=mtcars_sample);
```



SEABORN: BOX PLOT (3 VARIABLES)

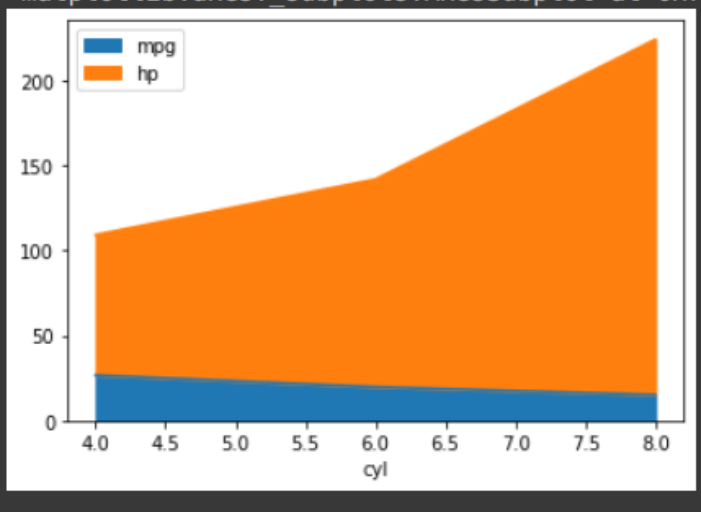
```
sns.catplot(x="cyl", y="mpg", kind='box',hue='am',data=mtcars_sample);
```



STEP 8: AREA PLOT

```
mtcars_sample2 = mtcars_sample[['cyl','mpg','hp']]
mtcars_sample2.groupby(['cyl']).mean().plot.area()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5a7820c590>
```

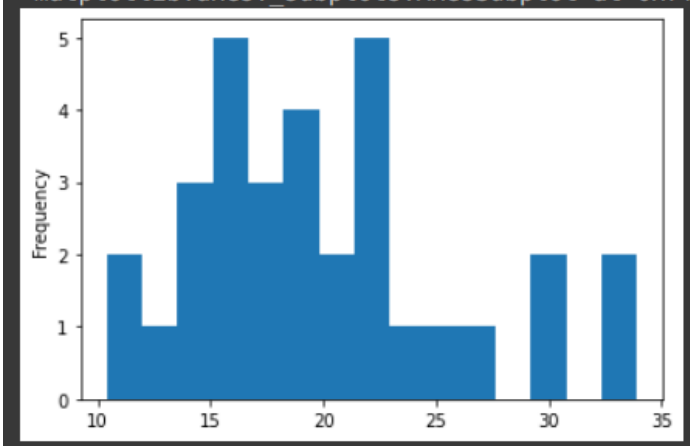


STEP 9: HISTOGRAM

PANDAS: HISTOGRAM

```
mtcars_sample.mpg.plot.hist(bins=15)
```

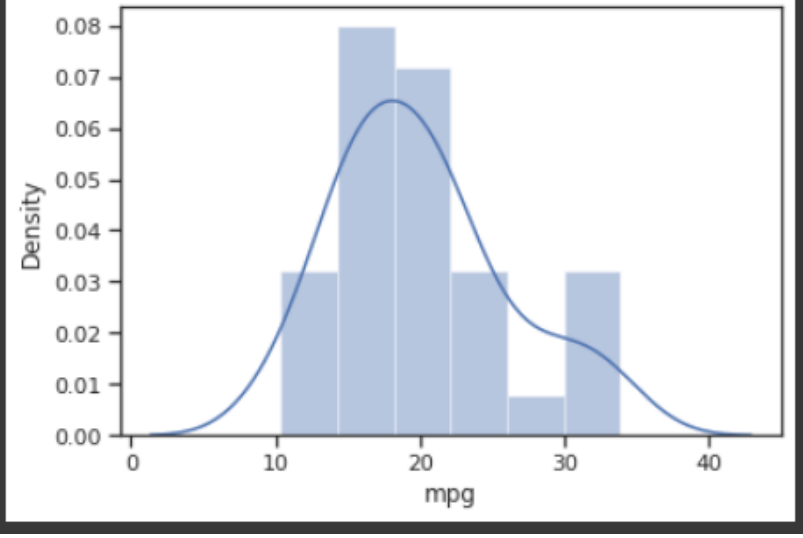
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5a7827c6d0>
```



SEABORN: HISTOGRAM WITH LINE

```
sns.distplot(mtcars_sample.mpg);
```

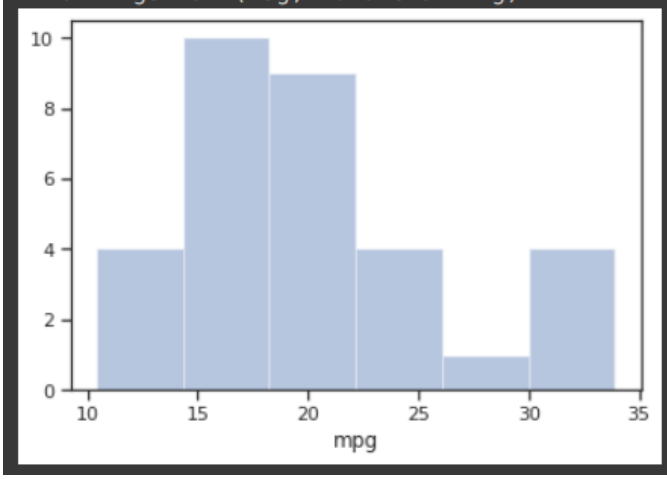
```
/usr/local/lib/python3.7/dist-packages/seaborn/dist
  warnings.warn(msg, FutureWarning)
```



SEABORN: HISTOGRAM WITHOUT LINE

```
sns.distplot(mtcars_sample.mpg,kde=False);
```

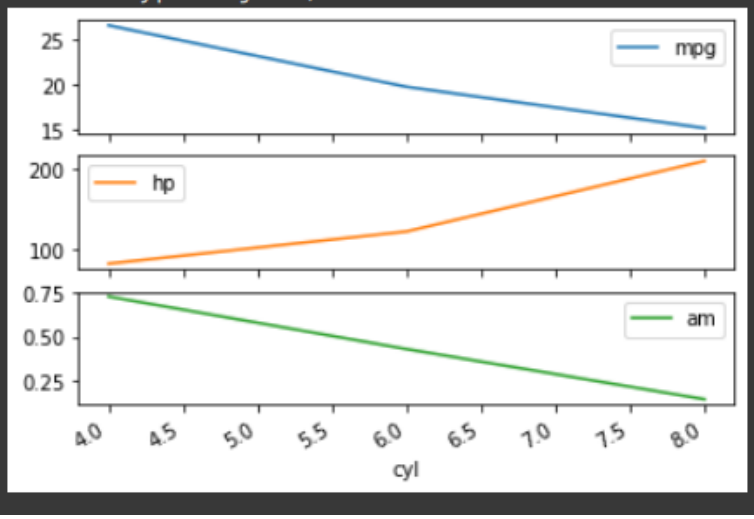
```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.
  warnings.warn(msg, FutureWarning)
```



STEP 10: GROUPBY SUB PLOT

```
mtcars_sample.groupby('cyl').mean().plot(subplots=True)
```

