Seaborn visualization_ Youtube学习

Topics to be covered:

1. Distribution Plots
    - distplot
    - jointplot
    - pairplot
2. Categorical plots
    - barplot
    - countplot
    - boxplot
3. Heatmaps
4. Facet grids
5. Regression plots

Imports

In [2]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
# Seaborn comes with built-in data sets!
sns.get_dataset_names()
```

Out[3]:
```
['anagrams',
 'anscombe',
 'attention',
 'brain_networks',
 'car_crashes',
 'diamonds',
 'dots',
 'exercise',
 'flights',
 'fmri',
 'gammas',
 'geyser',
 'iris',
 'mpg',
 'penguins',
 'planets',
 'taxis',
 'tips',
 'titanic']
```

In [4]:
```
df = sns.load_dataset('mpg')
```

In [5]:
```
df.head()
```

Out[5]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	name
0	18.0	8	307.0	130.0	3504	12.0	70	usa	chevrolet chevelle malibu
1	15.0	8	350.0	165.0	3693	11.5	70	usa	buick skylark 320
2	18.0	8	318.0	150.0	3436	11.0	70	usa	plymouth satellite
3	16.0	8	304.0	150.0	3433	12.0	70	usa	amc rebel sst
4	17.0	8	302.0	140.0	3449	10.5	70	usa	ford torino

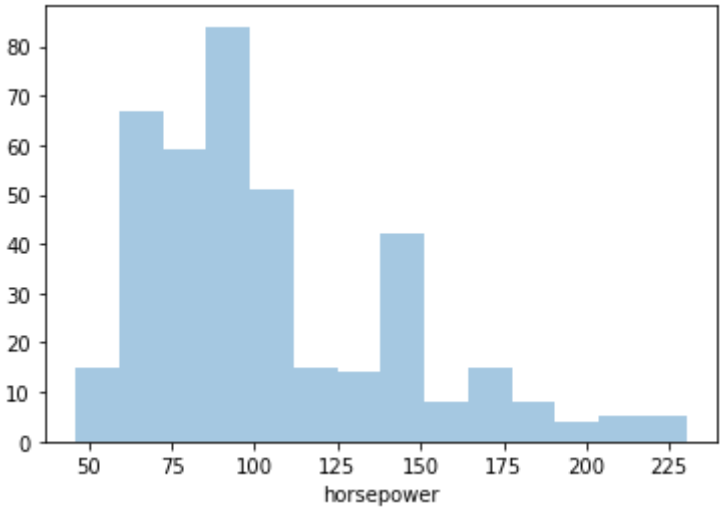
In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   mpg           398 non-null    float64
 1   cylinders     398 non-null    int64  
 2   displacement  398 non-null    float64
 3   horsepower    392 non-null    float64
 4   weight        398 non-null    int64  
 5   acceleration  398 non-null    float64
 6   model_year    398 non-null    int64  
 7   origin        398 non-null    object 
 8   name          398 non-null    object 
dtypes: float64(4), int64(3), object(2)
memory usage: 28.1+ KB
```

1. distribution plots

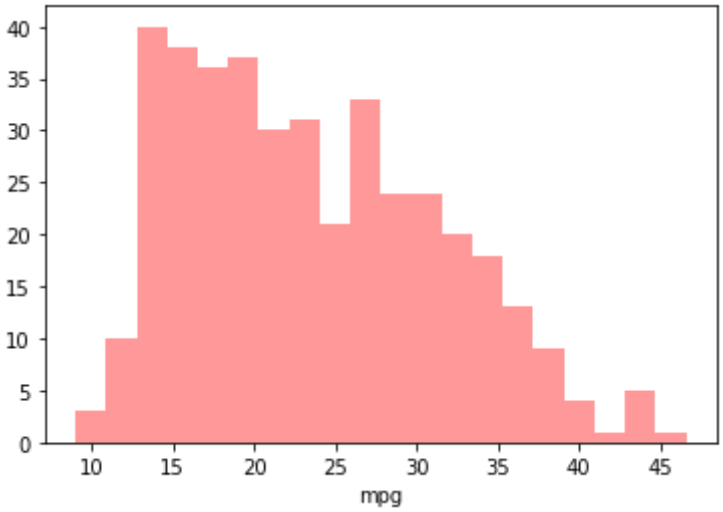
In [13]:
```
sns.distplot(df['horsepower'],kde=False)
plt.show()
# Kde Kernel Density Estimation (KDE) is a way to estimate
# the probability density function of a continuous random variable.
```

```
/Users/mac/opt/anaconda3/lib/python3.9/site-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



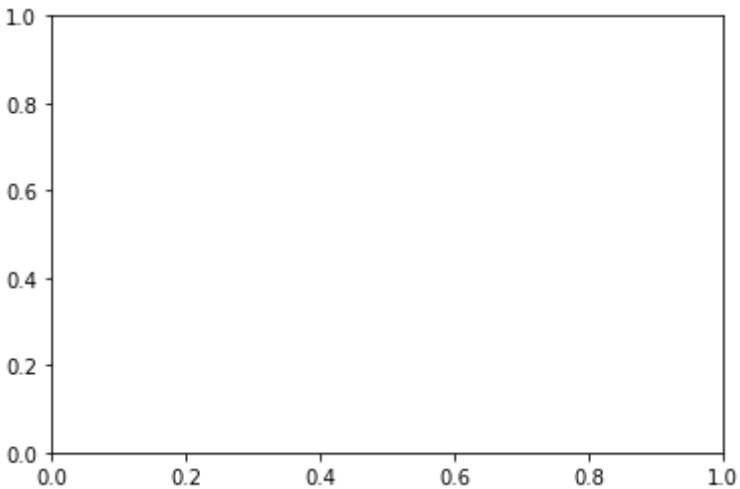
In [10]:
```
sns.distplot(df['mpg'],kde=False,bins=20, color="red")
```

Out[10]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1fbcd4e6b70>
```



