Data Visualisation In Data Analysis

It is very important for a data scientist to visualise a data as well as to clean it before modelling it .Here I have clean the data and then visualise it using different techniques in Python.Visualisation help in understanding how different paramteres are releated to each other

Different libraries are imported over here for visualisation .Pandas used for table formation while numpy is used for mathmatical operations

In [50]:

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

Now the data file is imported using pd.read_csv as the data was stored in Excel

In [51]:

```
cars=pd.read_csv('cars.csv')
```

Now lets see how data looks at first sight using head() function

In [53]:

```
cars.head()
```

Out[53]:

	Car;MPG;Cylinders;Displacement;Horsepower;Weight;Acceleration;Model;Origin
0	STRING;DOUBLE;INT;DOUBLE;DOUBLE;DOUBLE;DOUBLE;...
1	Chevrolet Chevelle Malibu;18.0;8;307.0;130.0;3...
2	Buick Skylark 320;15.0;8;350.0;165.0;3693.;11....
3	Plymouth Satellite;18.0;8;318.0;150.0;3436.;11...
4	AMC Rebel SST;16.0;8;304.0;150.0;3433.;12.0;70;US

As it can be seen all columns are merged into one column .So we need to separate these columns.This can be done using split function().

In [54]:

```
a='Car;MPG;Cylinders;Displacement;Horsepower;Weight;Acceleration;Model;Origin'
b=a.split(';')
for i in range(0,9):
    cars[b[i]]=cars[a].str.split(';').str[i]
del cars[a]
```

Now we can see the columns are separated and data looks perfect

In [56]:

```
cars.head()
```

Out[56]:

	Car	MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model
0	STRING	DOUBLE	INT	DOUBLE	DOUBLE	DOUBLE	DOUBLE	INT
1	Chevrolet Chevelle Malibu	18.0	8	307.0	130.0	3504.	12.0	70
2	Buick Skylark 320	15.0	8	350.0	165.0	3693.	11.5	70
3	Plymouth Satellite	18.0	8	318.0	150.0	3436.	11.0	70
4	AMC Rebel SST	16.0	8	304.0	150.0	3433.	12.0	70

The 0th row is of no use so we can eliminate that row as shown below

In [57]:

```
cars=cars[1:]
```

In [58]:

```
cars.head()
```

Out[58]:

	Car	MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model	Origi
1	Chevrolet Chevelle Malibu	18.0	8	307.0	130.0	3504.	12.0	70	U
2	Buick Skylark 320	15.0	8	350.0	165.0	3693.	11.5	70	U
3	Plymouth Satellite	18.0	8	318.0	150.0	3436.	11.0	70	U
4	AMC Rebel SST	16.0	8	304.0	150.0	3433.	12.0	70	U
5	Ford Torino	17.0	8	302.0	140.0	3449.	10.5	70	U

The info() function is use for checking out numder of rows and columns .Also to see what type of data is there.As one can see all numeric data is present as object which is basically string , so we need to convert it into numeric data type

In [59]:

```
cars.info() # all numeric data is in string , so need to convert into int or float tyyp
e
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 406 entries, 1 to 406
Data columns (total 9 columns):
Car             406 non-null object
MPG             406 non-null object
Cylinders       406 non-null object
Displacement    406 non-null object
Horsepower      406 non-null object
Weight          406 non-null object
Acceleration    406 non-null object
Model           406 non-null object
Origin          406 non-null object
dtypes: object(9)
memory usage: 28.7+ KB
```

I am selecting column 1 to 7 as these are numeric data which are represented as string type.The data is converted using astype() function .The function converts string into float format

In [60]:

```
d=cars.columns[1:7]
for i in range(0,6):
    cars[d[i]]=cars[d[i]].astype(float)
```

In [61]:

```
cars.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 406 entries, 1 to 406
Data columns (total 9 columns):
Car             406 non-null object
MPG             406 non-null float64
Cylinders       406 non-null float64
Displacement    406 non-null float64
Horsepower      406 non-null float64
Weight          406 non-null float64
Acceleration    406 non-null float64
Model           406 non-null object
Origin          406 non-null object
dtypes: float64(6), object(3)
memory usage: 28.7+ KB
```

In [62]:

```
cars.head()
```

Out[62]:

	Car	MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model	Origi
1	Chevrolet Chevelle Malibu	18.0	8.0	307.0	130.0	3504.0	12.0	70	U
2	Buick Skylark 320	15.0	8.0	350.0	165.0	3693.0	11.5	70	U
3	Plymouth Satellite	18.0	8.0	318.0	150.0	3436.0	11.0	70	U
4	AMC Rebel SST	16.0	8.0	304.0	150.0	3433.0	12.0	70	U
5	Ford Torino	17.0	8.0	302.0	140.0	3449.0	10.5	70	U

