In [51]:
```
print("Author: Louis-Antoine Etchian")
```

```
Author: Louis-Antoine Etchian
```

In [52]:
```
%matplotlib inline

import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
```

Plotting with pandas

> We use the standard convention for referencing the matplotlib API ... We provide the basics in pandas to easily create decent looking plots.

https://pandas.pydata.org/pandas-docs/stable/user_guide/visualization.html (https://pandas.pydata.org/pandas-docs/stable/user_guide/visualization.html)

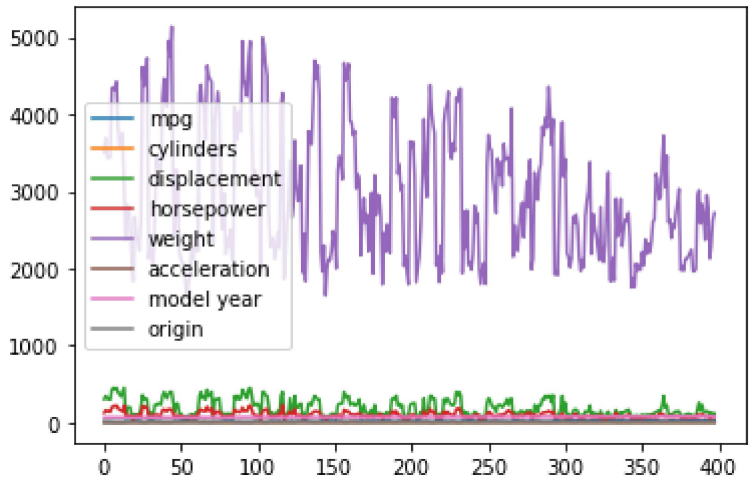
Let's load the heart attack dataset

In [55]:
```
data_names=np.array(["mpg", "cylinders", "displacement", "horsepower", "weight",
data=pd.read_csv('auto-mpg.data', delim_whitespace=True, header=None, names = dat
```

Plotting all columns, works, but does not provide a lot of insight.

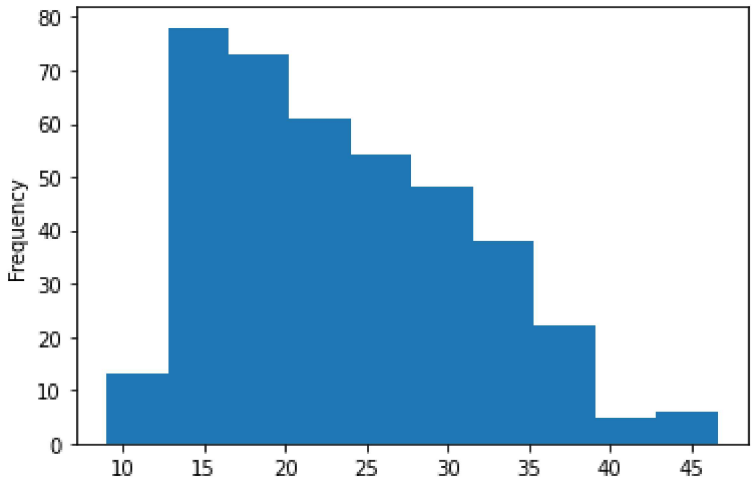
In [56]:
```
data.plot()
```

Out[56]:
```
<AxesSubplot:>
```



Let's look at the age distribution (a histogram)

In [57]:
```
data['mpg'].plot.hist();
```



How many male and female samples do we have?

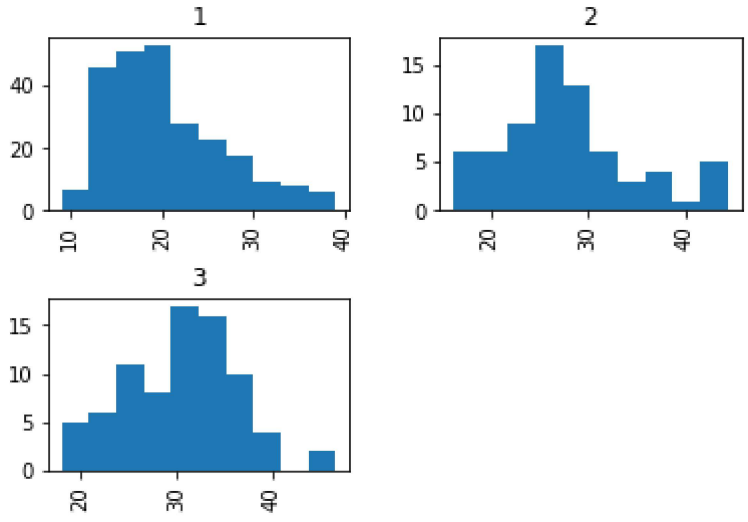
In [58]:
```
data.origin.value_counts()
```

