Ans1.

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
```

In [2]:
```
df=pd.read_csv('download.csv')
```

In [3]:
```
df.head()
```

Out[3]:

	car name	cyl	disp	hp	wt	acc	yr	mpg
0	chevrolet chevelle malibu	8	307.0	130	3504	12.0	70	18.0
1	buick skylark 320	8	350.0	165	3693	11.5	70	15.0
2	plymouth satellite	8	318.0	150	3436	11.0	70	18.0
3	amc rebel sst	8	304.0	150	3433	12.0	70	16.0
4	ford torino	8	302.0	140	3449	10.5	70	17.0

In [4]:
```
X=df.drop('car name', axis=1)
```

In [5]:
```
X.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
Data columns (total 7 columns):
cyl     398 non-null int64
disp    398 non-null float64
hp      398 non-null object
wt      398 non-null int64
acc     398 non-null float64
yr      398 non-null int64
mpg     398 non-null float64
dtypes: float64(3), int64(3), object(1)
memory usage: 21.8+ KB
```

In [6]:
```
X['hp'].replace(to_replace='?', value=np.NaN, inplace=True)
```

In [7]:
```
X['hp']=X['hp'].astype('float32')
```

In [8]:
```
X['hp'].fillna(X['hp'].median(), inplace=True)
```

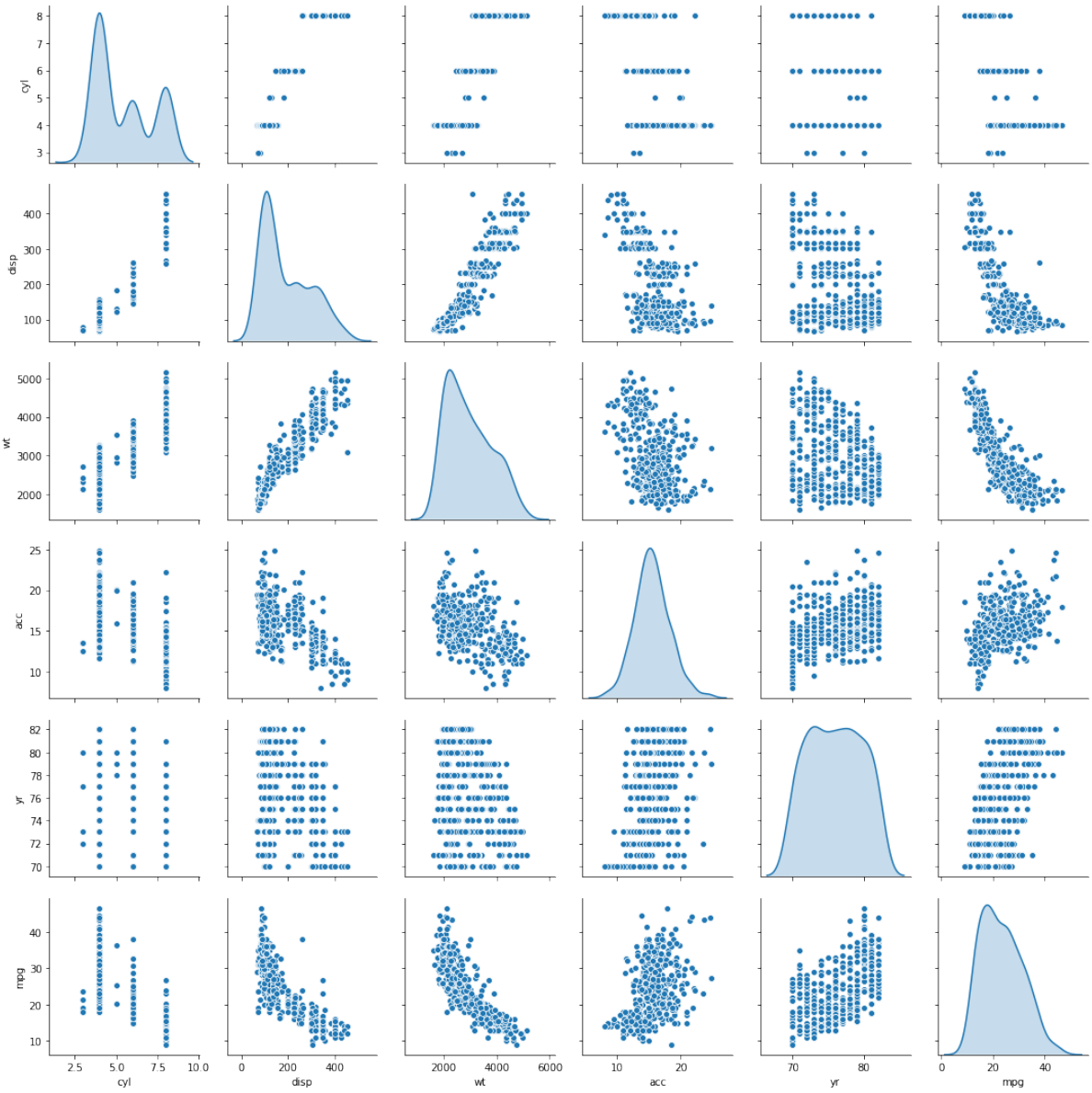
Ans2.

In [10]:
```
sns.pairplot(df, diag_kind ='kde')
```