In [17]:
```
sns.scatterplot()
```

Out[17]:
```
<AxesSubplot:>
```



In [18]:
```
!pip install tqdm
```

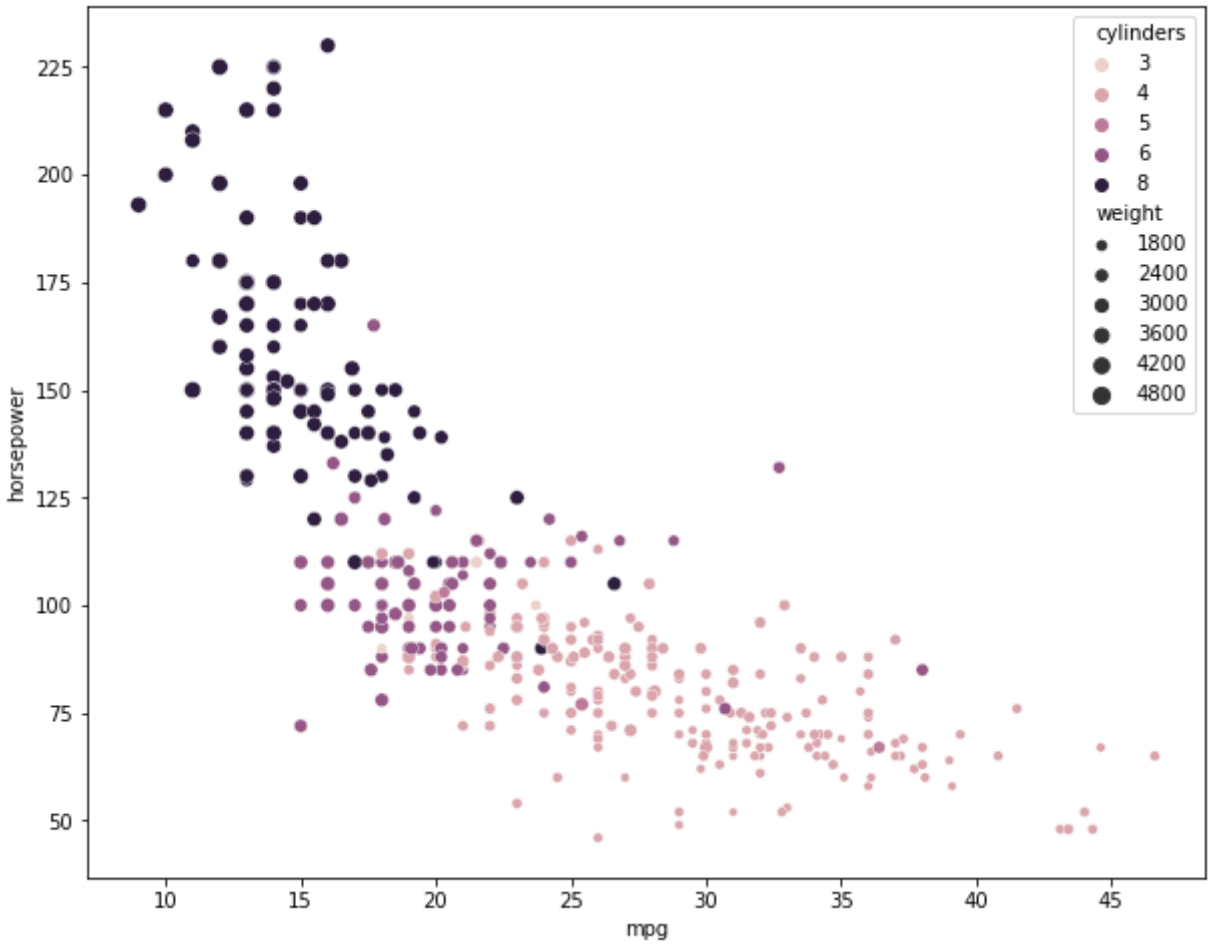
```
Requirement already satisfied: tqdm in /Users/mac/opt/anaconda3/lib/pytho
n3.9/site-packages (4.62.3)
```

In [19]:
```
df.head()
```

Out[19]:

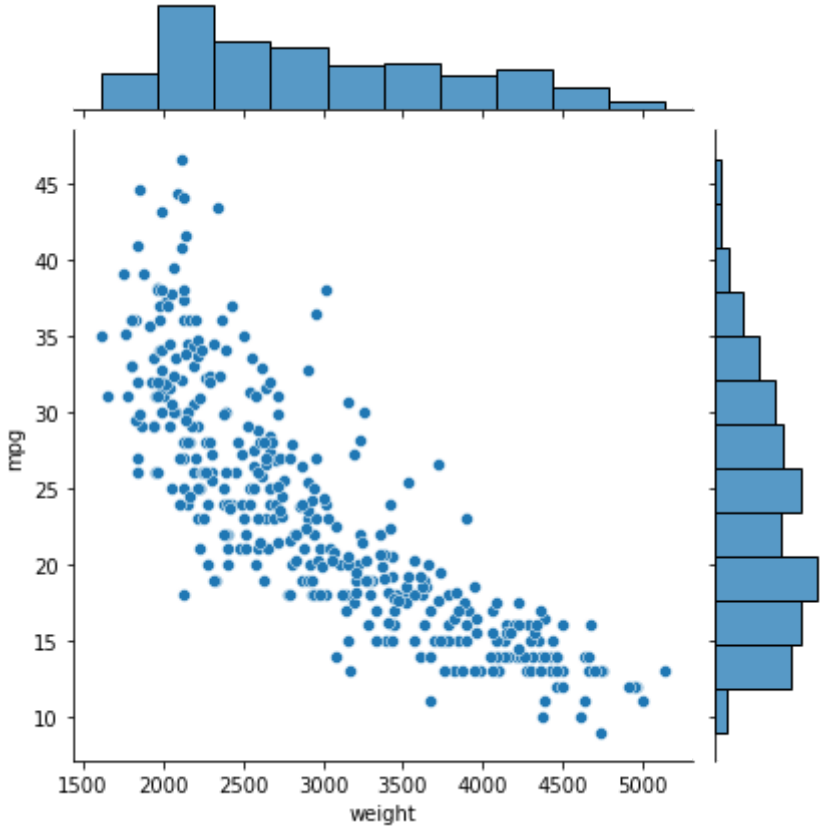
	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	name
0	18.0	8	307.0	130.0	3504	12.0	70	usa	chevrolet chevelle malibu
1	15.0	8	350.0	165.0	3693	11.5	70	usa	buick skylark 320
2	18.0	8	318.0	150.0	3436	11.0	70	usa	plymouth satellite
3	16.0	8	304.0	150.0	3433	12.0	70	usa	amc rebel sst
4	17.0	8	302.0	140.0	3449	10.5	70	usa	ford torino