Out[58]:
```
1    249
3     79
2     70
Name: origin, dtype: int64
```

Notice that we accessed the gender column with dot notation. This can be done whenever the column name is 'nice' enough to be a python variable name.

Do we have similar ages in females and males?
Plotting two histograms for each gender side beside directly form the dataframe:

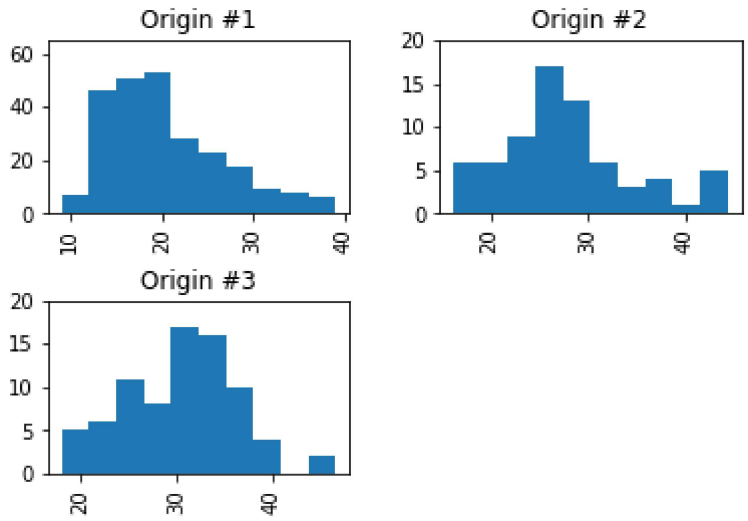
In [59]:
```
#Same focus than the previous documents (mpg by origin)
axs = data.hist(column='mpg', by='origin')
```



To format this plot, we can work on the axes (array) that is returned by the plot call. We use Matplotlib object oriented interface methods to do this

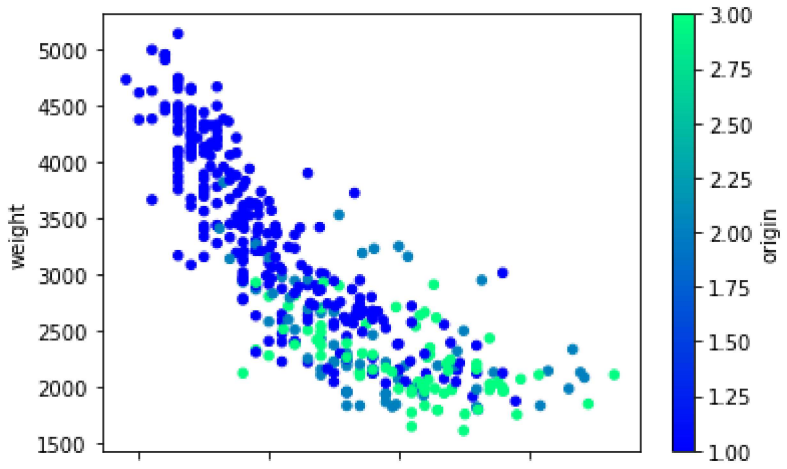
In [65]:
```
axs = data.hist(column='mpg', by='origin')
axs[0][0].set(title='Origin #1', ylim=[0, 65])
axs[0][1].set(title='Origin #2', ylim=[0, 20])
axs[1][0].set(title='Origin #3', ylim=[0, 20])
```

Out[65]:
```
[Text(0.5, 1.0, 'Origin #3'), (0.0, 20.0)]
```



Is age and blood pressure correlated? Maybe it is different for females and males? Let's have a look with a scatter plot.