```
/Users/ghost/anaconda3/lib/python3.7/site-packages/scipy/stats/sta
ts.py:1713: FutureWarning: Using a non-tuple sequence for multidim
ensional indexing is deprecated; use `arr[tuple(seq)]` instead of
`arr[seq]`. In the future this will be interpreted as an array ind
ex, `arr[np.array(seq)]`, which will result either in an error or
a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sum
val
/Users/ghost/anaconda3/lib/python3.7/site-packages/scipy/stats/sta
ts.py:1713: FutureWarning: Using a non-tuple sequence for multidim
ensional indexing is deprecated; use `arr[tuple(seq)]` instead of
`arr[seq]`. In the future this will be interpreted as an array ind
ex, `arr[np.array(seq)]`, which will result either in an error or
a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sum
val
/Users/ghost/anaconda3/lib/python3.7/site-packages/scipy/stats/sta
ts.py:1713: FutureWarning: Using a non-tuple sequence for multidim
ensional indexing is deprecated; use `arr[tuple(seq)]` instead of
`arr[seq]`. In the future this will be interpreted as an array ind
ex, `arr[np.array(seq)]`, which will result either in an error or
a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sum
val
/Users/ghost/anaconda3/lib/python3.7/site-packages/scipy/stats/sta
ts.py:1713: FutureWarning: Using a non-tuple sequence for multidim
ensional indexing is deprecated; use `arr[tuple(seq)]` instead of
`arr[seq]`. In the future this will be interpreted as an array ind
ex, `arr[np.array(seq)]`, which will result either in an error or
a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sum
val
/Users/ghost/anaconda3/lib/python3.7/site-packages/scipy/stats/sta
ts.py:1713: FutureWarning: Using a non-tuple sequence for multidim
ensional indexing is deprecated; use `arr[tuple(seq)]` instead of
`arr[seq]`. In the future this will be interpreted as an array ind
ex, `arr[np.array(seq)]`, which will result either in an error or
a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sum
val
/Users/ghost/anaconda3/lib/python3.7/site-packages/scipy/stats/sta
ts.py:1713: FutureWarning: Using a non-tuple sequence for multidim
ensional indexing is deprecated; use `arr[tuple(seq)]` instead of
`arr[seq]`. In the future this will be interpreted as an array ind
ex, `arr[np.array(seq)]`, which will result either in an error or
a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sum
```

```
val
```

Out[10]: <seaborn.axisgrid.PairGrid at 0x1a145fb2e8>



In [11]:
```
from sklearn.cluster import KMeans
from scipy.spatial import distance
```

In [100]:
```
distortion=[]
```

In [101]:
```
cluster_range=range(2,6)
```

In [102]:
```
cluster_error=[]
```

In [103]:
```
ndf.head()
```

Out[103]:

	cyl	disp	hp	wt	acc	yr	mpg	group
0	1.498191	1.090604	0.673118	0.630870	-1.295498	-1.627426	-0.706439	1
1	1.498191	1.503514	1.589958	0.854333	-1.477038	-1.627426	-1.090751	1
2	1.498191	1.196232	1.197026	0.550470	-1.658577	-1.627426	-0.706439	1
3	1.498191	1.061796	1.197026	0.546923	-1.295498	-1.627426	-0.962647	1
4	1.498191	1.042591	0.935072	0.565841	-1.840117	-1.627426	-0.834543	1

Ans3.

In [104]:
```
for num in cluster_range:
    clusters=KMeans(n_clusters=num, n_init=5 )
    clusters.fit(X)
    centroids=clusters.cluster_centers_
    cluster_error.append(clusters.inertia_)
    distortion.append(sum(np.min(distance.cdist(X, centroids, 'eucl
```

In [105]:
```
len(cluster_error)
```

Out[105]: 4

In [106]:
```
cluster_df=pd.DataFrame({'Number Cluster':cluster_range, 'Cluster E
```

In [107]:
```
cluster_df
```

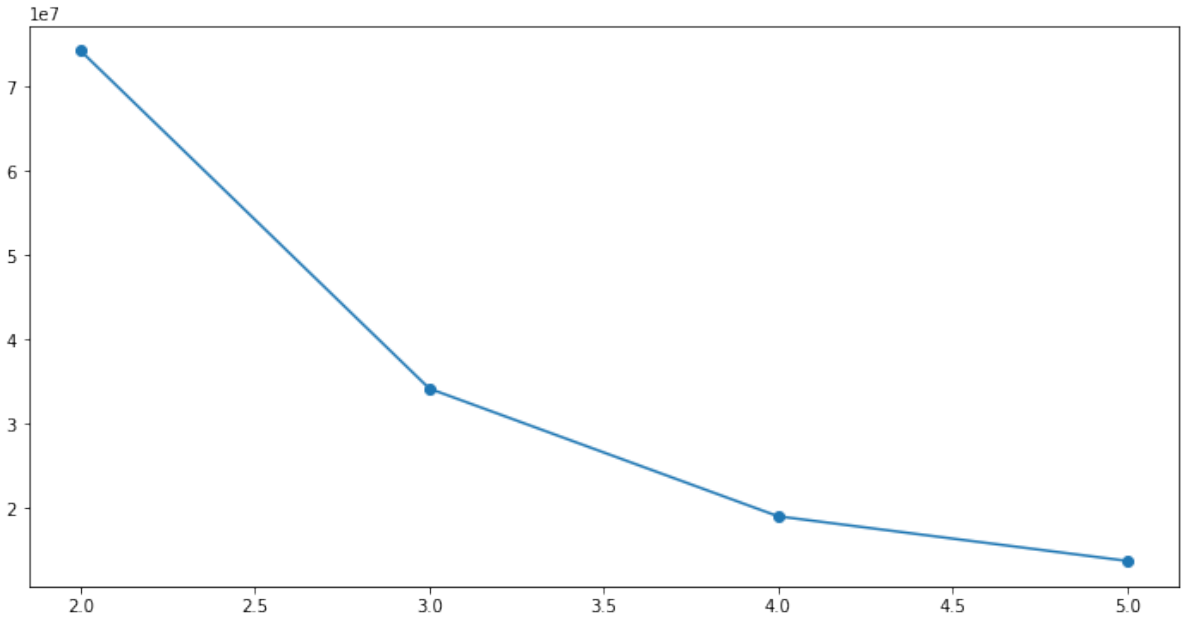
Out[107]:

	Number Cluster	Cluster Errors
0	2	7.428987e+07
1	3	3.420852e+07
2	4	1.905701e+07
3	5	1.376963e+07

In [108]:
```
from matplotlib import pyplot as plt
```

In [109]:
```
plt.figure(figsize=(12,6))
plt.plot(cluster_df['Number Cluster'], cluster_df['Cluster Errors']
```

Out[109]:
```
[<matplotlib.lines.Line2D at 0x1a20112e10>]
```

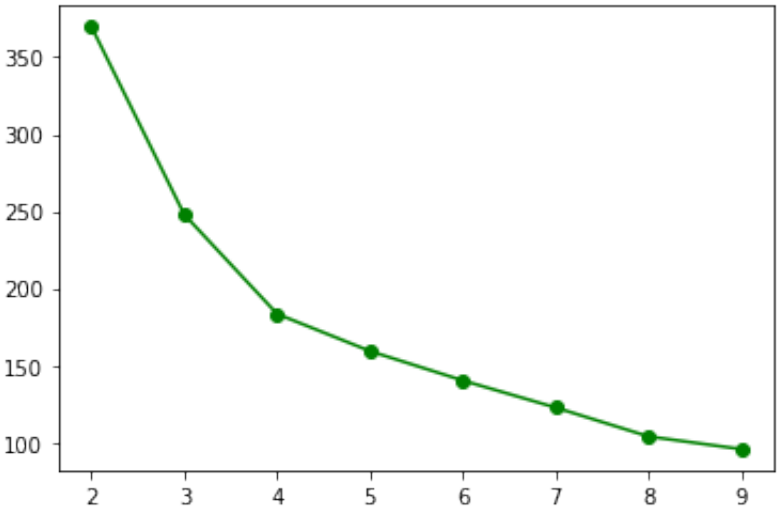


In [80]:
```
distortion
```

Out[80]:
```
[369.7424310089116,
 247.81798687691148,
 183.36581096312557,
 159.23196020706771,
 140.14818551033127,
 122.59425198839452,
 103.83216987082987,
 95.61753934919263]
```

In [82]:
```
plt.plot( range(2,10), distortion, 'go-')
```

Out[82]:
```
[<matplotlib.lines.Line2D at 0x1a200f8cc0>]
```



In [116]:
```
kmeans = KMeans(n_clusters=3)
kmeans = kmeans.fit(ndf)
labels = kmeans.predict(ndf)
centroids = kmeans.cluster_centers_
print("Centroid values")
print(centroids)
print(labels)
```