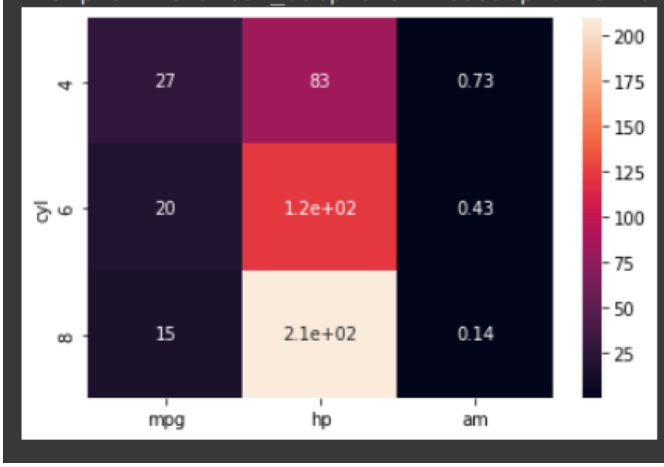
```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x7f5a78270c50>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x7f5a7830d5d0>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x7f5a79599690>],
      dtype=object)
```



STEP 11: HEAT MAP (SEABORN)

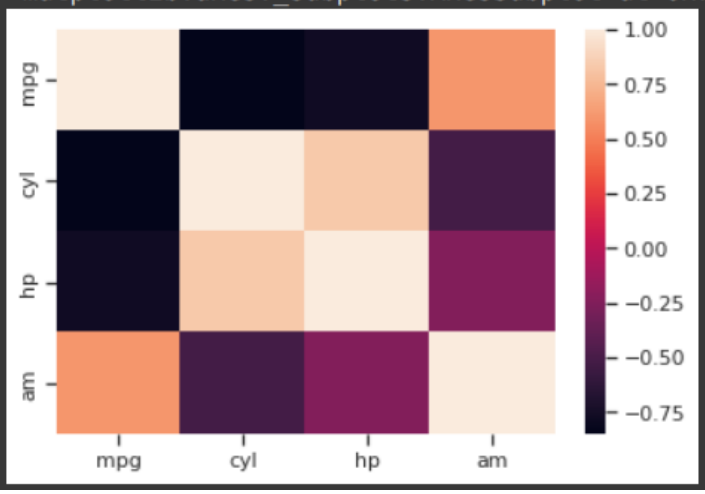
```
mtcars_cyl= mtcars_sample.groupby('cyl').mean()
sb.heatmap(mtcars_cyl, annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5a78095410>
```




```
corr = mtcars_sample.corr()
sns.heatmap(corr)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc0
```

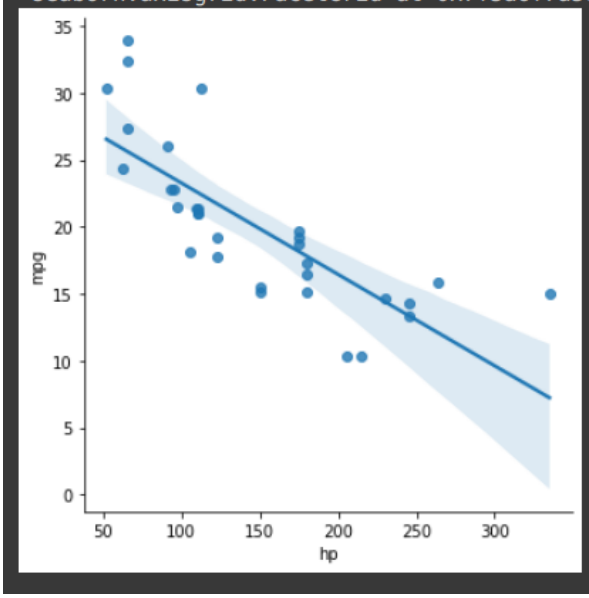


STEP 12: LINEAR REGRESSION PLOT

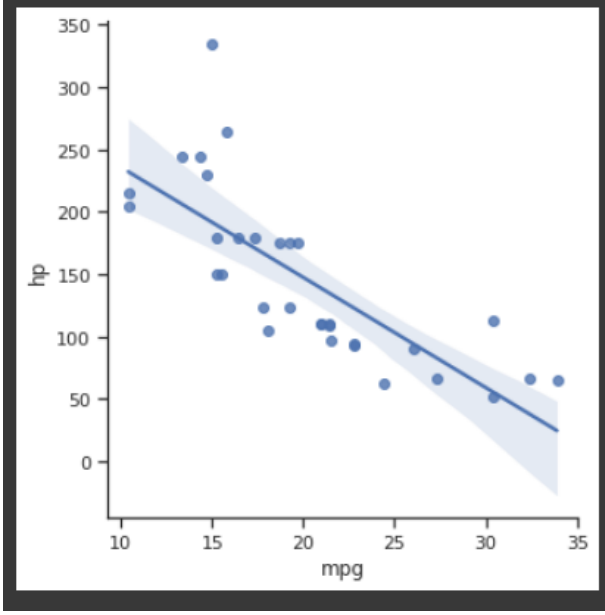
SEABORN: LMPLOT

```
sb.lmplot(x="hp", y="mpg",data=mtcars_sample,fit_reg=True)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f5a6f7a5690>
```

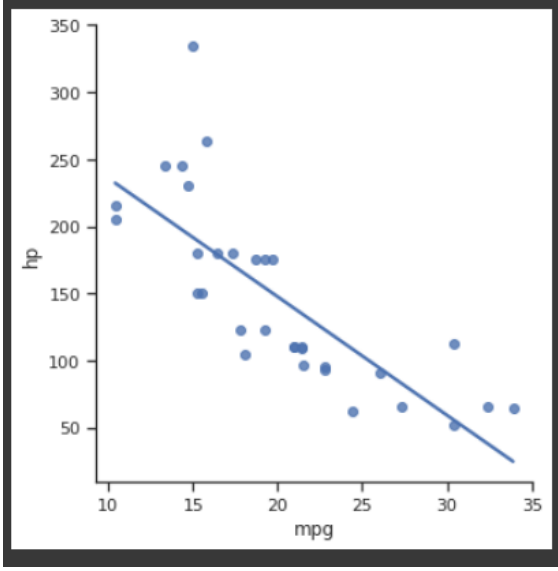



```
sns.lmplot(x="mpg", y="hp", data=mtcars_sample);
```



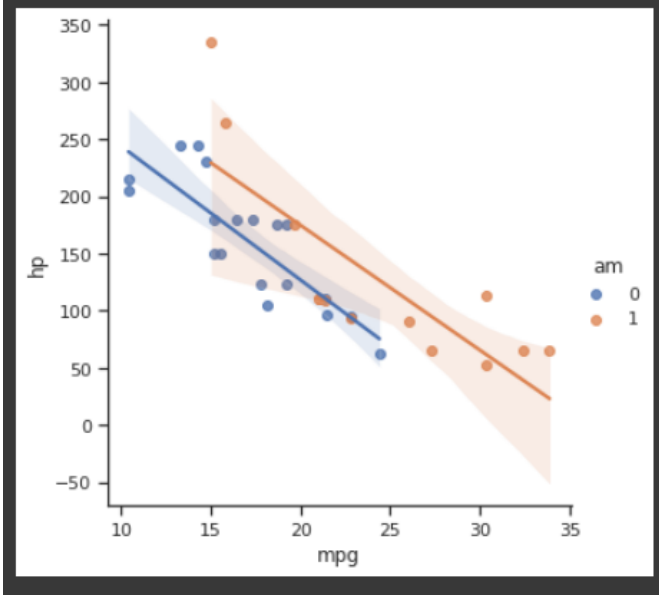
Seaborn: lmplot with No CI

```
sns.lmplot(x="mpg", y="hp", ci=None, data=mtcars_sample);
```