In [66]:
```
data.plot.scatter('mpg', 'weight', c='origin', colormap='winter');
```

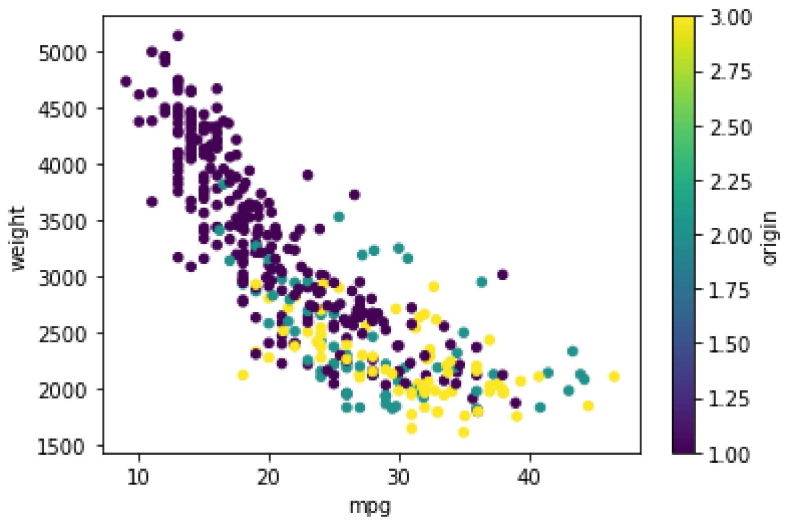


According to:
https://stackoverflow.com/questions/43578976/pandas-missing-x-tick-labels (https://stackoverflow.com/questions/43578976/pandas-missing-x-tick-labels)

the missing x-labels are a pandas bug.

Workaraound is to create axes prior to calling plot

In [67]:
```
fig, ax = plt.subplots()
data.plot.scatter('mpg', 'weight', c='origin', colormap='viridis', ax=ax);
```

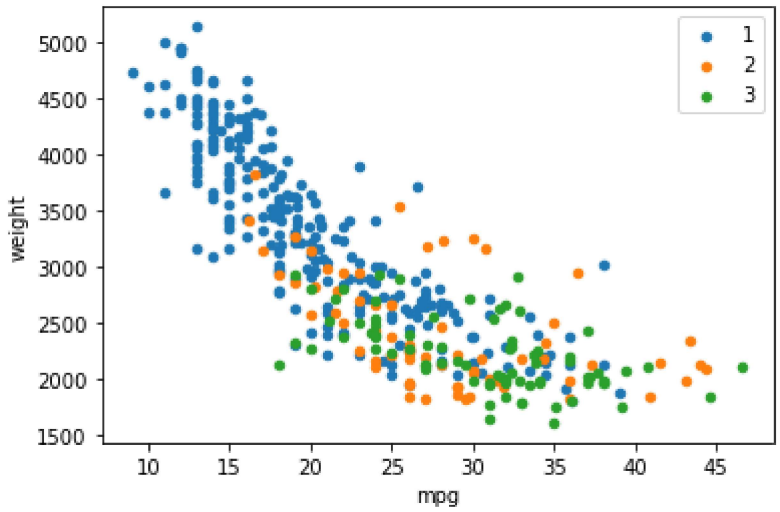


It is a bit annoying that there is a colorbar, we know gender is categorical.

One way to avoid the colorbar is to loop over the categories and assign colors based on the category.

See: [https://stackoverflow.com/questions/26139423/plot-different-color-for-different-categorical-levels-using-matplotlib (https://stackoverflow.com/questions/26139423/plot-different-color-for-different-categorical-levels-using-matplotlib)](https://stackoverflow.com/questions/26139423/plot-different-color-for-different-categorical-levels-using-matplotlib)

In [68]:
```
colors = {1: 'tab:blue', 2: 'tab:orange', 3: 'tab:green'}
fig, ax = plt.subplots()
for key, group in data.groupby(by='origin'):
    group.plot.scatter('mpg', 'weight', c=colors[key], label=key, ax=ax);
```