```
Centroid values
[[-0.85347696 -0.80321374 -0.67506205 -0.78549879  0.36133415  0.3
0992304
   0.75394661]
 [ 1.49819126  1.50068407  1.5141292   1.40001604 -1.07612225 -0.6
725559
  -1.15871315]
 [ 0.35772459  0.24687769 -0.04275486  0.30829922  0.28578589 -0.0
0272145
  -0.47905415]]
[1 1 1 1 1 1 1 1 1 1 1 1 1 0 2 2 2 0 0 0 0 0 0 2 1 1 1 1 0 0 0 0
2 2 2 2
 2 1 1 1 1 1 1 1 2 0 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
1 0 1 1
 1 1 2 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 0 1 1 1 1
2 0 0 0
 0 0 2 0 1 1 0 0 0 2 1 0 2 1 2 2 2 2 0 0 0 0 2 2 2 1 1 1 1 1 0 0 0
0 0 0 0
 0 0 0 0 2 2 2 2 1 1 1 1 2 2 2 2 2 2 1 0 0 2 0 0 0 0 2 0 2 0 0 0 0
0 0 0 0
 0 0 1 1 1 1 2 2 2 2 0 0 0 0 2 2 2 2 0 0 0 0 2 1 2 2 2 1 1 1 1 0 0
0 0 0 1
 2 1 1 2 2 2 2 1 1 1 1 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 2 1 1 2 2 2
0 2 2 2
 2 2 2 1 2 1 1 0 0 0 0 0 0 0 0 2 2 0 2 0 0 2 2 0 2 2 2 1 1 1 1 1 2
1 0 0 0
 0 2 2 0 2 0 0 0 0 0 2 2 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0
 2 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 2 2
2 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 2 0 0 0 0 0 0 0 0]
```

In [117]:
```
ndf.head()
```

Out[117]:

	cyl	disp	hp	wt	acc	yr	mpg
0	1.498191	1.090604	0.673118	0.630870	-1.295498	-1.627426	-0.706439
1	1.498191	1.503514	1.589958	0.854333	-1.477038	-1.627426	-1.090751
2	1.498191	1.196232	1.197026	0.550470	-1.658577	-1.627426	-0.706439
3	1.498191	1.061796	1.197026	0.546923	-1.295498	-1.627426	-0.962647
4	1.498191	1.042591	0.935072	0.565841	-1.840117	-1.627426	-0.834543

In [118]:
```
colnames = ndf.columns
prediction= kmeans.predict(ndf)
ndf1 = pd.DataFrame(ndf, columns= colnames)
ndf1["group"] = prediction