In [20]:
```
plt.figure(figsize=(10,8))
sns.scatterplot(x='mpg', y='horsepower', data=df, hue='cylinders',size='wei
plt.show()
```



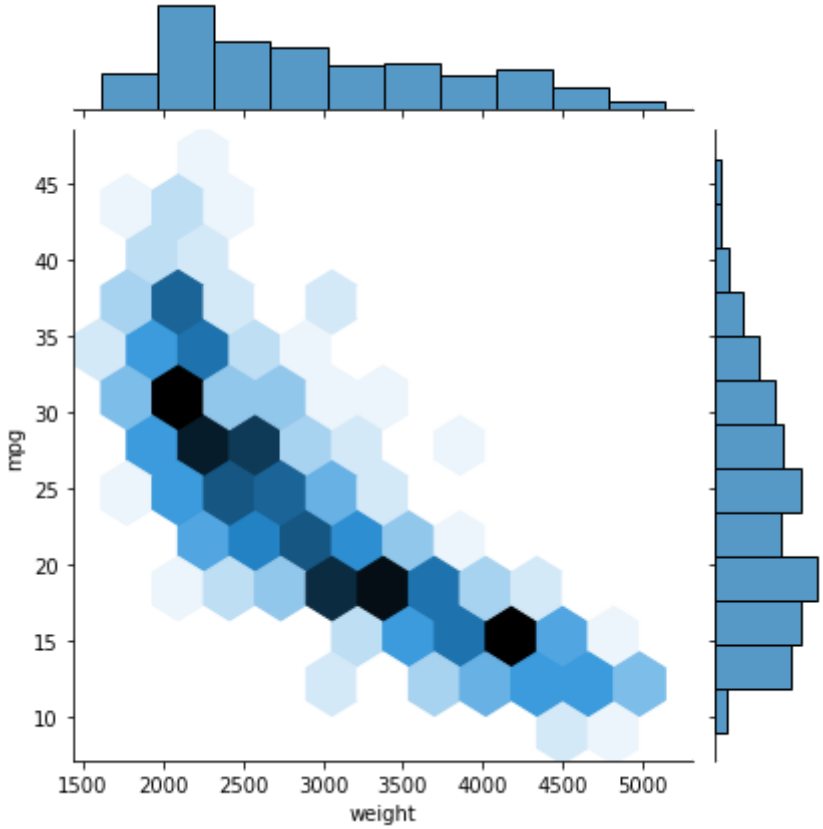
In [16]:
```
sns.jointplot(x='weight',y='mpg',data=df)
```

Out[16]: <seaborn.axisgrid.JointGrid at 0x7fe19e0b5580>



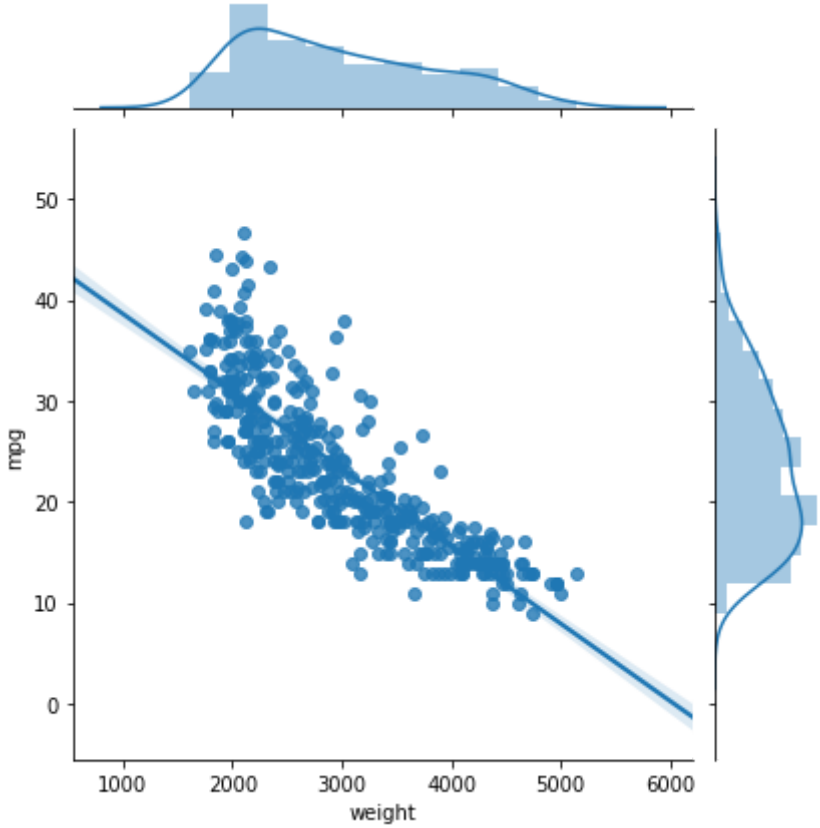
In [22]:
```
sns.jointplot(x='weight',y='mpg',data=df,kind='hex')
```

Out[22]:
```
<seaborn.axisgrid.JointGrid at 0x7fe19d88c070>
```



In [15]:
```
sns.jointplot(x='weight',y='mpg',data=df ,kind='reg')
```

Out[15]:
```
<seaborn.axisgrid.JointGrid at 0x1fbcd637780>
```



pairplot

pairplot will plot pairwise relationships across an entire dataframe (for the numerical columns) and supports a color hue argument (for categorical columns).