Seaborn

> Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

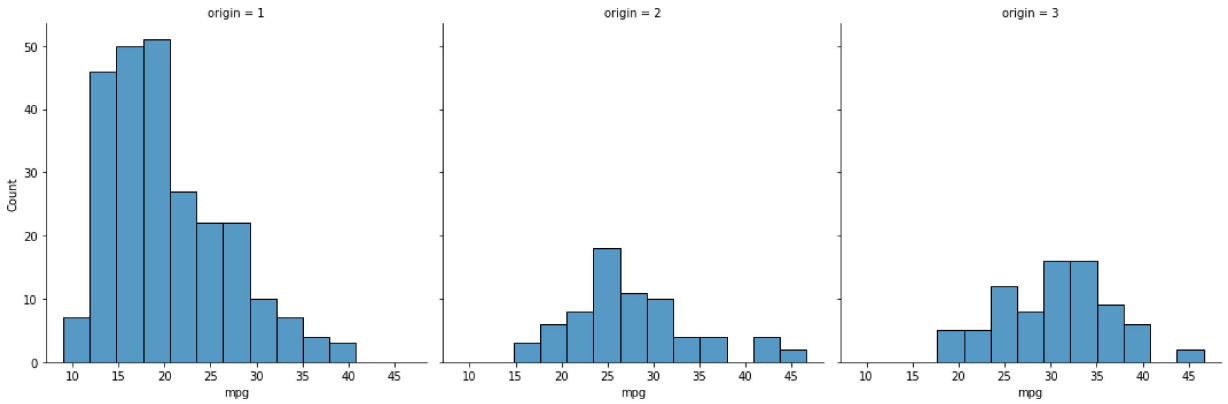
http://seaborn.pydata.org/index.html (http://seaborn.pydata.org/index.html)

Seaborn is usually imported as `sns`

In [69]:
```
import seaborn as sns
```

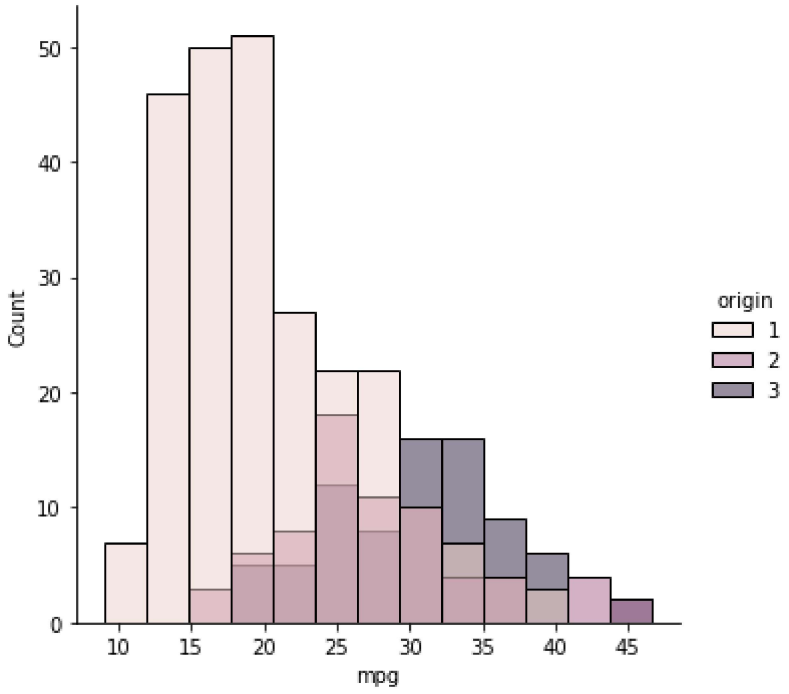
Let's re-create the histograms by gender with seaborn with the figure level `displot()` function.

In [70]:
```
# Use origin to split age into columns
sns.displot(x='mpg', col='origin', data=data);
```