ndf1.tail(5)
```

Out[118]:

	cyl	disp	hp	wt	acc	yr	mpg	group
393	-0.856321	-0.513026	-0.479482	-0.213324	0.011586	1.621983	0.446497	0
394	-0.856321	-0.925936	-1.370127	-0.993671	3.279296	1.621983	2.624265	0
395	-0.856321	-0.561039	-0.531873	-0.798585	-1.440730	1.621983	1.087017	0
396	-0.856321	-0.705077	-0.662850	-0.408411	1.100822	1.621983	0.574601	0
397	-0.856321	-0.714680	-0.584264	-0.296088	1.391285	1.621983	0.958913	0

In [119]:
```
from scipy.stats import zscore
```

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

Out[33]:

	cyl	disp	hp	wt	acc	yr	mpg	group
0	8	307.0	130.0	3504	12.0	70	18.0	3
1	8	350.0	165.0	3693	11.5	70	15.0	4
2	8	318.0	150.0	3436	11.0	70	18.0	3
3	8	304.0	150.0	3433	12.0	70	16.0	3
4	8	302.0	140.0	3449	10.5	70	17.0	3

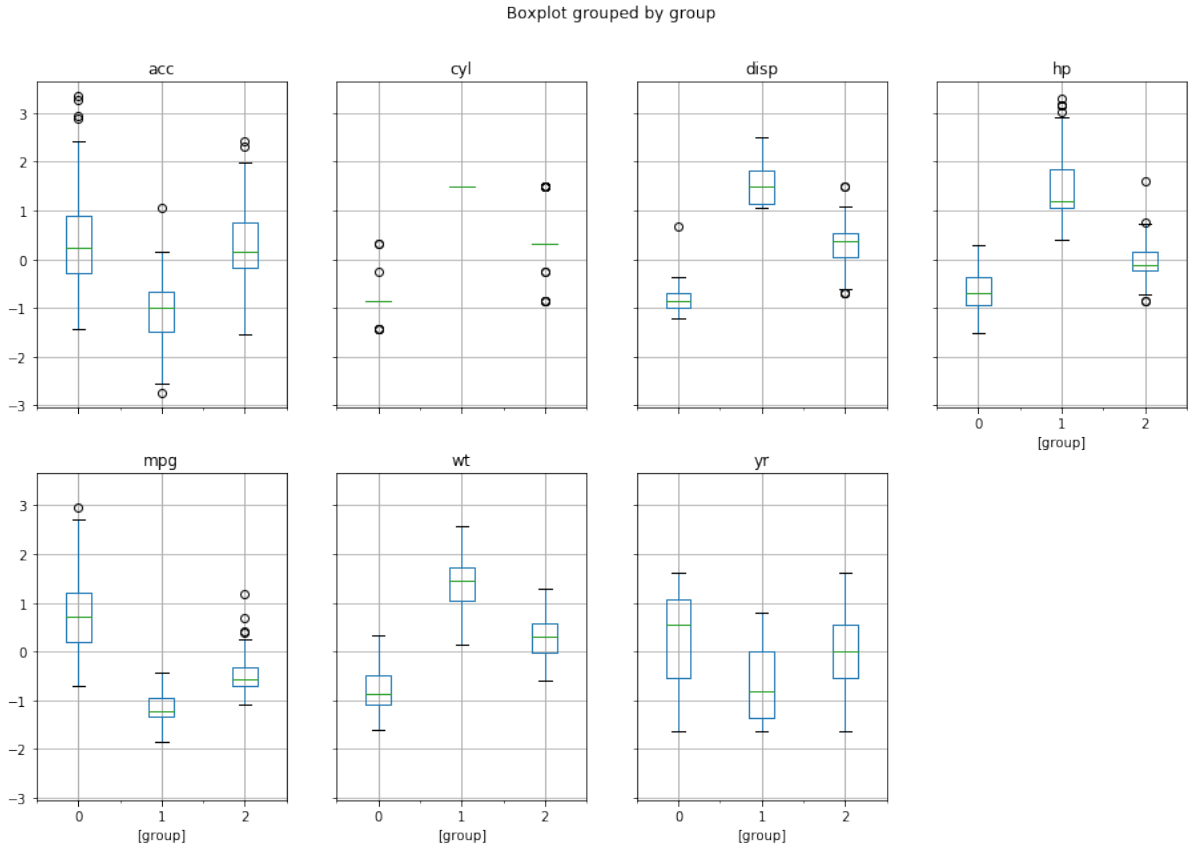
In [112]:
```
ndf=X.loc[:,'cyl':'mpg'].apply(zscore)
```

In [ ]:
```
ndf
```

In [120]:
```
import matplotlib.pylab as plt

ndf1.boxplot(by = 'group',  layout=(2,4), figsize=(15, 10))
```

Out[120]:
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x1a20a46
a58>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1a20151
128>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1a20ee2
390>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1a20f10
630>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x1a20f3f
8d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1a20f6f
b70>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1a20fa1
e10>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x1a20fde
128>]],
      dtype=object)
```



In [121]:
```
data = ndf1
def replace(group):
    median, std = group.median(), group.std()  #Get the median and
    outliers = (group - median).abs() > 2*std # Subtract median fro
    group[outliers] = group.median() # replacing group outliers wit
    return group

data_corrected = (data.groupby('group').transform(replace))
concat_data = data_corrected.join(pd.DataFrame(ndf1['group']))
```

```
/Users/ghost/anaconda3/lib/python3.7/site-packages/ipykernel_launc
her.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
(http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexin
g-view-versus-copy)
  """
/Users/ghost/anaconda3/lib/python3.7/site-packages/ipykernel_launc
her.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
(http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexin
g-view-versus-copy)
  """
```

In [166]:
```
concat_data
```

Out[166]:

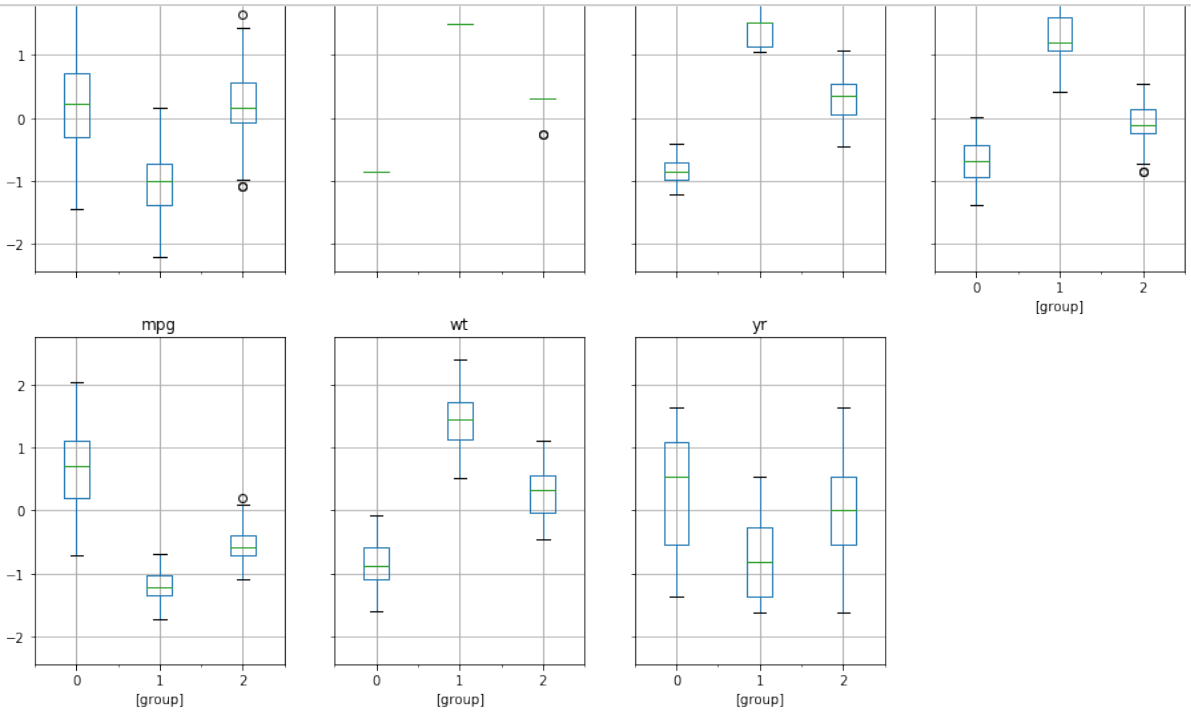
	cyl	disp	hp	wt	acc	yr	mpg	group
0	1.498191	1.090604	0.673118	0.630870	-1.295498	-1.627426	-0.706439	1
1	1.498191	1.503514	1.589958	0.854333	-1.477038	-1.627426	-1.090751	1
2	1.498191	1.196232	1.197026	0.550470	-1.658577	-1.627426	-0.706439	1
3	1.498191	1.061796	1.197026	0.546923	-1.295498	-1.627426	-0.962647	1
4	1.498191	1.042591	0.935072	0.565841	-1.840117	-1.627426	-0.834543	1
5	1.498191	2.262118	2.454408	1.620492	-2.021656	-1.627426	-1.090751	1
6	1.498191	1.503514	1.197026	1.635863	-1.005035	-1.627426	-1.218855	1
7	1.498191	1.503514	1.197026	1.586204	-1.005035	-1.627426	-1.218855	1
8	1.498191	1.503514	1.197026	1.719809	-2.021656	-1.627426	-1.218855	1
9	1.498191	1.887617	2.244844	1.039961	-1.005035	-1.627426	-1.090751	1
10	1.498191	1.820399	1.720935	0.700628	-2.021656	-1.627426	-1.090751	1
11	1.498191	1.407489	1.458981	0.755016	-1.005035	-1.627426	-1.218855	1

12	1.498191	1.983643	1.197026	0.934732	-2.203196	-1.627426	-1.090751	1
13	1.498191	1.503514	1.197026	1.441958	-2.021656	-1.627426	-1.218855	1
14	-0.856321	-0.772295	-0.243723	-0.707544	-0.206262	0.538847	0.062185	0
15	0.320935	0.043923	-0.243723	-0.162483	-0.024722	-1.627426	-0.194023	2
16	0.320935	0.053526	-0.191332	-0.232242	-0.024722	-1.627426	-0.706439	2
17	0.320935	0.063128	-0.505678	-0.453340	0.156817	-1.627426	-0.322127	2
18	-0.856321	-0.925936	-0.427091	-0.993671	-0.387801	0.538847	0.446497	0
19	-0.856321	-0.925936	-0.689046	-1.342463	1.790672	0.538847	0.318393	0
20	-0.856321	-0.801103	-0.453287	-0.352841	0.701436	0.538847	0.190289	0
21	-0.856321	-0.829911	-0.374700	-0.638968	-0.387801	0.538847	0.062185	0
22	-0.856321	-0.858718	-0.243723	-0.703997	0.701436	0.538847	0.190289	0
23	-0.856321	-0.695475	-0.689046	-0.870708	-1.113959	0.538847	0.318393	0
24	0.320935	0.053526	-0.374700	-0.381217	-0.206262	-1.627426	-0.322127	2
25	1.498191	1.599540	1.197026	1.944455	-0.569341	-1.627426	-1.731270	1
26	1.498191	1.090604	2.506799	1.661874	-0.206262	-1.627426	-1.731270	1
27	1.498191	1.196232	1.197026	1.668968	-0.750880	-1.627426	-1.603167	1
28	1.498191	1.061796	2.323431	2.082789	-1.005035	-1.627426	-1.218855	1
29	-0.856321	-0.925936	-0.427091	-0.993671	-0.387801	-1.356642	0.446497	0
...	...	...	...	...	...	...	...	...
368	-0.856321	-0.781898	-0.427091	-0.390676	1.100822	1.621983	0.446497	0
369	-0.856321	-0.781898	-0.427091	-0.680350	0.882975	1.621983	1.343225	0
370	-0.856321	-0.781898	-0.505678	-0.467528	0.229433	1.621983	0.958913	0
371	-0.856321	-0.561039	-0.531873	-0.526645	0.156817	1.621983	0.702705	0
372	-0.856321	-0.407398	-0.374700	-0.278353	0.882975	1.621983	0.446497	0
373	-0.856321	-0.513026	-0.322309	-0.124648	0.302049	1.621983	0.062185	0
374	-0.856321	-0.407398	-0.283016	-0.870708	1.790672	1.621983	-0.065919	0
375	-0.856321	-0.849116	-0.793827	-1.171023	-0.097338	1.621983	1.599433	0
376	-0.856321	-0.983552	-0.951000	-1.117818	0.955591	1.621983	1.727537	0
377	-0.856321	-0.983552	-0.951000	-1.182846	0.737743	1.621983	0.958913	0
378	-0.856321	-0.849116	-1.081977	-0.999583	-0.315185	1.621983	1.855641	0
379	-0.856321	-0.916334	-0.898609	-0.999583	0.628820	1.621983	1.599433	0
380	-0.856321	-0.705077	-0.427091	-0.958201	-0.387801	1.621983	1.599433	0
381	-0.856321	-0.829911	-0.767632	-0.904996	-0.387801	1.621983	1.599433	0
382	-0.856321	-0.820308	-0.898609	-0.857702	0.483588	1.621983	1.343225	0