In [23]:
```
df.head(5)
```

Out[23]:

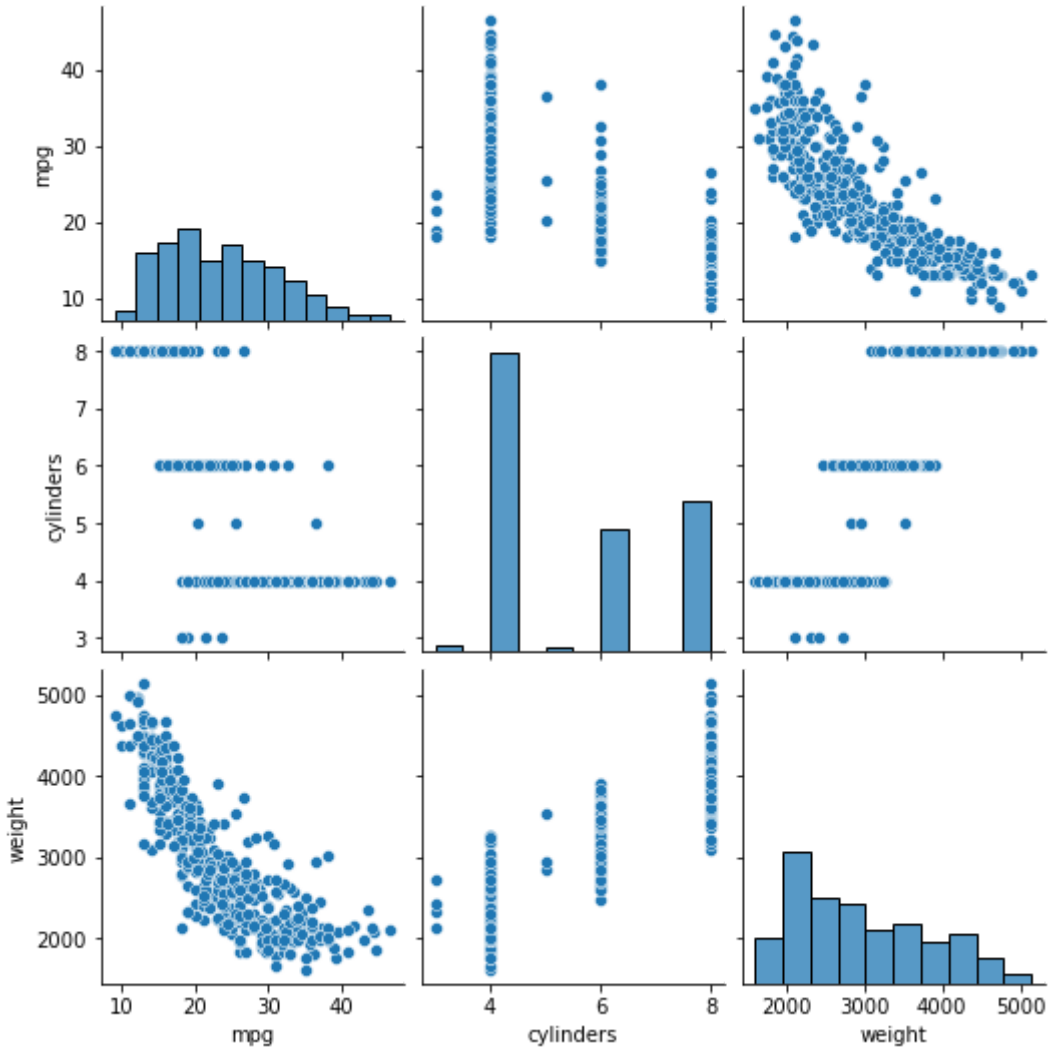
	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	name
0	18.0	8	307.0	130.0	3504	12.0	70	usa	chevrolet chevelle malibu
1	15.0	8	350.0	165.0	3693	11.5	70	usa	buick skylark 320
2	18.0	8	318.0	150.0	3436	11.0	70	usa	plymouth satellite
3	16.0	8	304.0	150.0	3433	12.0	70	usa	amc rebel sst
4	17.0	8	302.0	140.0	3449	10.5	70	usa	ford torino

In [24]:
```
df1 = df[['mpg','cylinders','weight']]
df1.head(5)
```

Out[24]:

	mpg	cylinders	weight
0	18.0	8	3504
1	15.0	8	3693
2	18.0	8	3436
3	16.0	8	3433
4	17.0	8	3449

In [25]:
```
sns.pairplot(df1) # pairplot look at the joint relationship between numeric
plt.show()
```