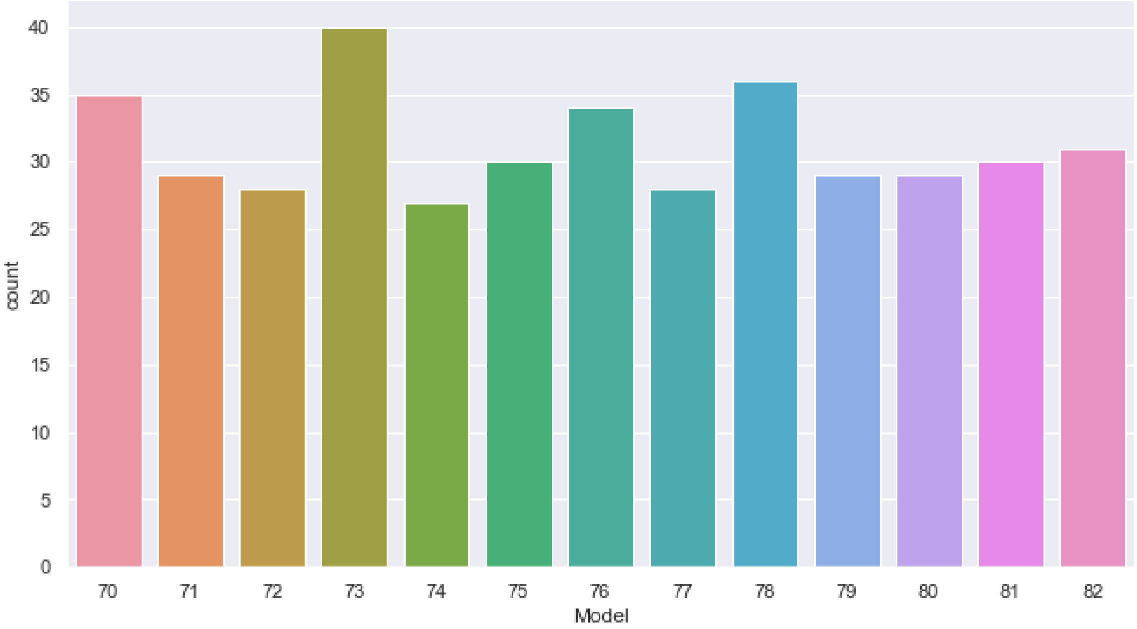
Now if I want to check the models of car our data is taking and the countries we are talking about .So we can go for sns.countplot over here

In [90]:

```
plt.figure(figsize = (11,6))
sns.countplot(data=cars, x='Model')
```

Out[90]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x16515a613c8>
```

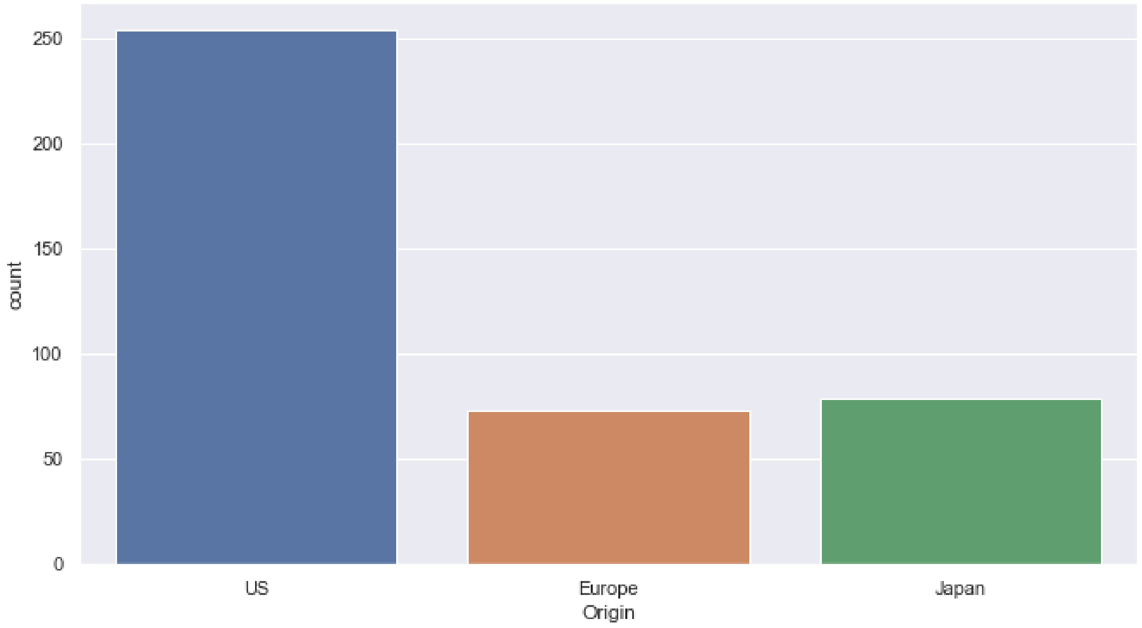


In [91]:

```
plt.figure(figsize = (11,6))
sns.countplot(data=cars, x='Origin')
```

Out[91]:

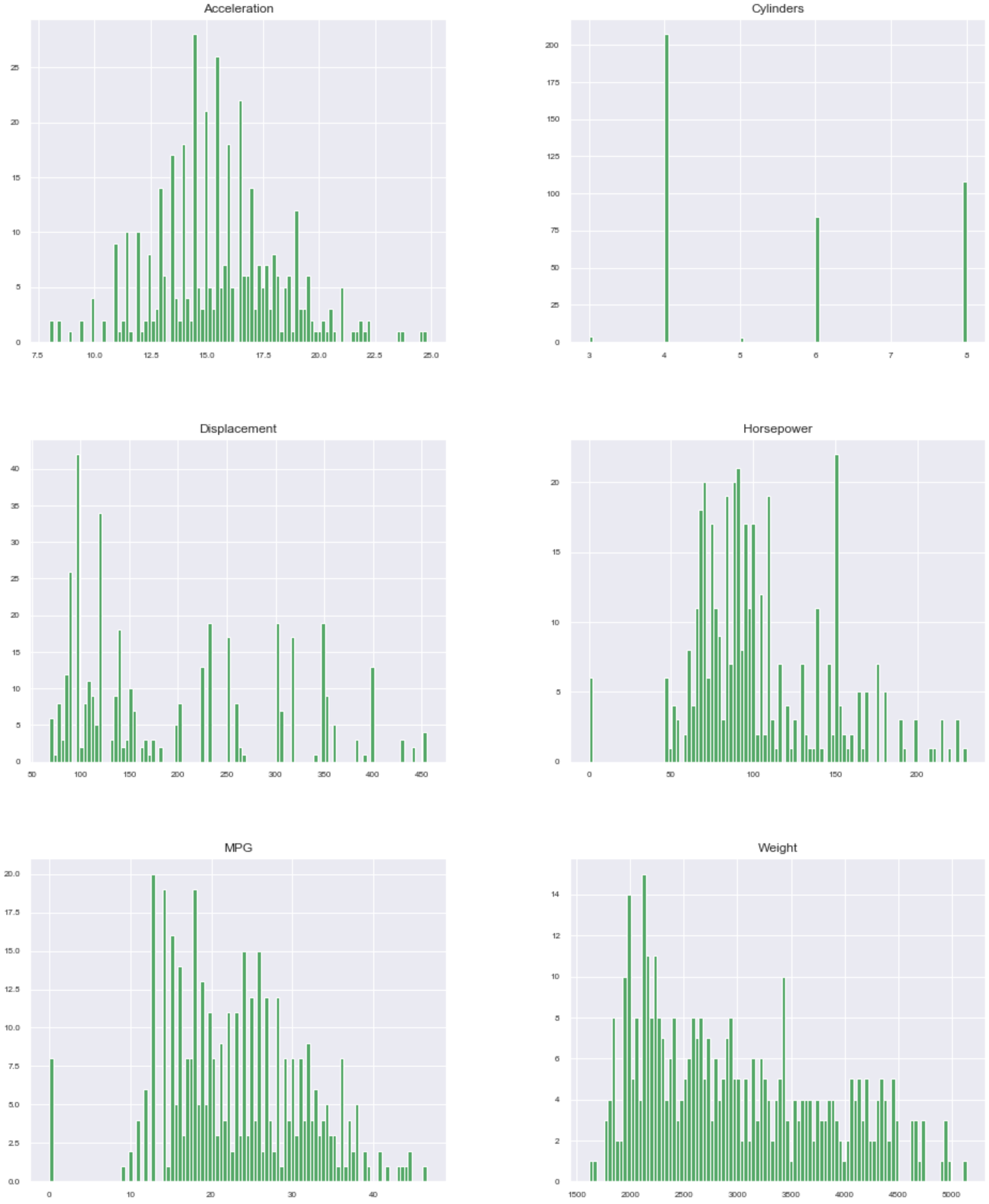
```
<matplotlib.axes._subplots.AxesSubplot at 0x165152dbb48>
```



Now we start visualise the data .We can use here histogram function .So the x axis represent column of our data while y axis represent total count of it .Like for example the plot with cylinders represent x axis as cylinder while y axis as counts of those cylinders

In [63]:

```
cars.hist(figsize=(16, 20), bins=100, xlabelsize=8, ylabelsize=8,color='g');
```


Acceleration
0
5
10
15
20
25
7.5
10.0
12.5
15.0
17.5
20.0
22.5
25.0
Cylinders
0
25
50
75
100
125
150
175
200
3
4
5
6
7
8
Displacement
0
5
10
15
20
25
30
35
40
50
100
150
200
250
300
350
400
450
Horsepower
0
5
10
15
20
0
50
100
150
200
MPG
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
0
10
20
30
40
Weight
0
2
4
6
8
10
12
14
1500
2000
2500
3000
3500
4000
4500
5000

Conclusion from above histogram chart
1.There are very few cars having acceleration nearlt 24.5 2.The data we are seeing having highest number of 4 cylinders followed by 8 and then 6 3.Maximun cars have displacement near to 100 4.There are few cars above 5000 weight

We can use boolean methos to find out some quick questions about our data , like car having

hightest acceleration , horsepower ,min acceleration

In [65]:

```
cars.loc[cars["Acceleration"] == cars["Acceleration"].max()]
```

Out[65]:

	Car	MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model	Ori
307	Peugeot 504	27.2	4.0	141.0	71.0	3190.0	24.8	79	Eur

In [66]:

```
cars.loc[cars["Horsepower"] == cars["Horsepower"].max()]
```

Out[66]:

	Car	MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model	Orig
124	Pontiac Grand Prix	16.0	8.0	400.0	230.0	4278.0	9.5	73	U

In [67]:

```
cars.loc[cars["Acceleration"] == cars["Acceleration"].min()]
```

Out[67]:

	Car	MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model	Orig
17	Plymouth 'Cuda 340	14.0	8.0	340.0	160.0	3609.0	8.0	70	U
18	Ford Mustang Boss 302	0.0	8.0	302.0	140.0	3353.0	8.0	70	U

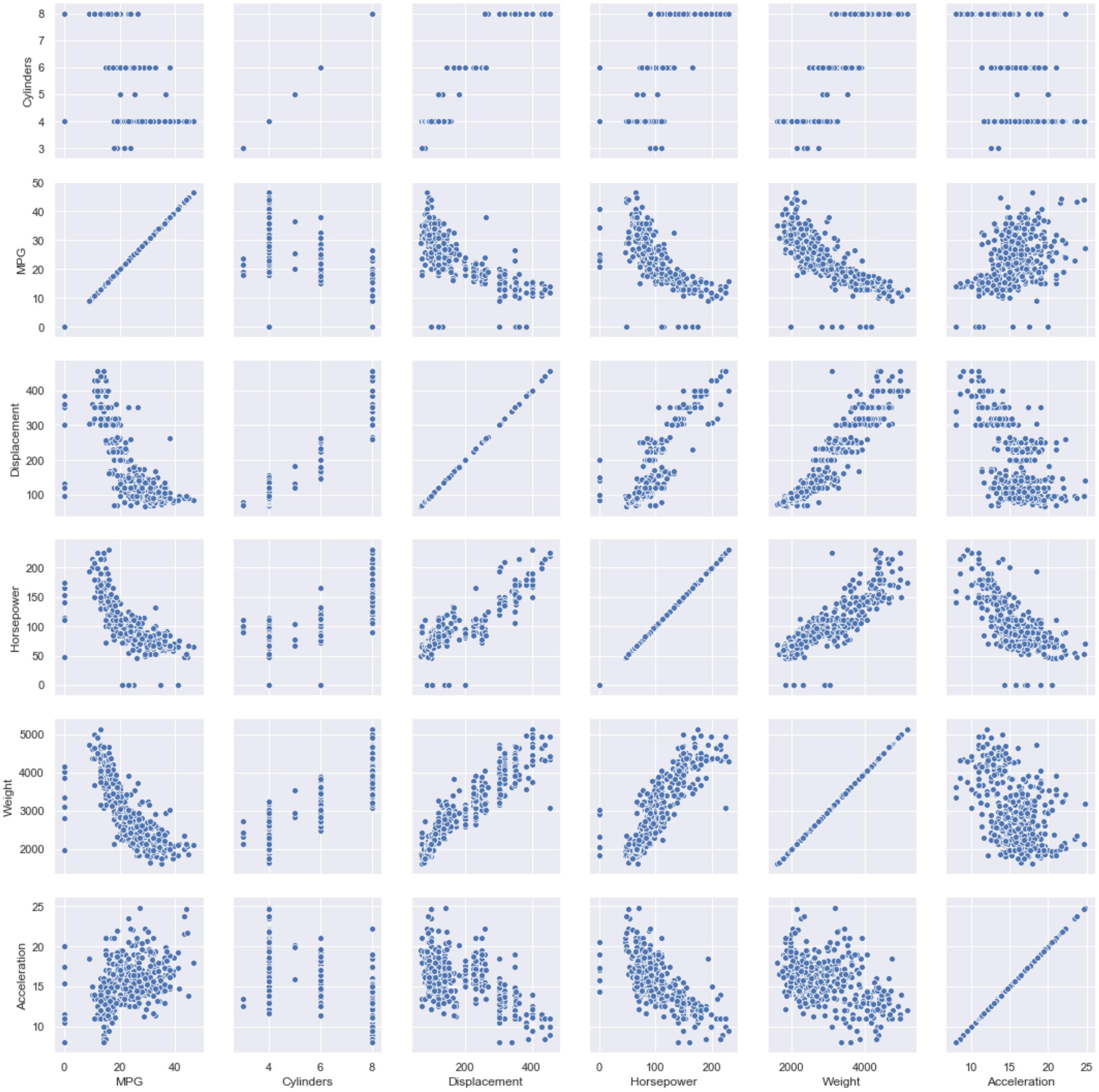
The second plot that is very useful is sns.pairplot .This plot tells how one parameter depends on different parameters.

In [102]:

```
# sns.pairplot(cars, vars=["Cylinders", "Acceleration"])
sns.pairplot(cars,x_vars=['MPG','Cylinders','Displacement','Horsepower','Weight','Acceleration'],y_vars=["Cylinders",'MPG','Displacement','Horsepower','Weight','Acceleration'])
```

Out[102]:

```
<seaborn.axisgrid.PairGrid at 0x1651748d108>
```


Cylinders
MPG
Displacement
Horsepower
Weight
Acceleration
3
4
5
6
7
8
0
10
20
30
40
50
100
200
300
400
0
50
100
150
200
2000
3000
4000
5000
10
15
20
25
0
20
40
MPG
4
6
8
Cylinders
200
400
Displacement
0
100
200
Horsepower
2000
4000
Weight
10
15
20
25
Acceleration

Conclusion from above plot **1.As you can see from acceleration row , it mainly depen on cylinders as when cylinders are increased, acceleration range goes down .also the acc depends on displacement and horsepower but these two have negative impact on accleration**

In this I have ploted a horizontal bar plot and i have taken a specific group of cars having 82 model USA and of cylinders 4 and i am comparing acceleration

In [75]:

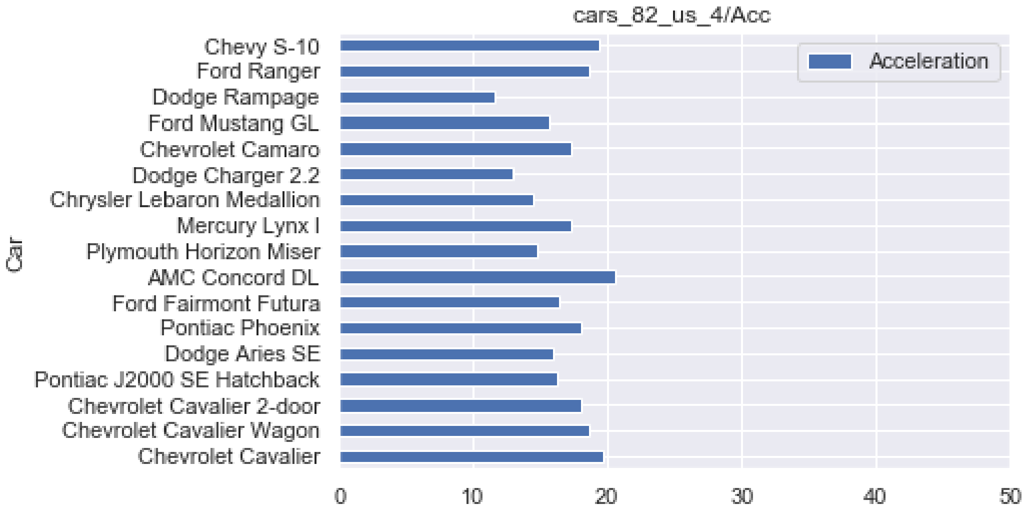
```
cars_82=cars.groupby('Model').get_group('82')
cars_82_us=cars_82.groupby('Origin').get_group('US')
cars_82_us_4=cars_82_us.groupby('Cylinders').get_group(4)
cars_82_us_6=cars_82_us.groupby('Cylinders').get_group(6)
```

In [77]:

```
cars_82_us_4.plot(x='Car', y='Acceleration', kind='barh', title='cars_82_us_4/Acc ', xl
im=(0,50))
```

Out[77]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x16512199788>
```