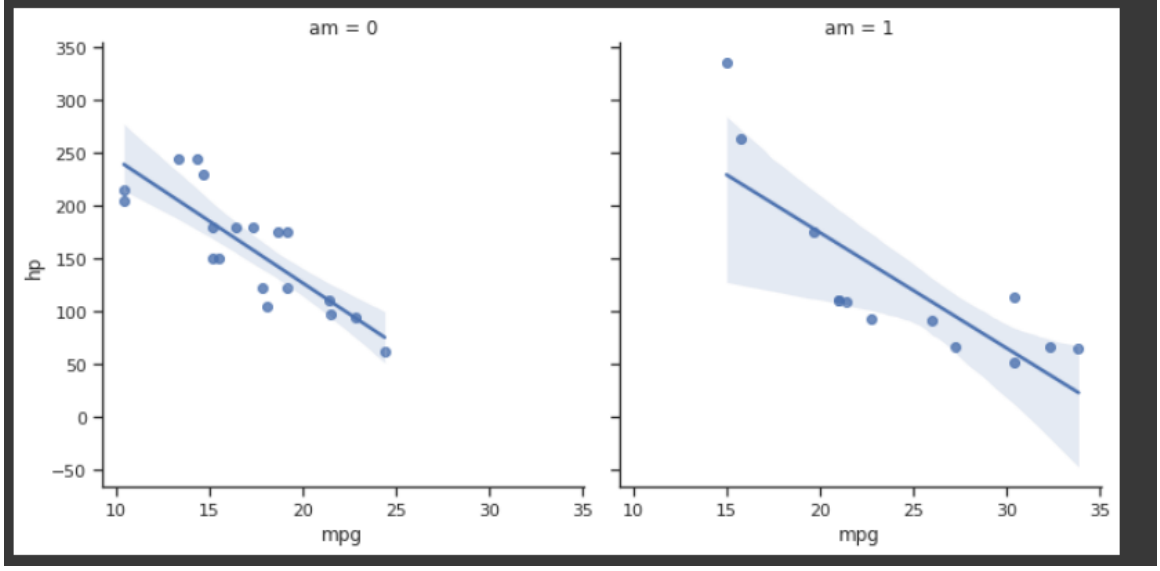
Seaborn: lmplot with 3 Variables

```
sns.lmplot(x="mpg", y="hp", hue="am", data=mtcars_sample);
```



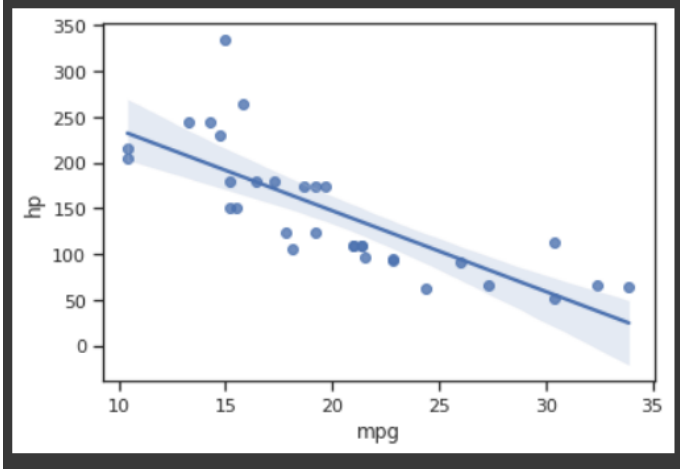
Seaborn: Double lmplot

```
sns.lmplot(x="mpg", y="hp", col="am", data=mtcars_sample);
```



SEABORN: REGPLOT

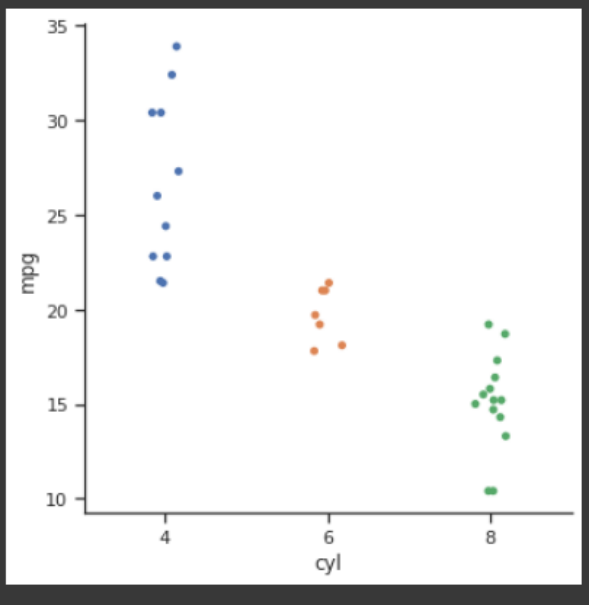
```
sns.regplot(x="mpg", y="hp", data=mtcars_sample);
```



STEP 13: CATEGORY PLOT

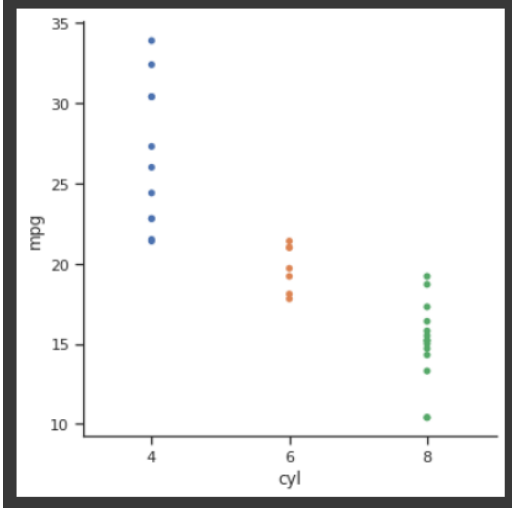
WITH JITTER

```
sns.catplot(x="cyl", y="mpg", data=mtcars_sample);
```



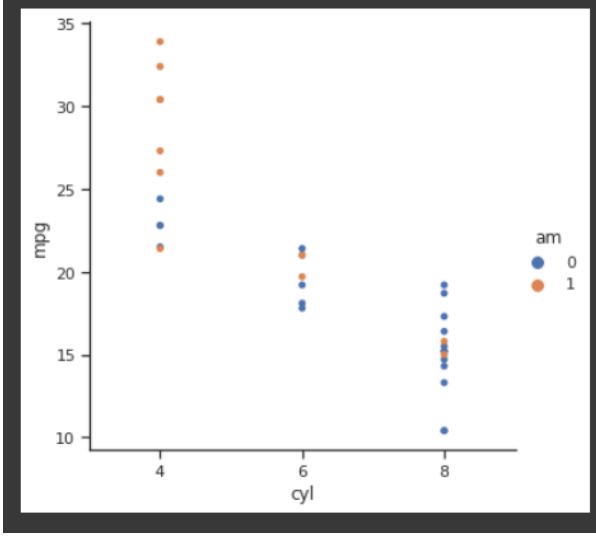
WITHOUT JITTER

```
sns.catplot(x="cyl", y="mpg", jitter=False,data=mtcars_sample);
```