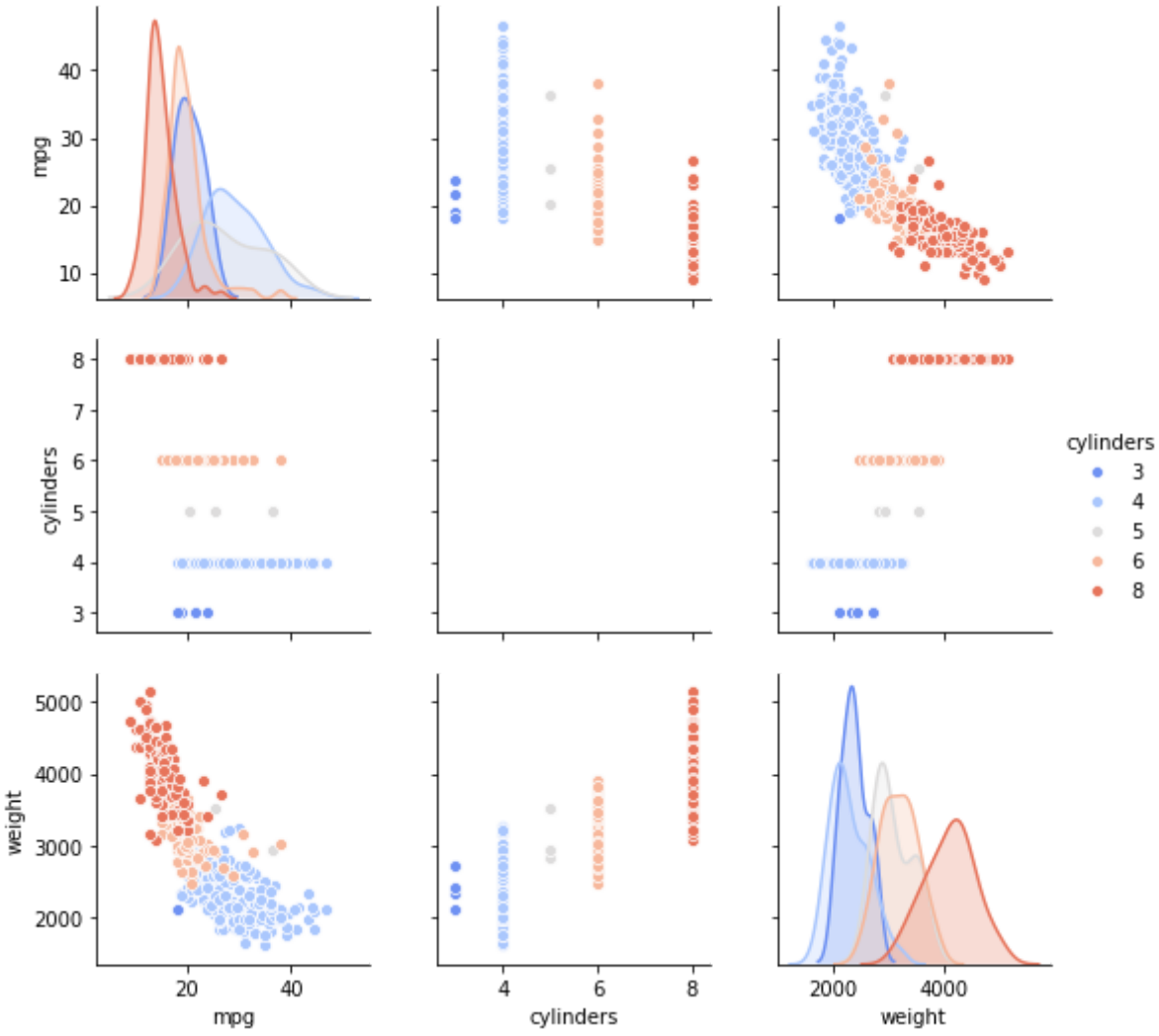


In [21]:
```
df1['cylinders'].value_counts()
```

Out[21]:
```
4    204
8    103
6     84
3      4
5      3
Name: cylinders, dtype: int64
```

In [23]:
```
sns.pairplot(df1,hue='cylinders',palette='coolwarm')
# hue is used for categorical variables
# palette specifies the colors
```

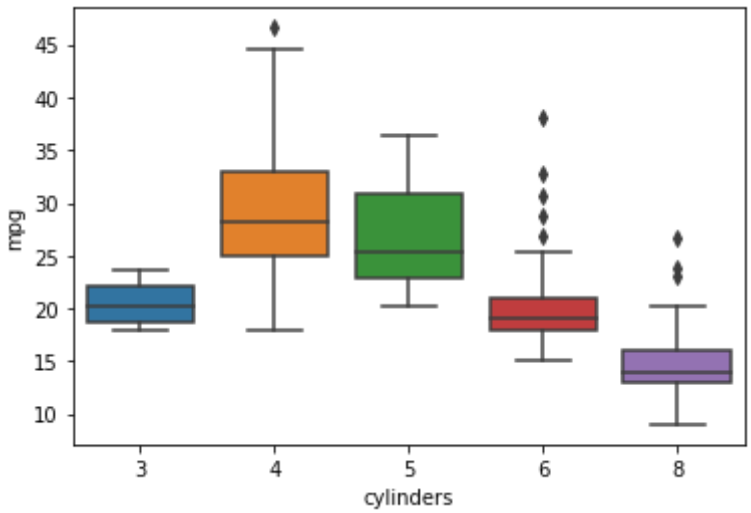
Out[23]:
```
<seaborn.axisgrid.PairGrid at 0x1fbcfe01b38>
```



2. Categorical plots

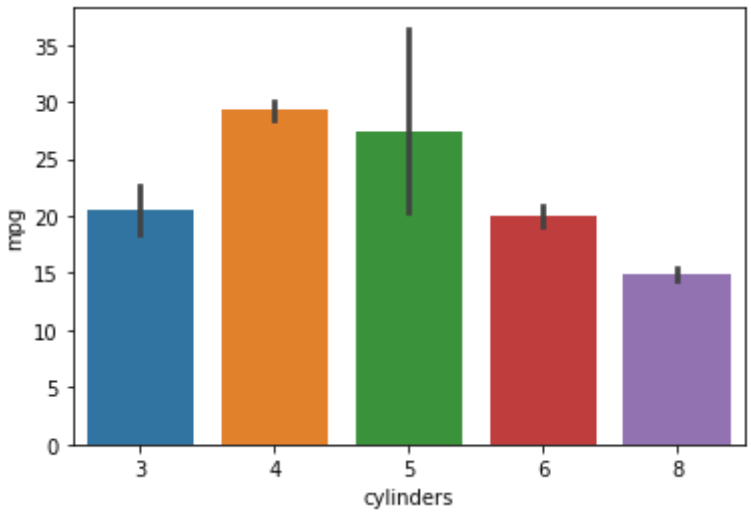
In [24]:
```
sns.boxplot(x='cylinders', y='mpg', data=df)
```

Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0x1fbd13e80b8>



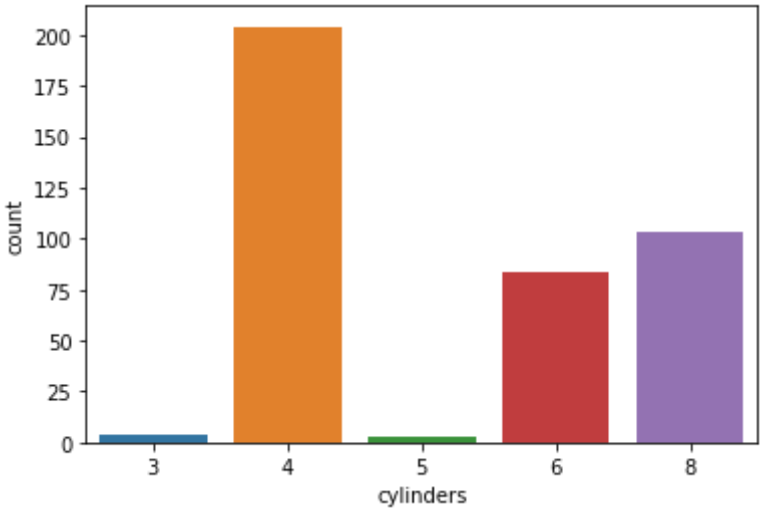
In [25]:
```
sns.barplot(x='cylinders', y='mpg', data=df)
# barplot is a general plot that allows you to aggregate the categorical da

# the black vertical lines are the error bars.
```