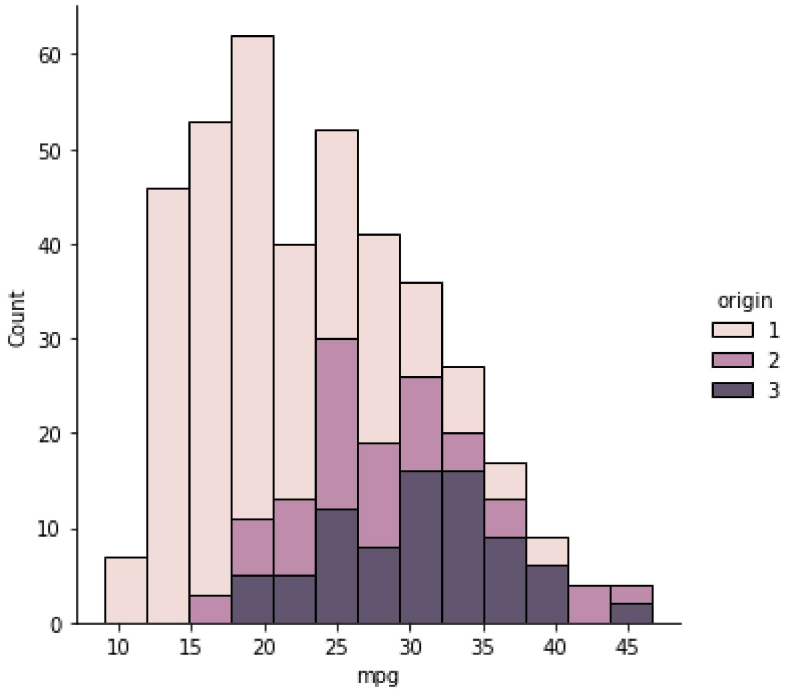
We can display the counts in the same plot, one on top of the other.

In [71]:
```
# Use origin to color (hue) in the same plot
sns.displot(x='mpg', hue='origin', data=data);
```



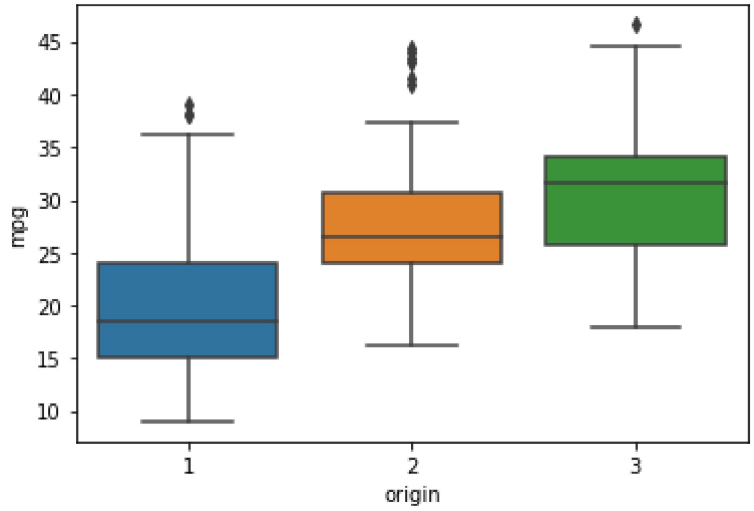
To have an idea of the split between male and female, we can stack the counts, adding up to total.

In [72]:
```
sns.displot(x='mpg', hue='origin', data=data, multiple='stack');
```



We can look at the differences in ages with a boxplot too

In [73]:
```
sns.boxplot(x='origin', y='mpg', data=data);
```

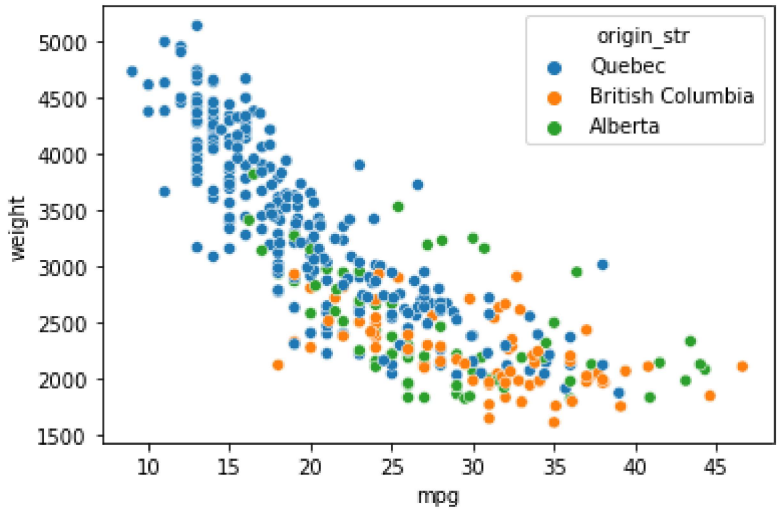


Let's re-create the scatter plot to see if age and blood pressure are correlated by gender.