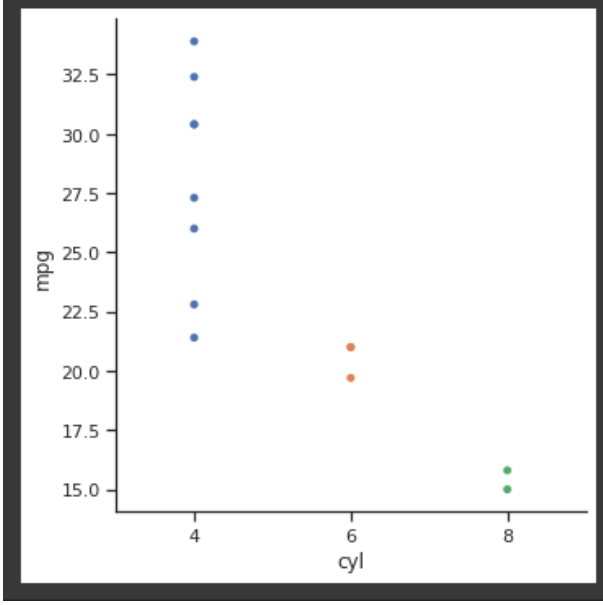
WITH 3 VARIABLES (AM = 0 OR AM = 1)

```
sns.catplot(x="cyl", y="mpg", hue='am',jitter=False,data=mtcars_sample);
```



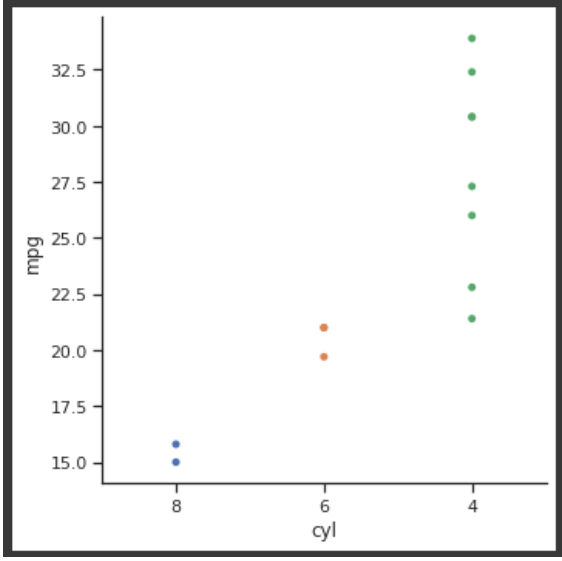
WITH 3 VARIABLES (AM = 1)

```
sns.catplot(x="cyl", y="mpg",
            jitter=False,data=mtcars_sample.query('am==1'));
```



WITH 3 VARIABLES IN PRESET ORDER

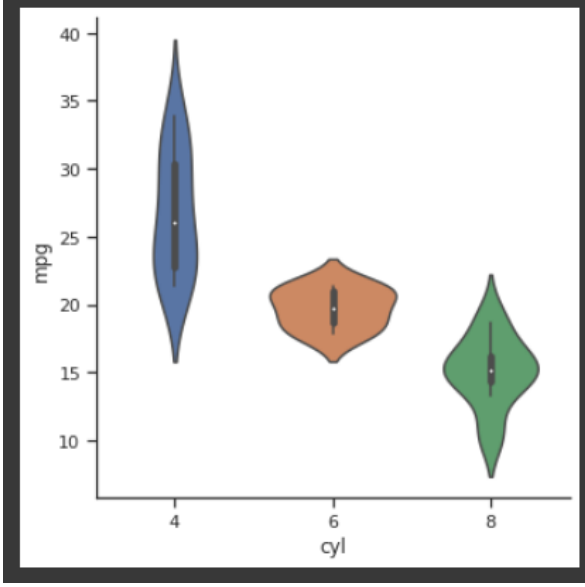
```
sns.catplot(x="cyl", y="mpg", order=[8,6,4],
            jitter=False,data=mtcars_sample.query('am==1'));
```



STEP 14: VIOLIN PLOT

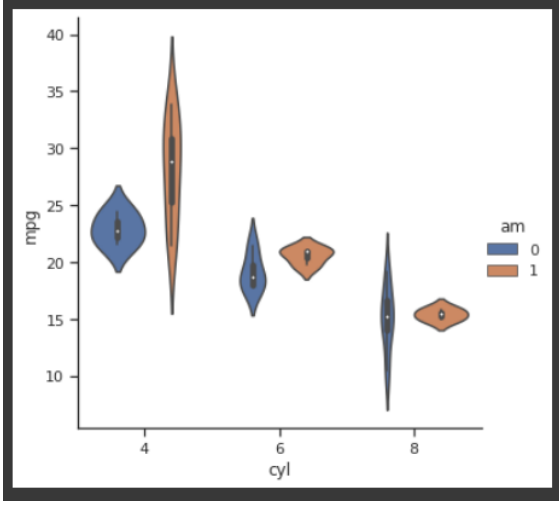
SEABORN: VIOLIN PLOT WITH 2 VARIABLES

```
sns.catplot(x="cyl", y="mpg", kind='violin',data=mtcars_sample);
```



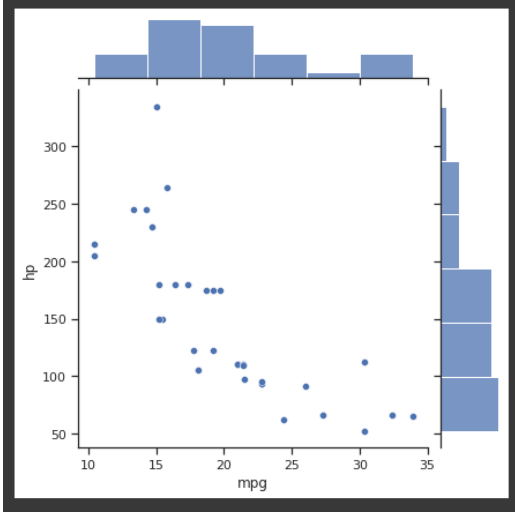
SEABORN: VIOLIN PLOT WITH 3 VARIABLES

```
sns.catplot(x="cyl", y="mpg", kind='violin',hue='am',data=mtcars_sample);
```

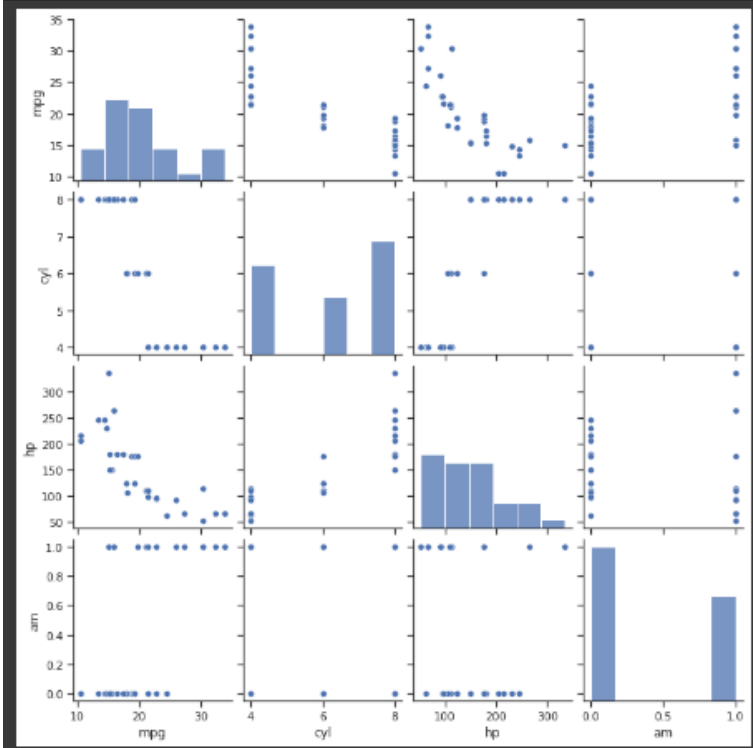


STEP 15: SEABORN PAIR PLOT

```
sns.jointplot(x="mpg", y="hp", data=mtcars_sample);
```




```
sns.pairplot(mtcars_sample);
```