Out[25]: <matplotlib.axes._subplots.AxesSubplot at 0x1fbd148c978>



In [26]:
```
sns.countplot(x='cylinders',data=df)
```

Out[26]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1fbd1503dd8>
```



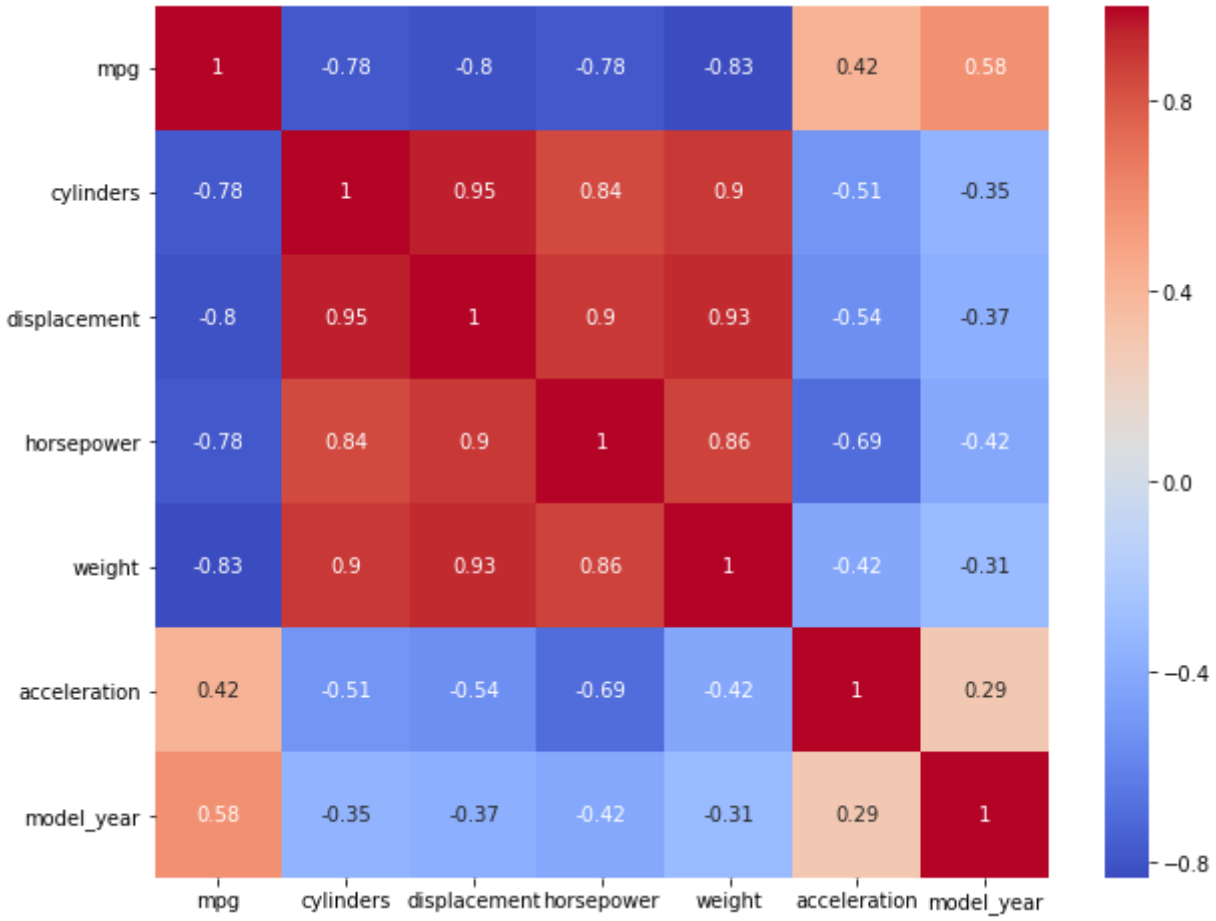
3. Heatmap

In [27]:
```
df.corr()
```

Out[27]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year
mpg	1.000000	-0.775396	-0.804203	-0.778427	-0.831741	0.420289	0.579267
cylinders	-0.775396	1.000000	0.950721	0.842983	0.896017	-0.505419	-0.348746
displacement	-0.804203	0.950721	1.000000	0.897257	0.932824	-0.543684	-0.370164
horsepower	-0.778427	0.842983	0.897257	1.000000	0.864538	-0.689196	-0.416361
weight	-0.831741	0.896017	0.932824	0.864538	1.000000	-0.417457	-0.306564
acceleration	0.420289	-0.505419	-0.543684	-0.689196	-0.417457	1.000000	0.288137
model_year	0.579267	-0.348746	-0.370164	-0.416361	-0.306564	0.288137	1.000000

In [28]:
```
plt.figure(figsize=(10,8))
sns.heatmap(df.corr(), cmap='coolwarm',annot=True)
plt.show()
```



4. Facet grids

In [ ]:
```
```

In [29]:
```
df.head()
```

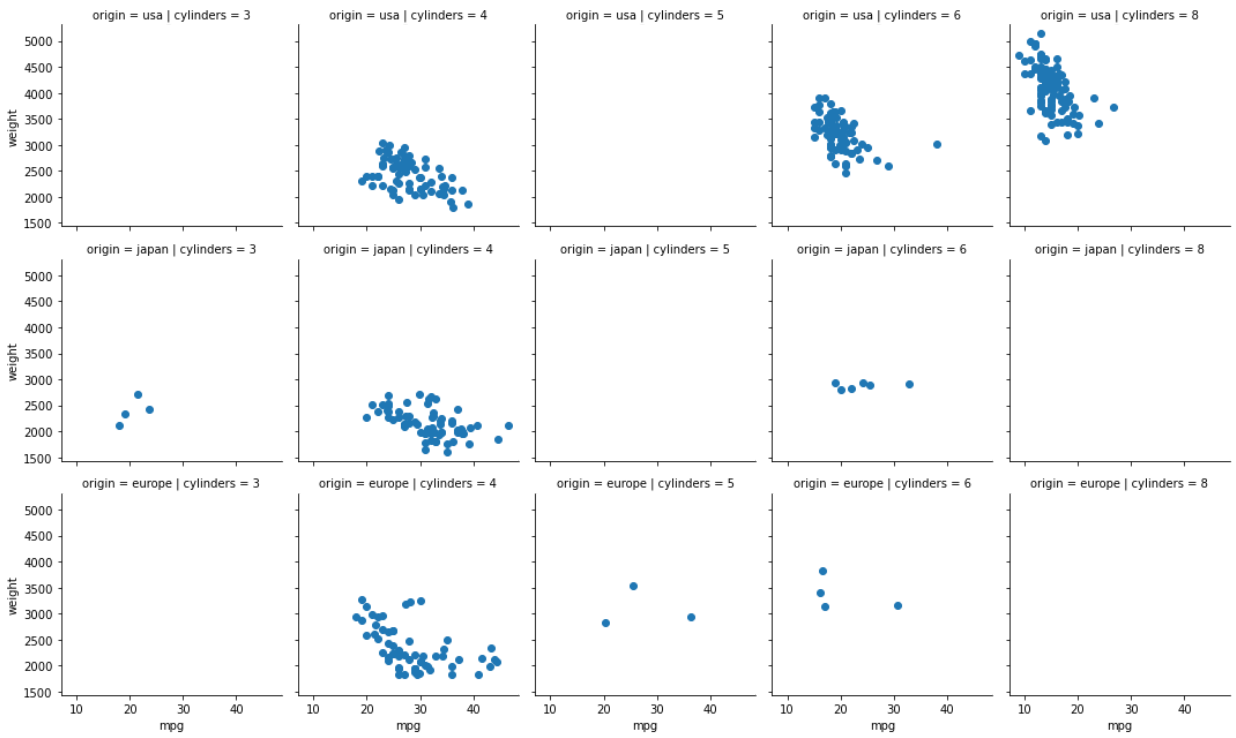
Out[29]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	name
0	18.0	8	307.0	130.0	3504	12.0	70	usa	chevrolet chevelle malibu
1	15.0	8	350.0	165.0	3693	11.5	70	usa	buick skylark 320
2	18.0	8	318.0	150.0	3436	11.0	70	usa	plymouth satellite
3	16.0	8	304.0	150.0	3433	12.0	70	usa	amc rebel sst
4	17.0	8	302.0	140.0	3449	10.5	70	usa	ford torino

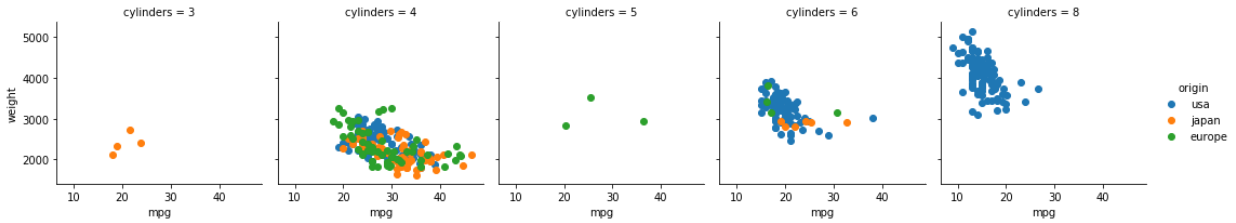