To make the legend show strings we will create a gender string column with female and male strings rather than 0 and 1.

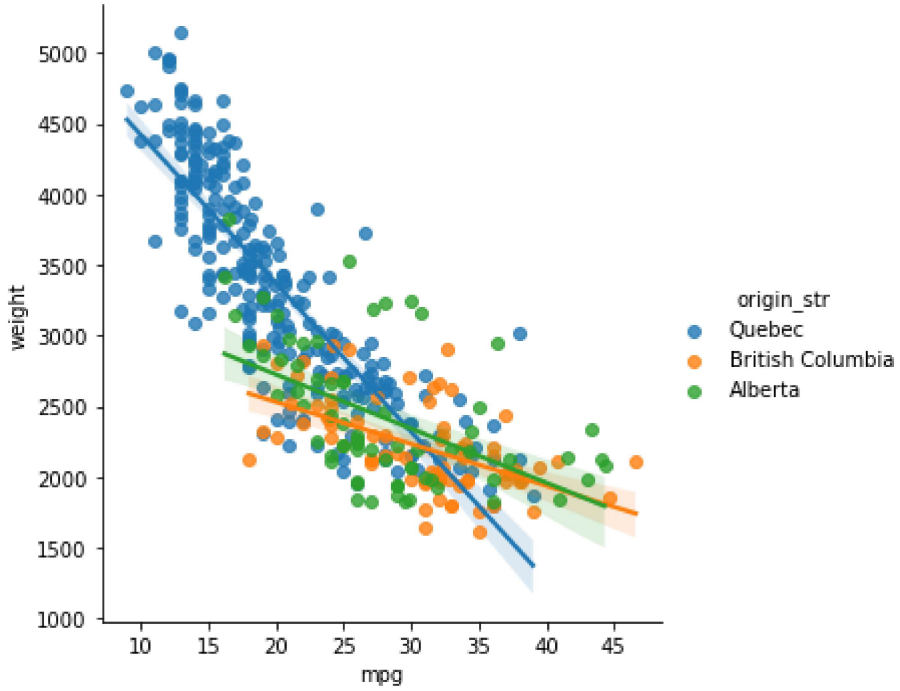
In [79]:
```
data['origin_str'] = data['origin'].replace([1,2,3], ['Quebec','Alberta','British
```

In [80]:
```
ax = sns.scatterplot(x='mpg', y='weight', data=data, hue='origin_str')
```