COLLEGE.CSV

The file can be found here: https://www.alvinang.sg/s/college.csv

STEP 1: TAKE A PEEK AT THE DATA

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	id	name	city	state	region	highest_degree	control	gender	admission_rate	sat_avg	undergrads	tuition	faculty_salary_avg	loan_default_rate	median_debt	lon	lat
2	102669	Alaska Pacific University	Anchorage	AK	West	Graduate	Private	CoEd	0.4207	1054	275	19610	5804	0.077	23250	-149.9002778	61.2180556
3	101648	Marion Military Institute	Marion	AL	South	Associate	Public	CoEd	0.6139	1055	433	8778	5916	0.136	11500	-87.3191655	32.6323536
4	100830	Auburn University at Montgomery	Montgomery	AL	South	Graduate	Public	CoEd	0.8017	1009	4304	9080	7255	0.106	21335	-86.2999689	32.3668052
5	101879	University of North Alabama	Florence	AL	South	Graduate	Public	CoEd	0.6788	1029	5485	7412	7424	0.111	21500	-87.677251	34.79981
6	100858	Auburn University	Auburn	AL	South	Graduate	Public	CoEd	0.8347	1215	20514	10200	9487	0.045	21831	-85.4807825	32.6098566
7	100663	University of Alabama at Birmingham	Birmingham	AL	South	Graduate	Public	CoEd	0.8569	1107	11383	7510	9957	0.062	21941.5	-86.80249	33.5206608
8	101480	Jacksonville State University	Jacksonville	AL	South	Graduate	Public	CoEd	0.8326	1041	7060	7092	6801	0.096	23000	-85.7613536	33.8137125
9	102049	Samford University	Birmingham	AL	South	Graduate	Private	CoEd	0.5954	1165	3033	27324	8367	0.007	23000	-86.80249	33.5206608
10	101709	University of Montevallo	Montevallo	AL	South	Graduate	Public	CoEd	0.743	1070	2644	10660	7437	0.103	23266	-86.8641558	33.1006746
11	100751	The University of Alabama	Tuscaloosa	AL	South	Graduate	Public	CoEd	0.5105	1185	29851	9826	9667	0.063	23750	-87.5691735	33.2098407
12	102261	Southeastern Bible College	Birmingham	AL	South	Bachelor	Private	CoEd	1	930	170	11370	4554	0.048	24000	-86.80249	33.5206608
13	100706	University of Alabama in Huntsville	Huntsville	AL	South	Graduate	Public	CoEd	0.8203	1219	5451	9158	9302	0.061	24097	-86.5861037	34.7303688
14	101587	University of West Alabama	Livingston	AL	South	Graduate	Public	CoEd	0.7199	990	1916	8018	6146	0.078	24253	-88.1872475	32.5843025
15	102094	University of South Alabama	Mobile	AL	South	Graduate	Public	CoEd	0.8335	1048	11267	7188	7195	0.075	24711	-88.0398912	30.6953657
16	102368	Troy University	Troy	AL	South	Graduate	Public	CoEd	0.4414	1050	15025	7564	6246	0.114	25000	-85.969951	31.8087678
17	101435	Huntingdon College	Montgomery	AL	South	Bachelor	Private	CoEd	0.5839	1026	1149	24550	5772	0.102	26230	-86.2999689	32.3668052
18	101693	University of Mobile	Mobile	AL	South	Graduate	Private	CoEd	0.5847	1014	1460	19475	4914	0.062	27000	-88.0398912	30.6953657
19	102234	Spring Hill College	Mobile	AL	South	Graduate	Private	CoEd	0.5177	1116	1215	32468	6071	0.066	27000	-88.0398912	30.6953657
20	100937	Birmingham Southern College	Birmingham	AL	South	Bachelor	Private	CoEd	0.5339	1181	1180	31708	7451	0.044	27000	-86.80249	33.5206608
21	101912	Oakwood University	Huntsville	AL	South	Graduate	Private	CoEd	0.4787	928	1878	16720	5147	0.125	27250	-86.5861037	34.7303688
22	101073	Concordia College Alabama	Selma	AL	South	Bachelor	Private	CoEd	0.5328	942	322	10320	5812	0.315	32000	-87.0211007	32.4073589
23	100724	Alabama State University	Montgomery	AL	South	Graduate	Public	CoEd	0.5326	851	4811	8720	6609	0.156	33118.5	-86.2999689	32.3668052
24	102377	Tuskegee University	Tuskegee	AL	South	Graduate	Private	CoEd	0.4922	978	2588	19570	8399	0.128	33500	-85.7077266	32.430237
25	100654	Alabama A & M University	Normal	AL	South	Graduate	Public	CoEd	0.5256	827	4206	9096	6892	0.172	33888	-86.5722237	34.7838409
26	102270	Stillman College	Tuscaloosa	AL	South	Bachelor	Private	CoEd	0.5901	811	1056	15865	4597	0.187	38218	-87.5691735	33.2098407
27	107585	University of Arkansas Community College-Morrilton	Morrilton	AR	South	Associate	Public	CoEd	0.6181	930	1920	2732	4855	0.169	8000.5	-92.7440538	35.1509173
28	107983	Southern Arkansas University Main Campus	Magnolia	AR	South	Graduate	Public	CoEd	0.7142	989	2784	7736	6269	0.18	17000	-93.239334	33.2670725
29	106467	Arkansas Tech University	Russellville	AR	South	Graduate	Public	CoEd	0.8626	1010	8845	5862	6083	0.171	17480	-93.1337856	35.2784173
30	107877	Williams Baptist College	Walnut Ridge	AR	South	Bachelor	Private	CoEd	0.7049	983	508	14360	4720	0.089	19000	-90.9559534	36.0684035
31	106458	Arkansas State University-Main Campus	Jonesboro	AR	South	Graduate	Public	CoEd	0.7239	1088	9139	7720	6927	0.102	19250	-90.704279	35.8422967
32	107071	Henderson State University	Arkadelphia	AR	South	Graduate	Public	CoEd	0.6278	989	3226	7860	5624	0.149	19586	-93.0537839	34.1209292
33	106713	Central Baptist College	Conway	AR	South	Bachelor	Private	CoEd	0.5697	985	788	13800	4847	0.109	21500	-92.4421011	35.0886963
34	106704	University of Central Arkansas	Conway	AR	South	Graduate	Public	CoEd	0.9426	1049	9232	7889	6562	0.091	21500	-92.4421011	35.0886963
35	106397	University of Arkansas	Fayetteville	AR	South	Graduate	Public	CoEd	0.6204	1155	21405	8210	9085	0.063	21500	-94.1718542	36.082156
36	106245	University of Arkansas at Little Rock	Little Rock	AR	South	Graduate	Public	CoEd	0.5928	1048	7942	7432	7881	0.086	21736	-92.2895948	34.7464809
37	107512	Ouachita Baptist University	Arkadelphia	AR	South	Bachelor	Private	CoEd	0.7061	1112	1455	23320	6255	0.068	23250	-93.0537839	34.1209292
38	107558	University of the Ozarks	Clarksville	AR	South	Bachelor	Private	CoEd	0.8892	1061	578	24470	6124	0.175	24200	-93.4665731	35.4714724