383	-0.856321	-0.983552	-0.977196	-1.188758	-0.206262	1.621983	1.855641	0
384	-0.856321	-0.983552	-0.977196	-1.188758	0.047894	1.621983	1.087017	0
385	-0.856321	-0.983552	-0.977196	-1.153288	0.229433	1.621983	1.855641	0
386	0.320935	-0.119320	0.149209	-0.030061	0.302049	1.621983	0.190289	2
387	-0.856321	-0.849116	-0.505678	-0.870708	0.519896	1.621983	1.855641	0
388	-0.856321	-0.849116	-0.322309	-0.455705	-0.387801	1.621983	0.318393	0
389	0.320935	0.370411	0.201600	-0.160119	-0.315185	1.621983	-0.194023	2
390	-0.856321	-0.474616	-0.217528	-0.361117	-0.605648	1.621983	1.087017	0
391	-0.856321	-0.561039	-0.531873	-0.709909	-0.932419	1.621983	1.599433	0
392	-0.856321	-0.407398	-0.374700	-0.870708	0.628820	1.621983	0.446497	0
393	-0.856321	-0.513026	-0.479482	-0.213324	0.011586	1.621983	0.446497	0
394	-0.856321	-0.925936	-1.370127	-0.993671	0.229433	1.621983	0.702705	0
395	-0.856321	-0.561039	-0.531873	-0.798585	-1.440730	1.621983	1.087017	0
396	-0.856321	-0.705077	-0.662850	-0.408411	1.100822	1.621983	0.574601	0
397	-0.856321	-0.714680	-0.584264	-0.296088	1.391285	1.621983	0.958913	0

398 rows × 8 columns

In [122]:
```
concat_data.boxplot(by='group', layout=(2,4), figsize=(15,10))
```



In [124]:
```
for i in cluster_range:
    points = np.array(concat_data.loc[concat_data['group'] == i])

points
```

Out[124]:
```
array([], shape=(0, 8), dtype=float64)
```

In [157]:
```
points
```

Out[157]:
```
array([[-0.85632057, -0.77229532, -0.24372303, ...,  0.53884683,
         0.06218515,  0.        ],
       [-0.85632057, -0.92593647, -0.42709115, ...,  0.53884683,
         0.44649707,  0.        ],
       [-0.85632057, -0.92593647, -0.68904561, ...,  0.53884683,
         0.31839309,  0.        ],
       ...,
       [-0.85632057, -0.56103873, -0.53187293, ...,  1.62198339,
         1.08701694,  0.        ],
       [-0.85632057, -0.70507731, -0.66285014, ...,  1.62198339,
         0.57460104,  0.        ],
       [-0.85632057, -0.71467988, -0.5842638 , ...,  1.62198339,
         0.95891297,  0.        ]])
```

In [140]:
```
k=np.array(X.loc[concat_data['group'] == 1])
```

In [141]:
```
k
```

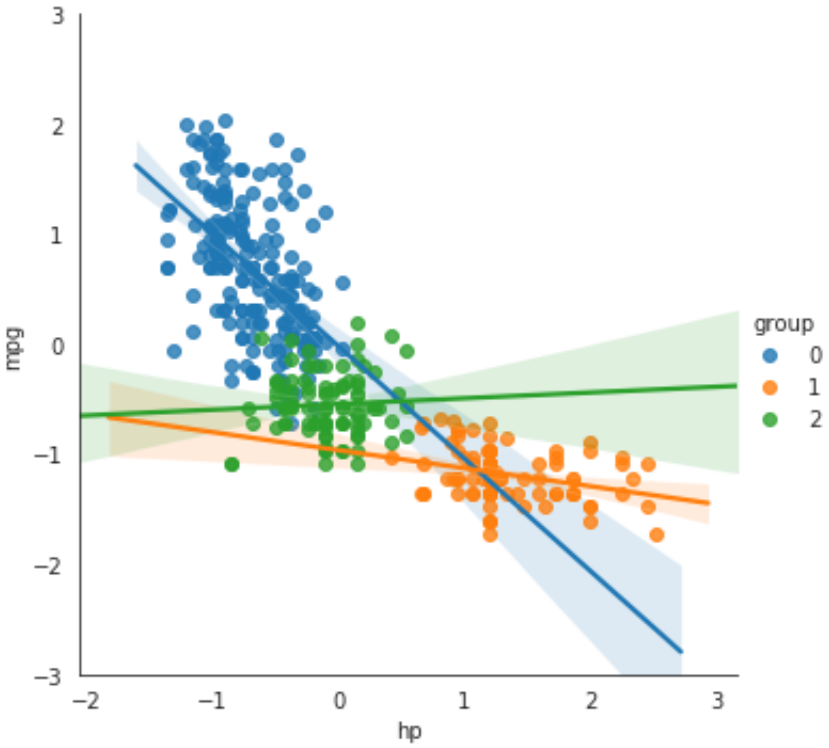
Out[141]:
```
array([[ 1.49819126,  1.0906037 ,  0.67311752,  0.63086987, -1.295
49834,
        -1.62742629, -0.7064387 ,  1.        ],
       [ 1.49819126,  1.5035143 ,  1.58995807,  0.85433297, -1.477
03779,
        -1.62742629, -1.09075062,  1.        ],
       [ 1.49819126,  1.19623199,  1.19702649,  0.55047045, -1.658
57724,
        -1.62742629, -0.7064387 ,  1.        ],
       [ 1.49819126,  1.06179598,  1.19702649,  0.54692342, -1.295
49834,
        -1.62742629, -0.96264665,  1.        ],
       [ 1.49819126,  1.04259084,  0.935072  ,  0.56584093, -1.840
11669,
        -1.62742629, -0.83454267,  1.        ],
       [ 1.49819126,  2.26211751,  2.45440769,  1.62049216, -2.021
65614,
        -1.62742629, -1.09075062,  1.        ],
       [ 1.49819126,  1.5035143 ,  1.19702649,  1.63586264, -1.005
03523,
```