In [30]:
```
df['origin'].value_counts()
```

Out[30]:
```
usa       249
japan      79
europe     70
Name: origin, dtype: int64
```

In [30]:
```
g = sns.FacetGrid(df, col="cylinders",row='origin') # object
g = g.map(plt.scatter, "mpg", "weight") # Apply a plotting function to each
```



In [32]:
```
g = sns.FacetGrid(df, col="cylinders", hue='origin')
g = g.map(plt.scatter, "mpg", "weight").add_legend()
```



5. Regression plots

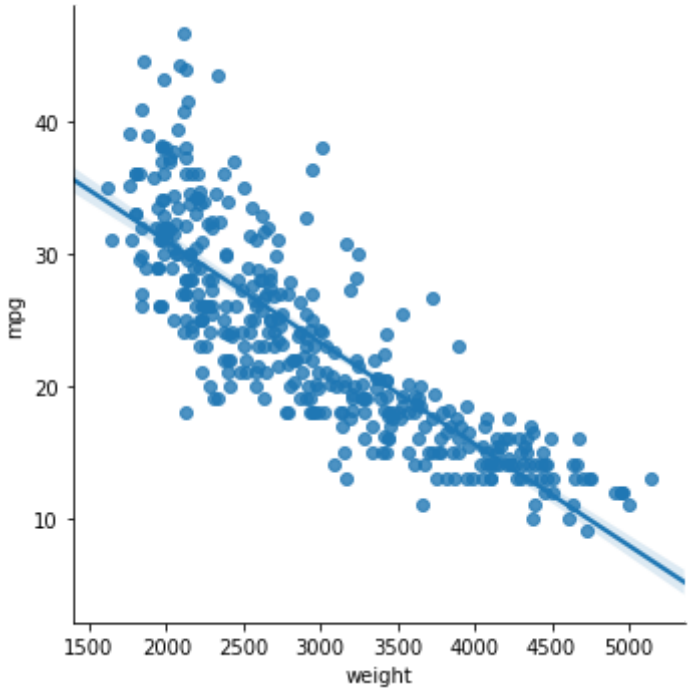
In [33]:
```
df.head()
```

Out[33]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	name
0	18.0	8	307.0	130.0	3504	12.0	70	usa	chevrolet chevelle malibu
1	15.0	8	350.0	165.0	3693	11.5	70	usa	buick skylark 320
2	18.0	8	318.0	150.0	3436	11.0	70	usa	plymouth satellite
3	16.0	8	304.0	150.0	3433	12.0	70	usa	amc rebel sst
4	17.0	8	302.0	140.0	3449	10.5	70	usa	ford torino

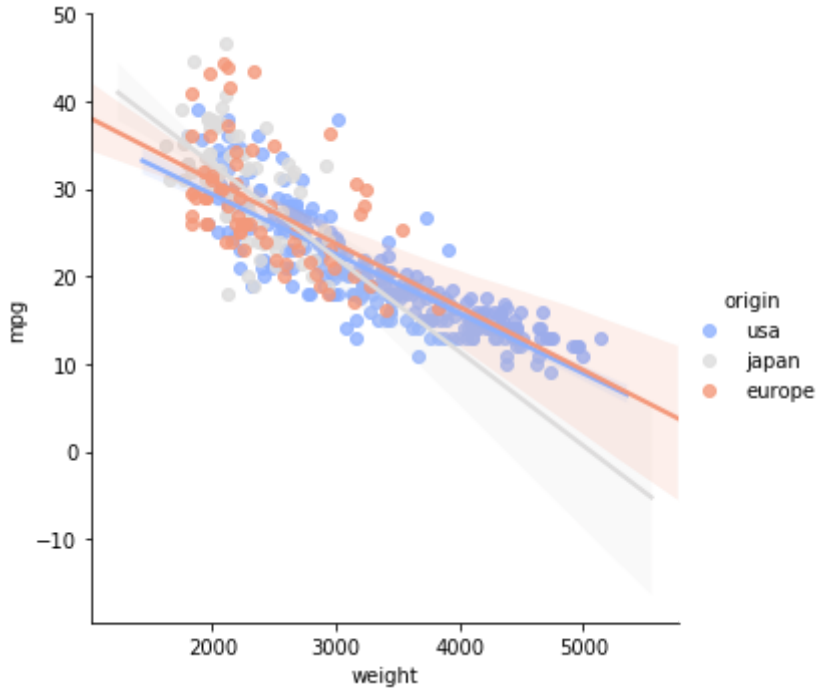
In [34]:
```
sns.lmplot(x='weight',y='mpg',data=df)
```

Out[34]:
```
<seaborn.axisgrid.FacetGrid at 0x1fbd36752b0>
```



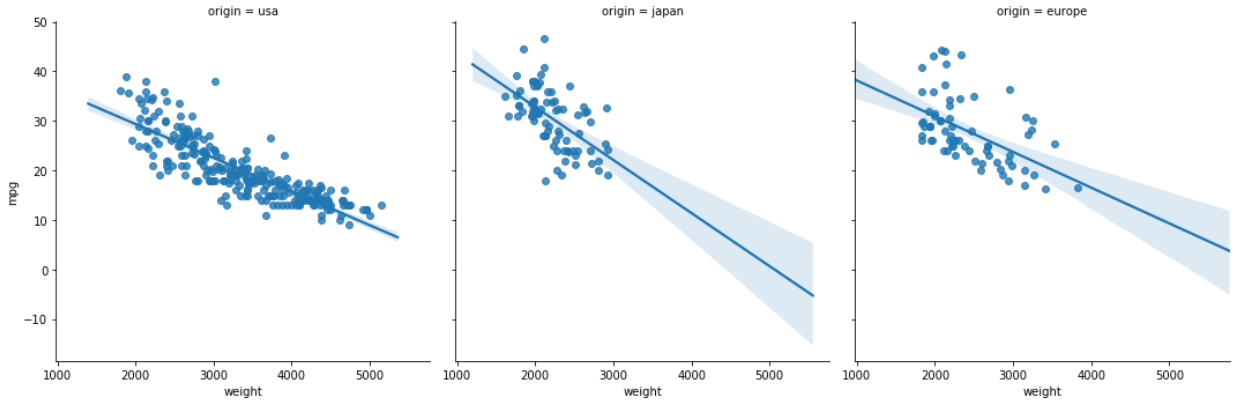
In [35]:
```
sns.lmplot(x='weight',y='mpg',data=df, hue='origin', palette='coolwarm')
```

Out[35]: <seaborn.axisgrid.FacetGrid at 0x1fbd36d6780>



In [36]:
```
sns.lmplot(x='weight',y='mpg',data=df, col='origin')
```

Out[36]: <seaborn.axisgrid.FacetGrid at 0x1fbd3759a20>



Check out the seaborn documentation website (https://seaborn.pydata.org/tutorial.html (https://seaborn.pydata.org/tutorial.html)) to explore more examples.