STEP 2: READ IN THE .CSV

```
college = pd.read_csv('https://www.alvinang.sg/s/college.csv')
college
```

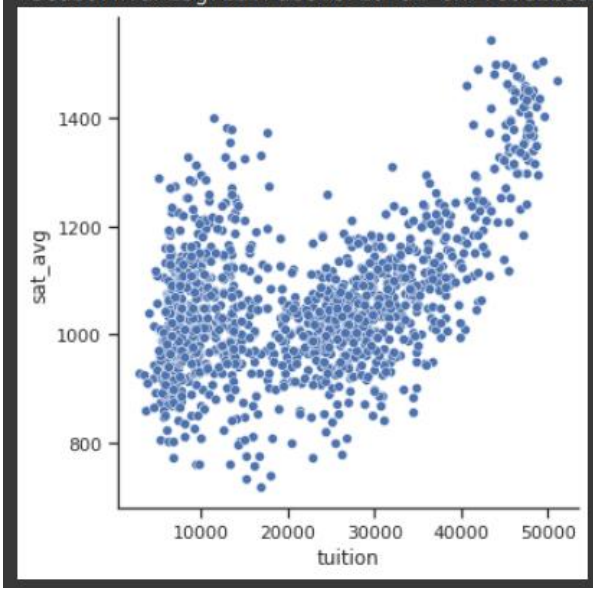
	id	name	city	state	region	highest_degree	control	gender	admission_rate	sat_avg	undergrads
0	102669	Alaska Pacific University	Anchorage	AK	West	Graduate	Private	CoEd	0.4207	1054	275
1	101648	Marion Military Institute	Marion	AL	South	Associate	Public	CoEd	0.6139	1055	433
2	100830	Auburn University at Montgomery	Montgomery	AL	South	Graduate	Public	CoEd	0.8017	1009	4304
3	101879	University of North Alabama	Florence	AL	South	Graduate	Public	CoEd	0.6788	1029	5485
4	100858	Auburn University	Auburn	AL	South	Graduate	Public	CoEd	0.8347	1215	20514

STEP 3: SCATTER PLOT

SEABORN RELATIONAL PLOT (NO HUE CONTROL)

```
sns.relplot(x='tuition', y='sat_avg', data=college)
```

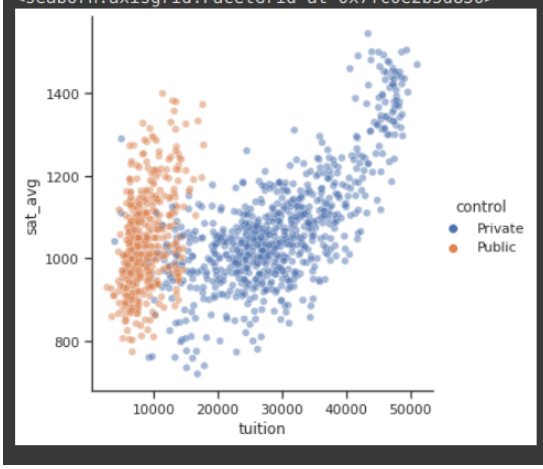
```
<seaborn.axisgrid.FacetGrid at 0x7fc0e2b99b50>
```



SEABORN: RELATIONAL PLOT (WITH HUE CONTROL)

```
sns.relplot(x='tuition', y='sat_avg', hue='control',alpha=0.5, data=college)
```

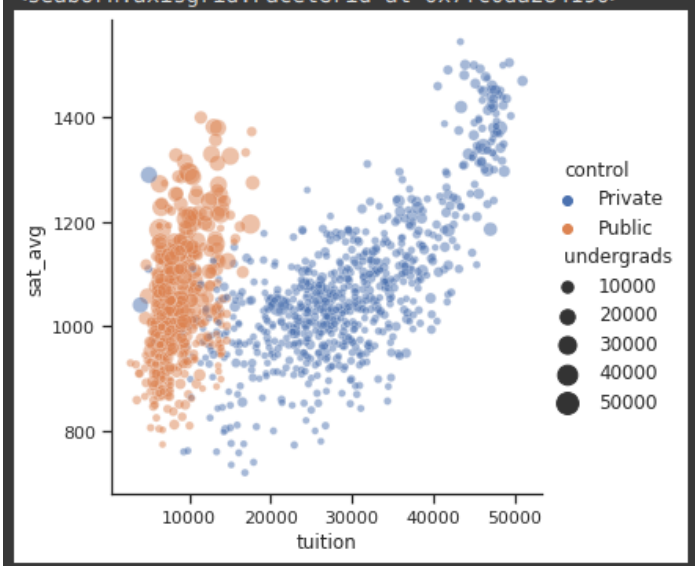
```
<seaborn.axisgrid.FacetGrid at 0x7fc0e2b5d850>
```



SEABORN: RELATIONAL PLOT WITH 4 VARIABLES

```
sns.relplot(x='tuition', y='sat_avg',
            hue='control',size='undergrads',sizes=(20,200),
            alpha=0.5, data=college)
```

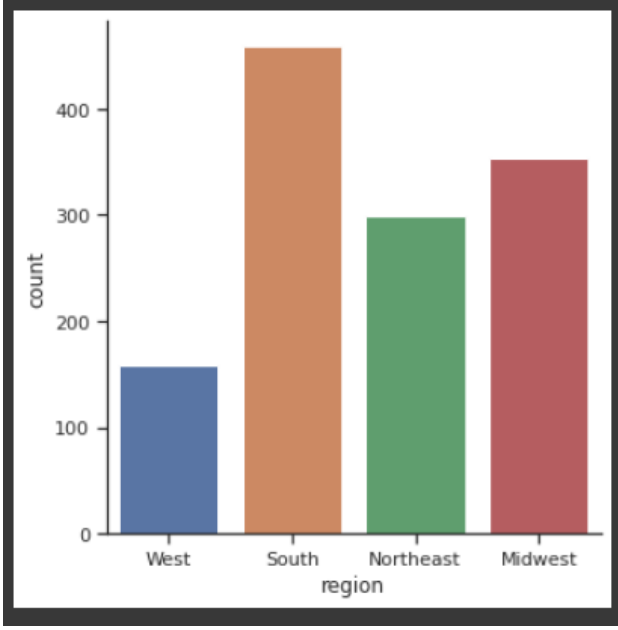
```
<seaborn.axisgrid.FacetGrid at 0x7fc0da284190>
```