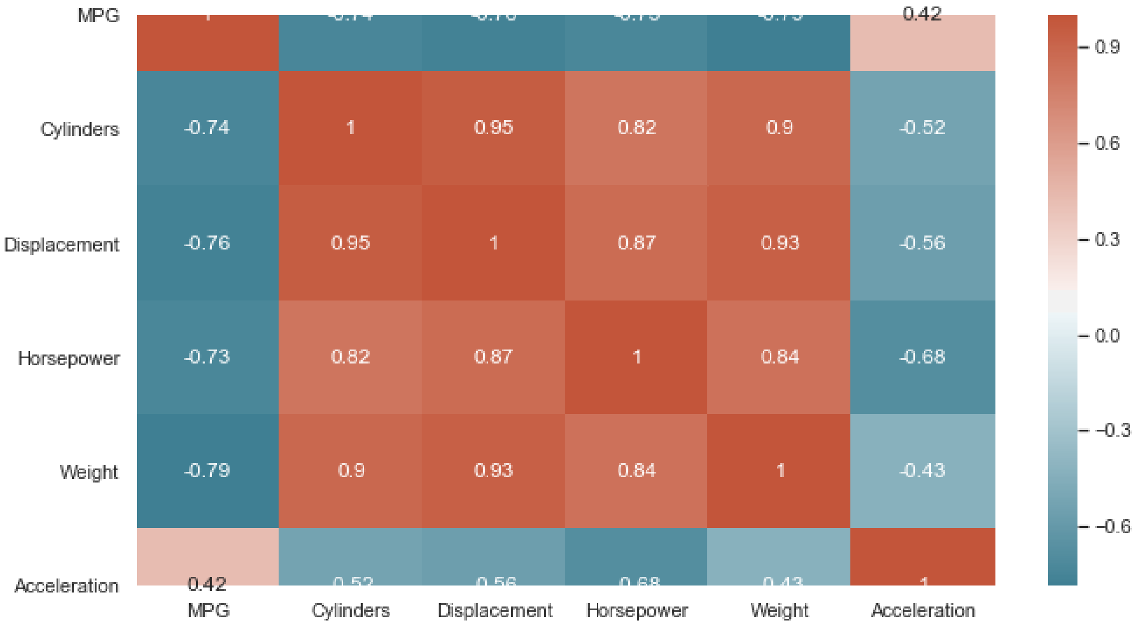
Heat map is a powerful tool for understanding correlation between different parameters

In [92]:

```
plt.figure(figsize = (11,6))
sns.heatmap(cars.corr(),
            xticklabels=cars.corr().columns,
            yticklabels=cars.corr().columns,
            annot=True,
            cmap=sns.diverging_palette(220,20,
            as_cmap=True))
```

Out[92]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x165151f3d08>
```



In [78]:

```
cars_82_us_4.columns
```

Out[78]:

```
Index(['Car', 'MPG', 'Cylinders', 'Displacement', 'Horsepower', 'Weight',
       'Acceleration', 'Model', 'Origin'],
      dtype='object')
```

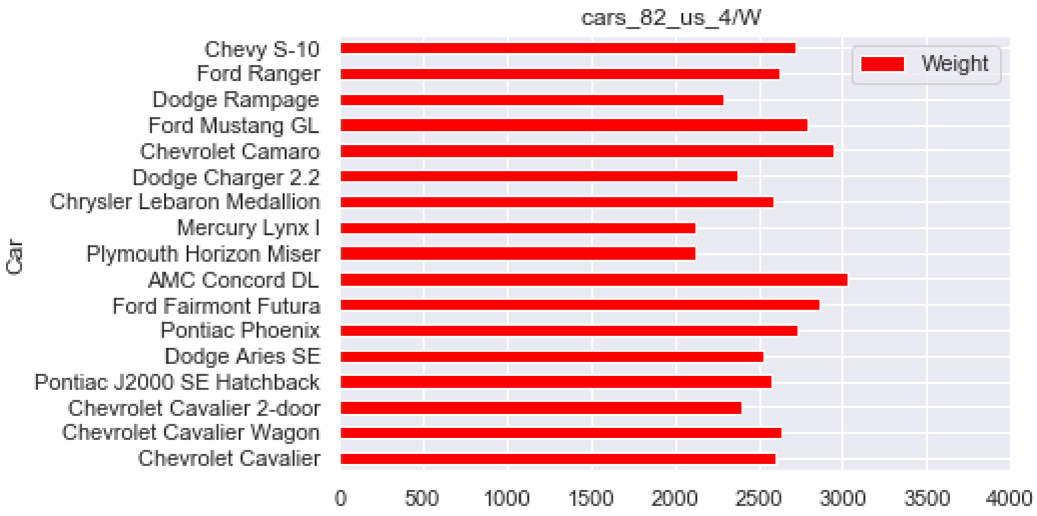
The value near to 1 means highly related while values closes to 0 means less related

In [79]:

```
cars_82_us_4.plot(x='Car', y='Weight', kind='barh', title='cars_82_us_4/W ', xlim=(0,40
00),color='red')
```

Out[79]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1651268a6c8>
```