In [168]:
```
var = 'hp'

with sns.axes_style("white"):
    plot = sns.lmplot(var,'mpg',data=concat_data,hue='group')
plot.set(ylim = (-3,3))
```

```
/Users/ghost/anaconda3/lib/python3.7/site-packages/scipy/stats/sta
ts.py:1713: FutureWarning: Using a non-tuple sequence for multidim
ensional indexing is deprecated; use `arr[tuple(seq)]` instead of
`arr[seq]`. In the future this will be interpreted as an array ind
ex, `arr[np.array(seq)]`, which will result either in an error or
a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sum
val
/Users/ghost/anaconda3/lib/python3.7/site-packages/scipy/stats/sta
ts.py:1713: FutureWarning: Using a non-tuple sequence for multidim
ensional indexing is deprecated; use `arr[tuple(seq)]` instead of
`arr[seq]`. In the future this will be interpreted as an array ind
ex, `arr[np.array(seq)]`, which will result either in an error or
a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sum
val
/Users/ghost/anaconda3/lib/python3.7/site-packages/scipy/stats/sta
ts.py:1713: FutureWarning: Using a non-tuple sequence for multidim
ensional indexing is deprecated; use `arr[tuple(seq)]` instead of
`arr[seq]`. In the future this will be interpreted as an array ind
ex, `arr[np.array(seq)]`, which will result either in an error or
a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sum
val
```

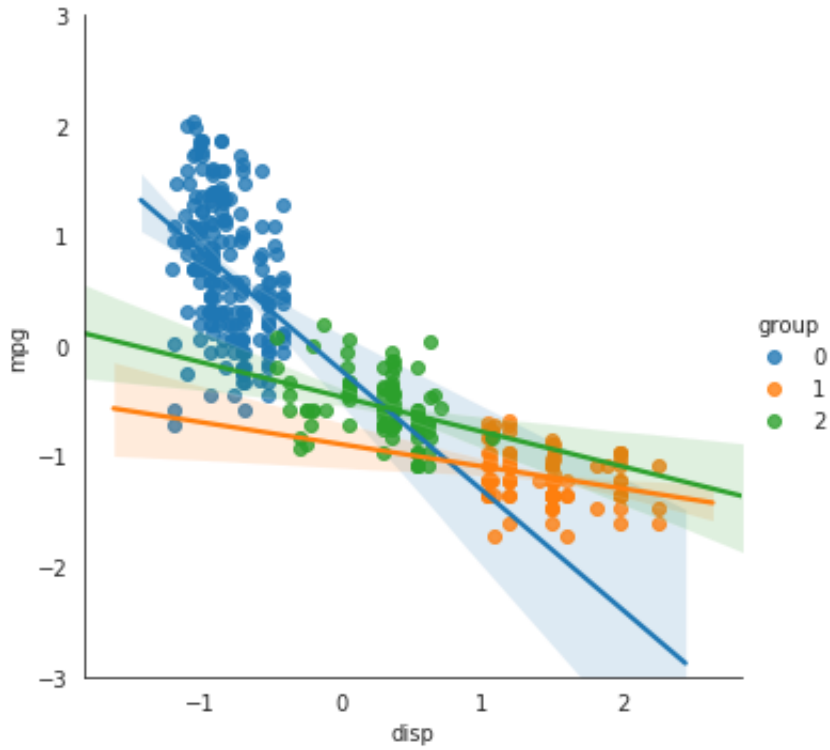
Out[168]: <seaborn.axisgrid.FacetGrid at 0x1a1cb4c710>



In [169]:
```
var = 'disp'
with sns.axes_style("white"):
    plot = sns.lmplot(var,'mpg',data=concat_data,hue='group')
plot.set(ylim = (-3,3))
```

```
/Users/ghost/anaconda3/lib/python3.7/site-packages/scipy/stats/sta
ts.py:1713: FutureWarning: Using a non-tuple sequence for multidim
ensional indexing is deprecated; use `arr[tuple(seq)]` instead of
`arr[seq]`. In the future this will be interpreted as an array ind
ex, `arr[np.array(seq)]`, which will result either in an error or
a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sum
val
/Users/ghost/anaconda3/lib/python3.7/site-packages/scipy/stats/sta
ts.py:1713: FutureWarning: Using a non-tuple sequence for multidim
ensional indexing is deprecated; use `arr[tuple(seq)]` instead of
`arr[seq]`. In the future this will be interpreted as an array ind
ex, `arr[np.array(seq)]`, which will result either in an error or
a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sum
val
/Users/ghost/anaconda3/lib/python3.7/site-packages/scipy/stats/sta
ts.py:1713: FutureWarning: Using a non-tuple sequence for multidim
ensional indexing is deprecated; use `arr[tuple(seq)]` instead of
`arr[seq]`. In the future this will be interpreted as an array ind
ex, `arr[np.array(seq)]`, which will result either in an error or
a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sum
val
```

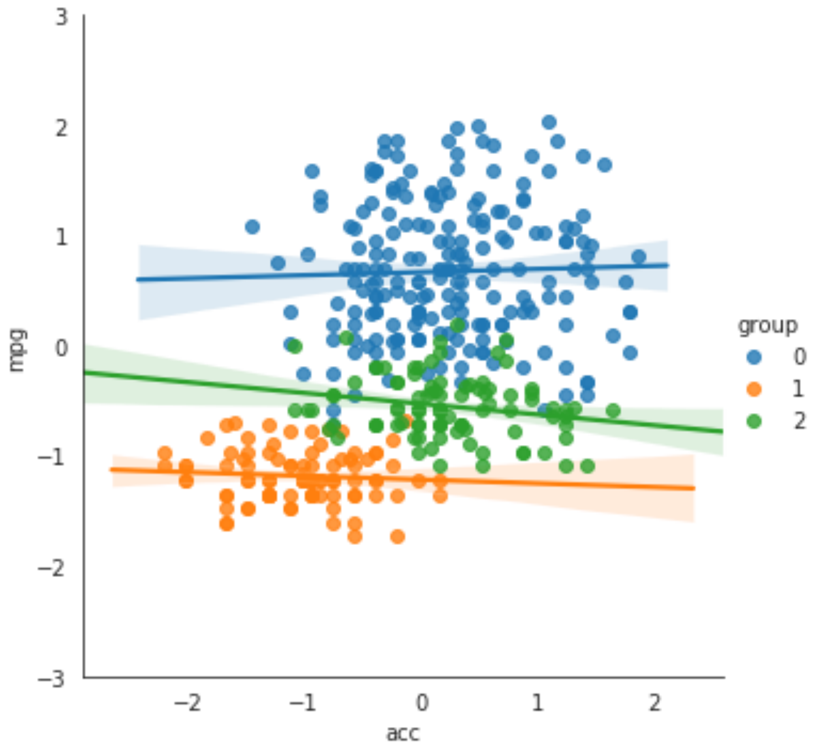
Out[169]: <seaborn.axisgrid.FacetGrid at 0x1a1e695518>