STEP 4: BAR PLOT

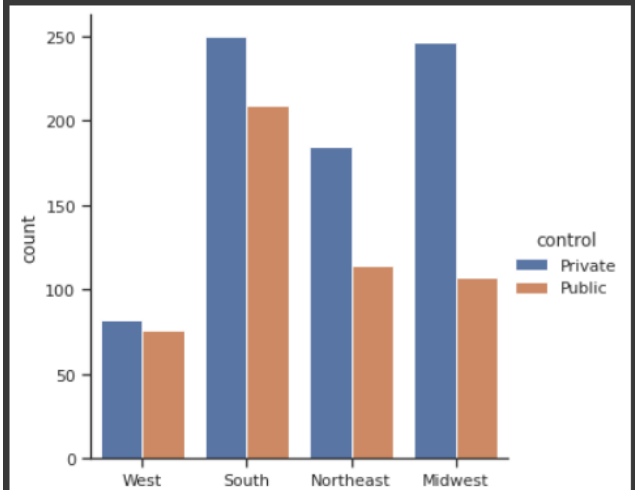
SEABORN: BAR PLOT (COUNT) WITH 2 VARIABLES

```
sns.catplot(x="region",kind='count',data=college);
```



SEABORN: BAR PLOT (COUNT) WITH 3 VARIABLES

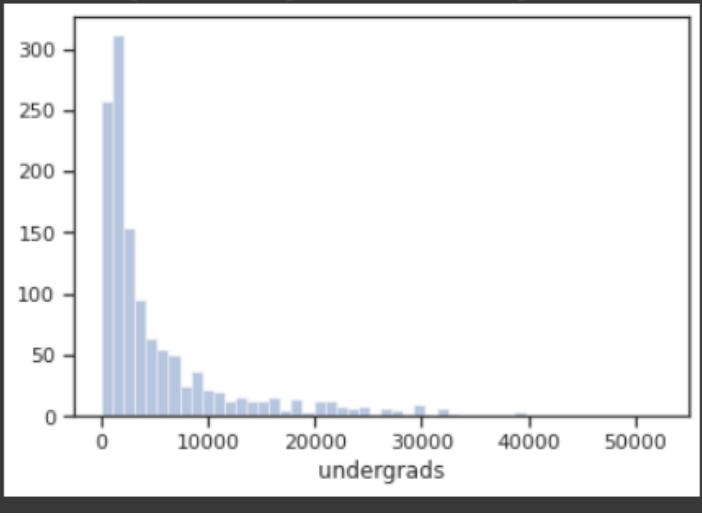
```
sns.catplot(x="region",kind='count', hue='control',data=college);
```



STEP 5: HISTOGRAM

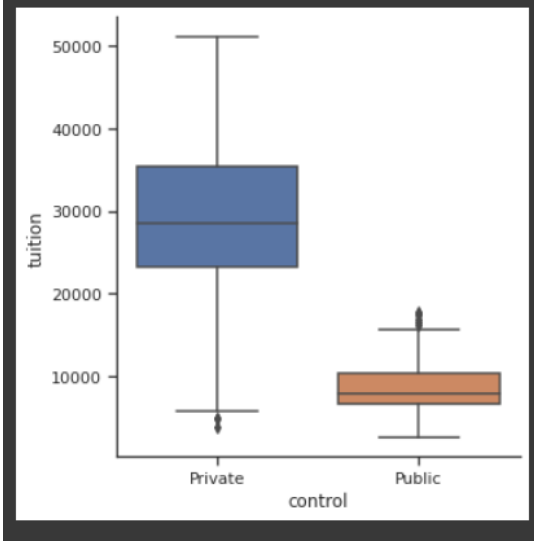
```
sns.distplot(college.undergrads,kde=False);
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py
  warnings.warn(msg, FutureWarning)
```



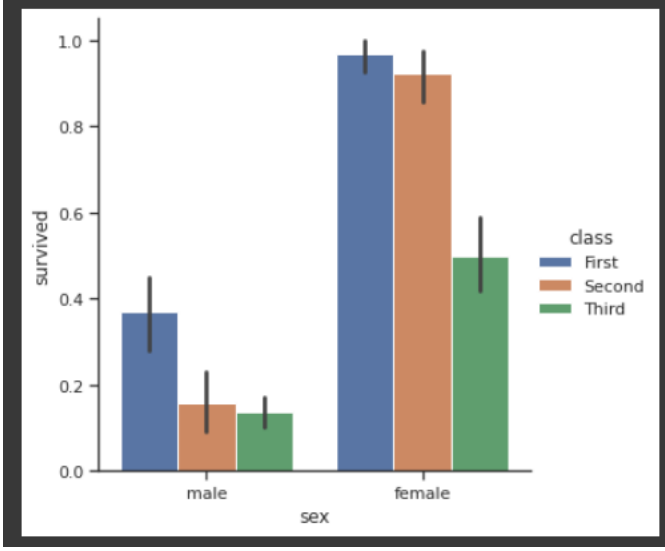
STEP 6: BOX PLOT

```
sns.catplot(data=college, x="control", y="tuition", kind="box");
```



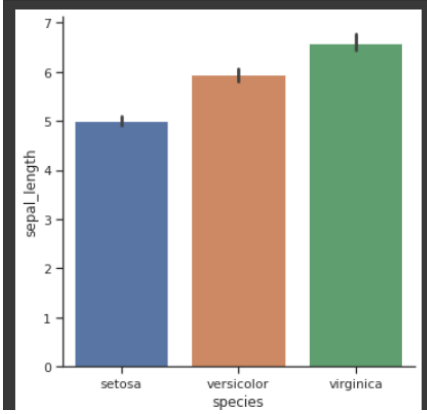
SEABORN DOUBLE BAR PLOT

```
titanic = sns.load_dataset("titanic")
sns.catplot(x="sex", y="survived", hue="class", kind="bar", data=titanic);
```



STEP 1: BAR PLOT

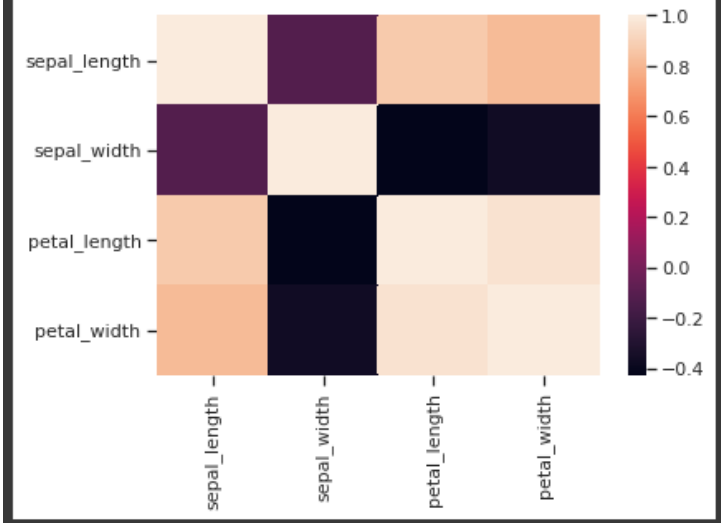
```
iris = sns.load_dataset("iris")
sns.catplot(data=iris, x="species", y="sepal_length", kind="bar");
```