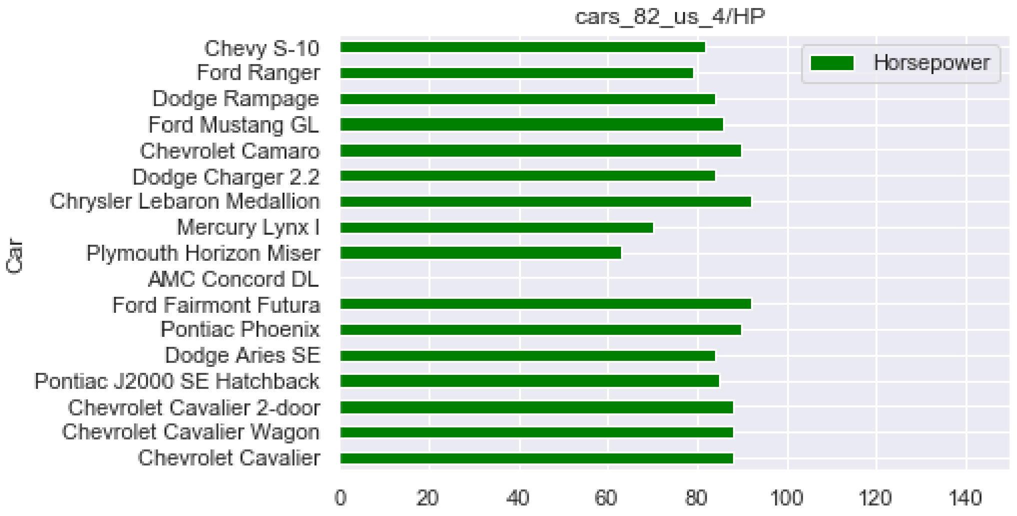
Here i have use the catplot to get a general overview .Like CComparing different cars on the basis of origin , Model,acceleration and cylinders

In [80]:

```
cars_82_us_4.plot(x='Car', y='Horsepower', kind='barh', title='cars_82_us_4/HP ', xlim=
(0,150),color='green')
```

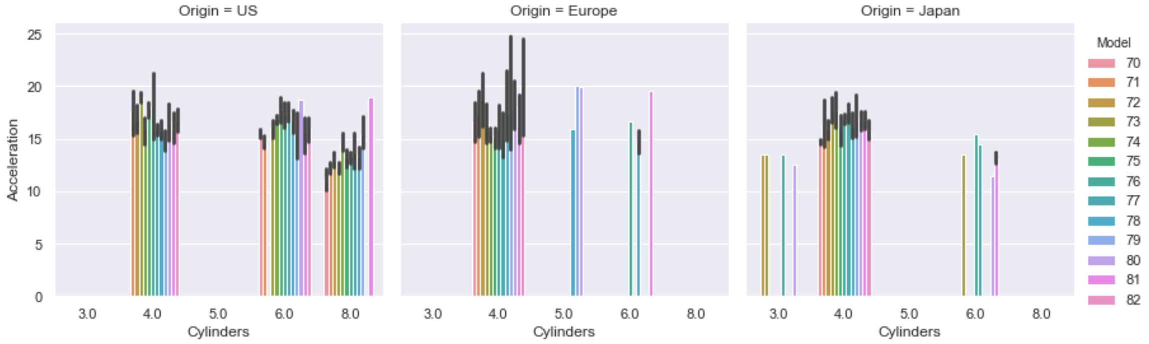
Out[80]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x165108fac08>
```



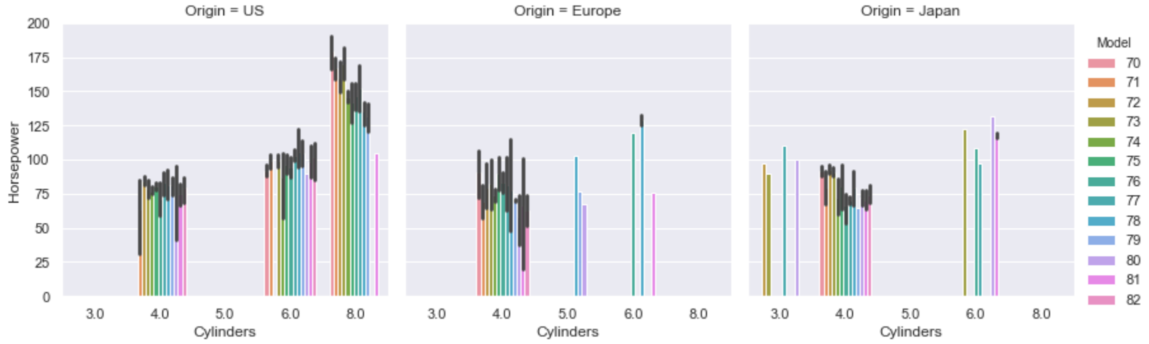
In [93]:

```
g = sns.catplot(x="Cylinders", y="Acceleration",
                hue="Model", col="Origin",
                data=cars, kind="bar",
                height=4);
```



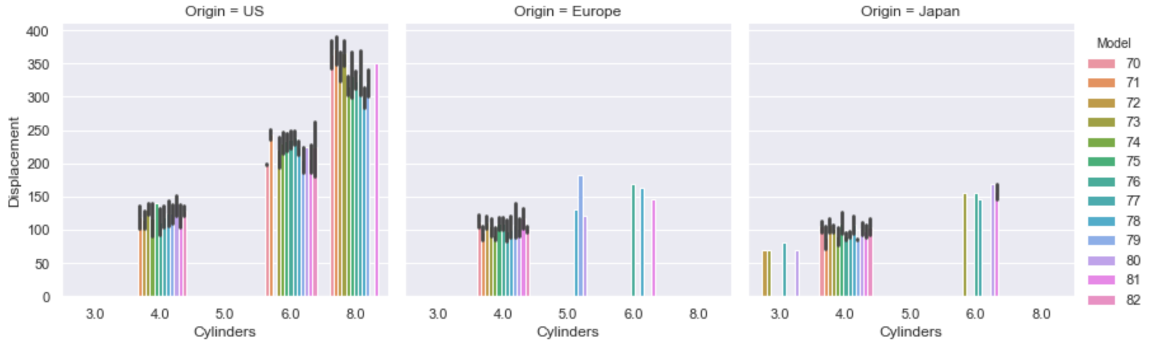
In [94]:

```
g = sns.catplot(x="Cylinders", y="Horsepower",
                hue="Model", col="Origin",
                data=cars, kind="bar",
                height=4);
```



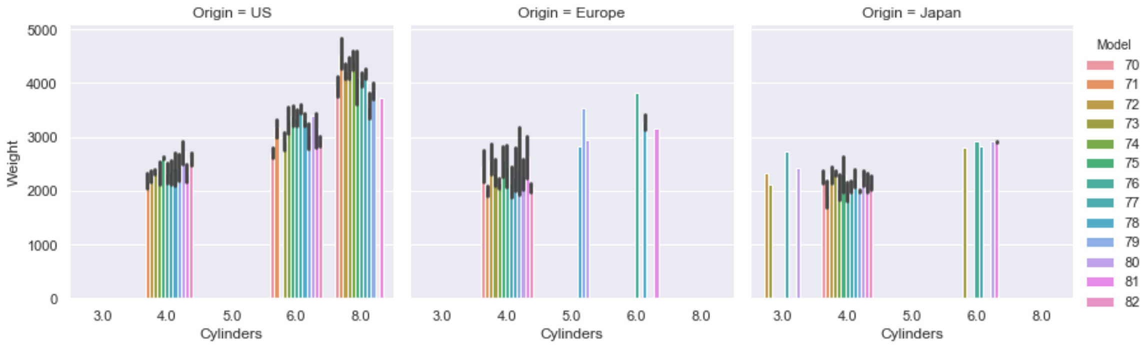
In [95]:

```
g = sns.catplot(x="Cylinders", y="Displacement",
                hue="Model", col="Origin",
                data=cars, kind="bar",
                height=4);
```



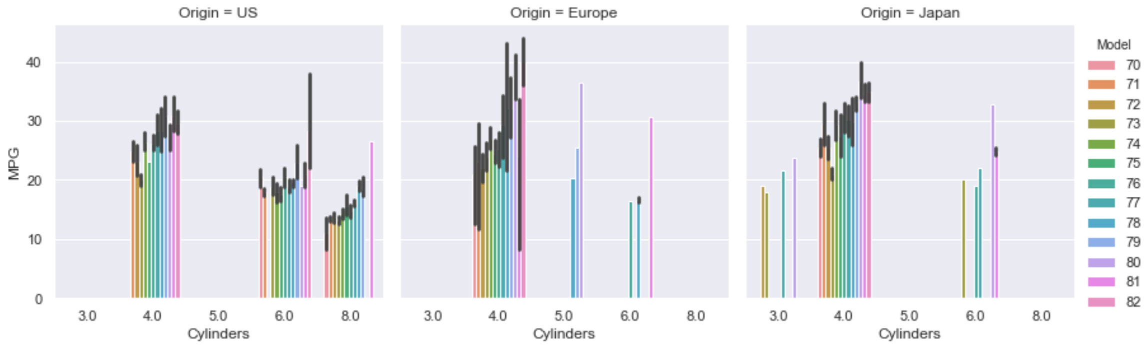
In [96]:

```
g = sns.catplot(x="Cylinders", y="Weight",
                hue="Model", col="Origin",
                data=cars, kind="bar",
                height=4);
```



In [97]:

```
g = sns.catplot(x="Cylinders", y="MPG",
                hue="Model", col="Origin",
                data=cars, kind="bar",
                height=4);
```



In [ ]:

```
```