In [171]:
```
var = 'acc'
with sns.axes_style("white"):
    plot = sns.lmplot(var,'mpg',data=concat_data,hue='group')
plot.set(ylim = (-3,3))
```

```
/Users/ghost/anaconda3/lib/python3.7/site-packages/scipy/stats/sta
ts.py:1713: FutureWarning: Using a non-tuple sequence for multidim
ensional indexing is deprecated; use `arr[tuple(seq)]` instead of
`arr[seq]`. In the future this will be interpreted as an array ind
ex, `arr[np.array(seq)]`, which will result either in an error or
a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sum
val
/Users/ghost/anaconda3/lib/python3.7/site-packages/scipy/stats/sta
ts.py:1713: FutureWarning: Using a non-tuple sequence for multidim
ensional indexing is deprecated; use `arr[tuple(seq)]` instead of
`arr[seq]`. In the future this will be interpreted as an array ind
ex, `arr[np.array(seq)]`, which will result either in an error or
a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sum
val
/Users/ghost/anaconda3/lib/python3.7/site-packages/scipy/stats/sta
ts.py:1713: FutureWarning: Using a non-tuple sequence for multidim
ensional indexing is deprecated; use `arr[tuple(seq)]` instead of
`arr[seq]`. In the future this will be interpreted as an array ind
ex, `arr[np.array(seq)]`, which will result either in an error or
a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sum
val
```

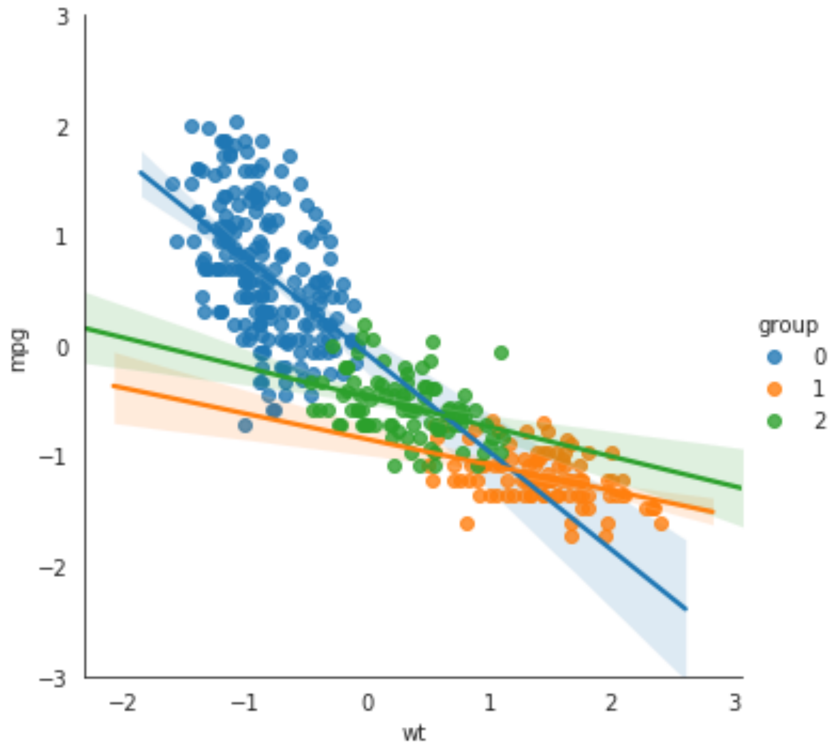
Out[171]: <seaborn.axisgrid.FacetGrid at 0x1a1e7b02b0>



In [172]:
```
var = 'wt'
with sns.axes_style("white"):
    plot = sns.lmplot(var,'mpg',data=concat_data,hue='group')
plot.set(ylim = (-3,3))
```

```
/Users/ghost/anaconda3/lib/python3.7/site-packages/scipy/stats/sta
ts.py:1713: FutureWarning: Using a non-tuple sequence for multidim
ensional indexing is deprecated; use `arr[tuple(seq)]` instead of
`arr[seq]`. In the future this will be interpreted as an array ind
ex, `arr[np.array(seq)]`, which will result either in an error or
a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sum
val
/Users/ghost/anaconda3/lib/python3.7/site-packages/scipy/stats/sta
ts.py:1713: FutureWarning: Using a non-tuple sequence for multidim
ensional indexing is deprecated; use `arr[tuple(seq)]` instead of
`arr[seq]`. In the future this will be interpreted as an array ind
ex, `arr[np.array(seq)]`, which will result either in an error or
a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sum
val
/Users/ghost/anaconda3/lib/python3.7/site-packages/scipy/stats/sta
ts.py:1713: FutureWarning: Using a non-tuple sequence for multidim
ensional indexing is deprecated; use `arr[tuple(seq)]` instead of
`arr[seq]`. In the future this will be interpreted as an array ind
ex, `arr[np.array(seq)]`, which will result either in an error or
a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sum
val
```

Out[172]:
```
<seaborn.axisgrid.FacetGrid at 0x1a2265ae80>
```



Ans 4.

In [179]:
```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

In [197]:
```
for i in range(0,3):
    x=concat_data.loc[concat_data['group'] == i]
    X=x.drop(['mpg','group'], axis=1)
    y=x.loc[:,'mpg']
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_
    model=LinearRegression()
    model.fit(X_train, y_train)
    y_pred=model.predict(X_test)
    print(accuracy.)
    print("Coef for ",(i+1),"Group" )
    print(model.coef_)
```

```
Coef for  1 Group
[ 0.         -0.19024435 -0.36276706 -0.73747192 -0.05948821  0.34
689352]
Coef for  2 Group
[ 0.33663539 -0.01667302 -0.08700691 -0.16693998 -0.06563639  0.07
330876]
Coef for  3 Group
[ 0.16023793  0.04094404 -0.03359065 -0.30852572 -0.00336713  0.14
410381]
```