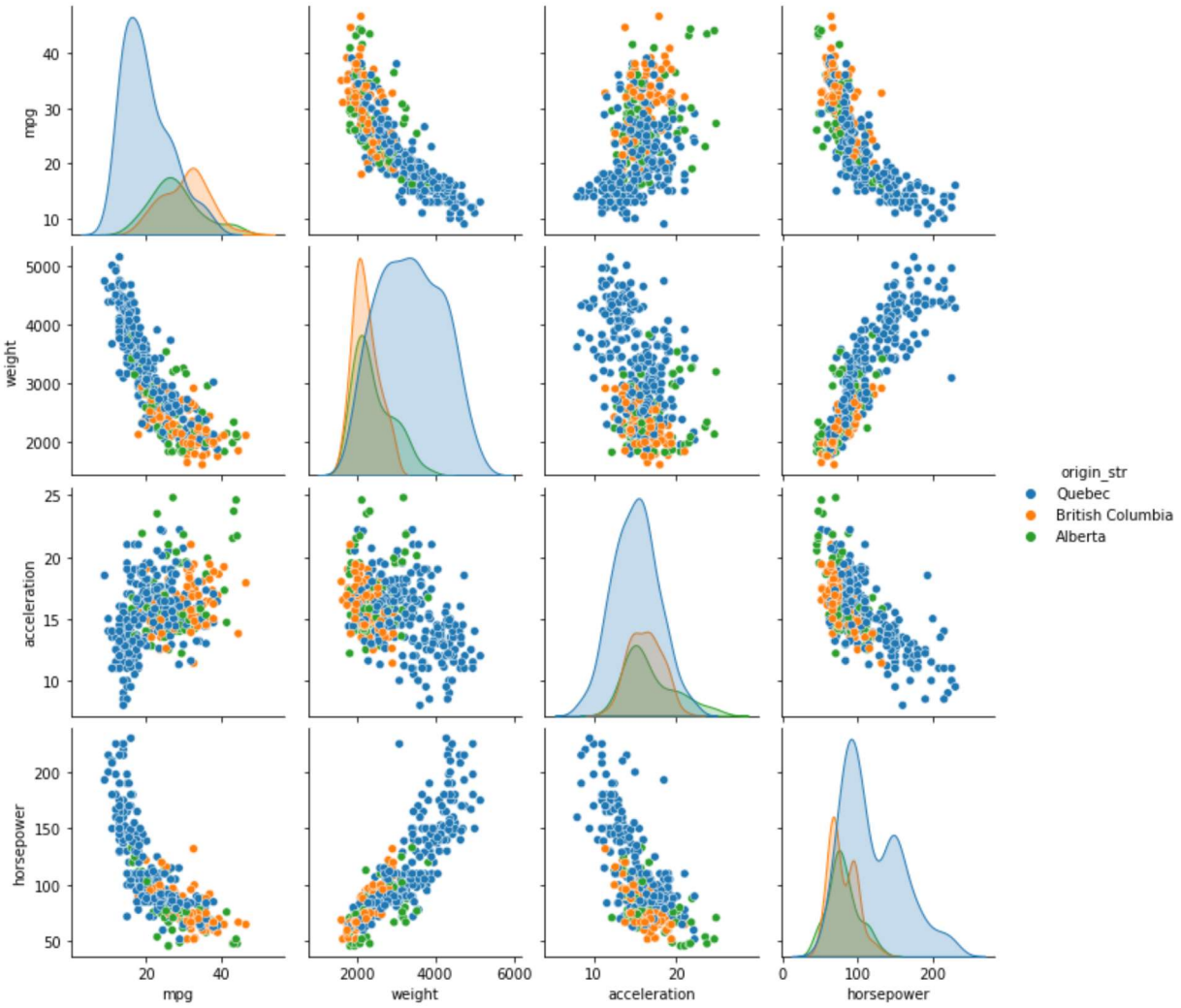
Adding a regression line helps with visualizing the relationship

In [81]:
```
ax = sns.lmplot(x='mpg', y='weight', data=data, hue='origin_str')
```



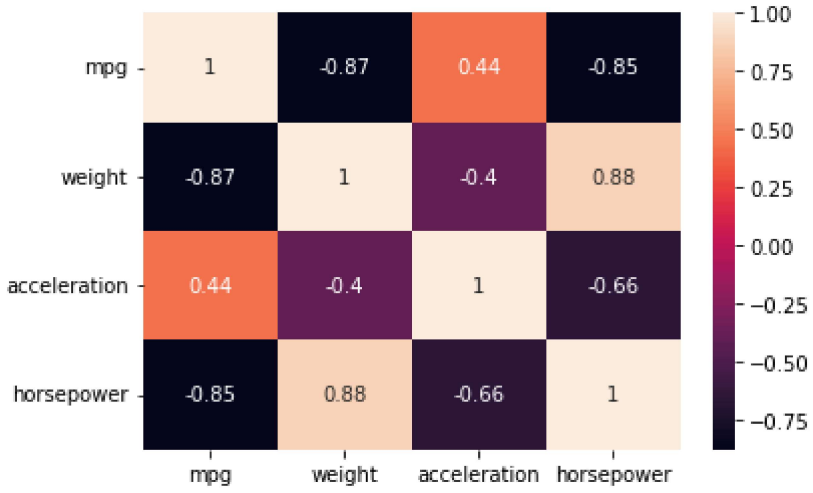
Maybe there are other correlations in the data set. Pairplot is a great way to get an overview

In [82]:
```
sns.pairplot(data, vars=['mpg', 'weight', 'acceleration', 'horsepower'], hue='ori
```



As an alternative, we can visualize the correlation matrix as a heatmap

In [83]:
```
g = sns.heatmap(data[['mpg', 'weight', 'acceleration', 'horsepower']].corr(metho
                annot=True)
```



There are nice tutorials on the Seaborn website, be sure to check these out.

In [ ]:
```
```