STEP 2: HEATMAP

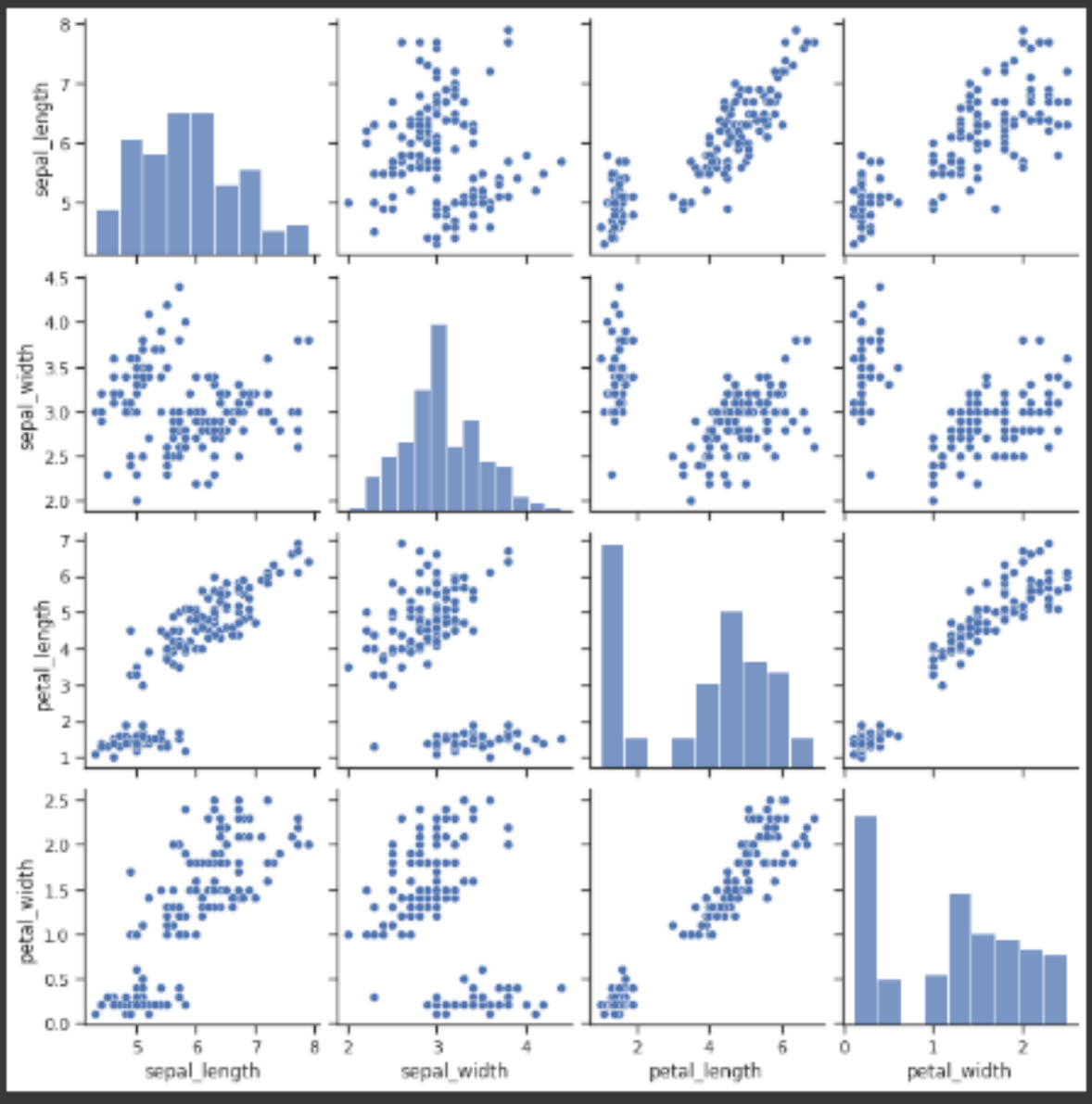
```
corr = iris.corr()
sns.heatmap(corr)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc0d61d82
```



STEP 3: PAIR PLOT

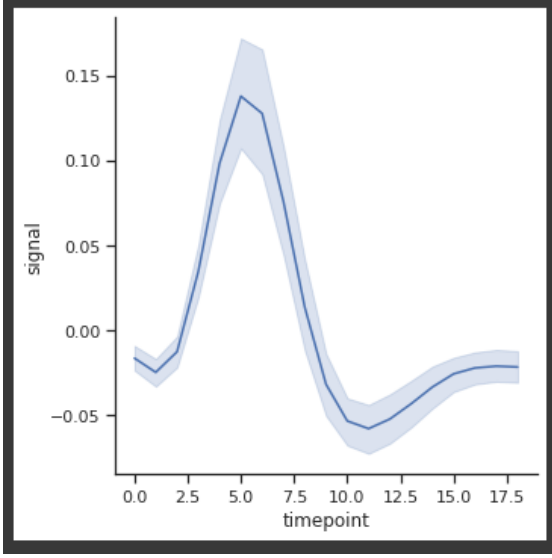
```
iris = sns.load_dataset("iris")
sns.pairplot(iris);
```



FMRI.CSV

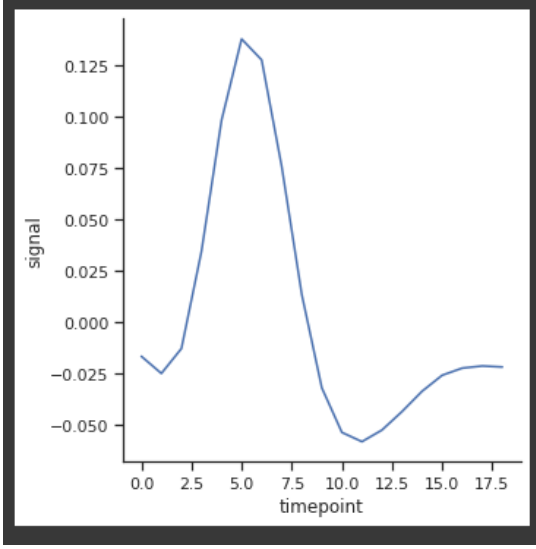
LINE PLOT WITH CI

```
fmri = sns.load_dataset("fmri")
sns.relplot(x="timepoint", y="signal", kind="line", data=fmri);
```



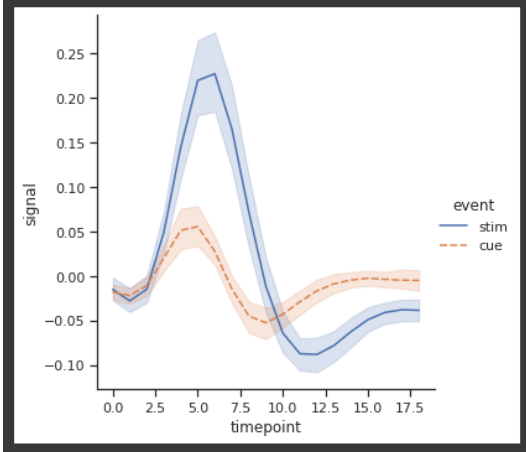
LINE PLOT WITHOUT CI

```
sns.relplot(x="timepoint", y="signal", ci=None, kind="line", data=fmri);
```



LINE PLOT WITH 3 VARIABLES

```
sns.relplot(x="timepoint", y="signal", hue="event", style="event",kind="line", data=fmri);
```



ABOUT DR. ALVIN ANG

Dr. Alvin Ang earned his Ph.D., Masters and Bachelor degrees from NTU, Singapore. He is a scientist, entrepreneur, as well as a personal/business advisor. More about him at www.AlvinAng.sg.