In [1]:

```
# -*- coding: utf-8 -*-
"""
Created on Wed Sep 21 18:10:08 2022

@author: Brandon Botzer - btb5103
"""


"""
Assignment:

1. Plot am-based histogram to compare mpg (20 points)
2. Use scatterplot to plot mpg VS. hp (20 points)
3. Create a scatterplot matrix for new data consisting of columns [disp, hp, drat
4. Create boxplots for new data consisting of columns [disp, hp, drat, wt, qsect]
5. Use plots to answer which variable has the most impact on mpg. (20 points)

A note about plotting:  I was having some trouble with the interactive
plotting in Spyder.  I was unable to click and select any of the plots
to zoom or pan.  I did modify the settings as shown in the online notes
but this caused no plots to show up.

I ended up using Activate Support, Autoload pylab and NumPy, and Inline backend
just to get the plots to display. While still unable to dynamically interact,
Jupyter at least loaded the plots inline properly.

"""
#imports (may not need all of these but better safe than sorry later)
import os
from pandas import Series, DataFrame
import pandas as pd
import numpy as np
import csv
from numpy import NaN as NA

#Import a slew of plotting functions to play with
import matplotlib.pyplot as plt
import seaborn as sns

#Had to install plotly first and didn't really use for this work
import plotly.express as px

#regular expressions
import re
```

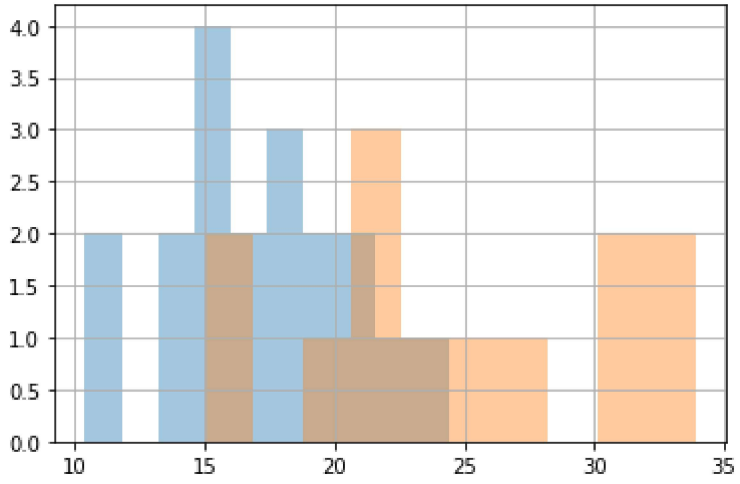
In [2]:
```
#Set the path for the CSV file
readPath = "J:\DSDegree\PennState\DAAN_862\Week 6\Homework"

#Change the directory
os.chdir(readPath)

#Read the CSV file in
mtcars = pd.read_csv("mtcars.csv")

print(mtcars)
```

```
                 model   mpg  cyl   disp   hp  drat     wt   qsec  vs  am  \
0            Mazda RX4  21.0    6  160.0  110  3.90  2.620  16.46   0   1
1        Mazda RX4 Wag  21.0    6  160.0  110  3.90  2.875  17.02   0   1
2           Datsun 710  22.8    4  108.0   93  3.85  2.320  18.61   1   1
3       Hornet 4 Drive  21.4    6  258.0  110  3.08  3.215  19.44   1   0
4    Hornet Sportabout  18.7    8  360.0  175  3.15  3.440  17.02   0   0
5              Valiant  18.1    6  225.0  105  2.76  3.460  20.22   1   0
6           Duster 360  14.3    8  360.0  245  3.21  3.570  15.84   0   0
7            Merc 240D  24.4    4  146.7   62  3.69  3.190  20.00   1   0
8             Merc 230  22.8    4  140.8   95  3.92  3.150  22.90   1   0
9             Merc 280  19.2    6  167.6  123  3.92  3.440  18.30   1   0
10            Merc 280C  17.8    6  167.6  123  3.92  3.440  18.90   1   0
11           Merc 450SE  16.4    8  275.8  180  3.07  4.070  17.40   0   0
12           Merc 450SL  17.3    8  275.8  180  3.07  3.730  17.60   0   0
13          Merc 450SLC  15.2    8  275.8  180  3.07  3.780  18.00   0   0
14   Cadillac Fleetwood  10.4    8  472.0  205  2.93  5.250  17.98   0   0
15  Lincoln Continental  10.4    8  460.0  215  3.00  5.424  17.82   0   0
16    Chrysler Imperial  14.7    8  440.0  230  3.23  5.345  17.42   0   0
17             Fiat 128  32.4    4   78.7   66  4.08  2.200  19.47   1   1
18          Honda Civic  30.4    4   75.7   52  4.93  1.615  18.52   1   1
19       Toyota Corolla  33.9    4   71.1   65  4.22  1.835  19.90   1   1
20        Toyota Corona  21.5    4  120.1   97  3.70  2.465  20.01   1   0
21     Dodge Challenger  15.5    8  318.0  150  2.76  3.520  16.87   0   0
22          AMC Javelin  15.2    8  304.0  150  3.15  3.435  17.30   0   0
23           Camaro Z28  13.3    8  350.0  245  3.73  3.840  15.41   0   0
24     Pontiac Firebird  19.2    8  400.0  175  3.08  3.845  17.05   0   0
25            Fiat X1-9  27.3    4   79.0   66  4.08  1.935  18.90   1   1
26        Porsche 914-2  26.0    4  120.3   91  4.43  2.140  16.70   0   1
27         Lotus Europa  30.4    4   95.1  113  3.77  1.513  16.90   1   1
28       Ford Pantera L  15.8    8  351.0  264  4.22  3.170  14.50   0   1
29         Ferrari Dino  19.7    6  145.0  175  3.62  2.770  15.50   0   1
30        Maserati Bora  15.0    8  301.0  335  3.54  3.570  14.60   0   1
31           Volvo 142E  21.4    4  121.0  109  4.11  2.780  18.60   1   1

    gear  carb
0      4     4
1      4     4
2      4     1
3      3     1
4      3     2
5      3     1
6      3     4
7      4     2
8      4     2
9      4     4
```

10	4	4
11	3	3
12	3	3
13	3	3
14	3	4
15	3	4
16	3	4
17	4	1
18	4	2
19	4	1
20	3	1
21	3	2
22	3	2
23	3	4
24	3	2
25	4	1
26	5	2
27	5	2
28	5	4
29	5	6
30	5	8
31	4	2

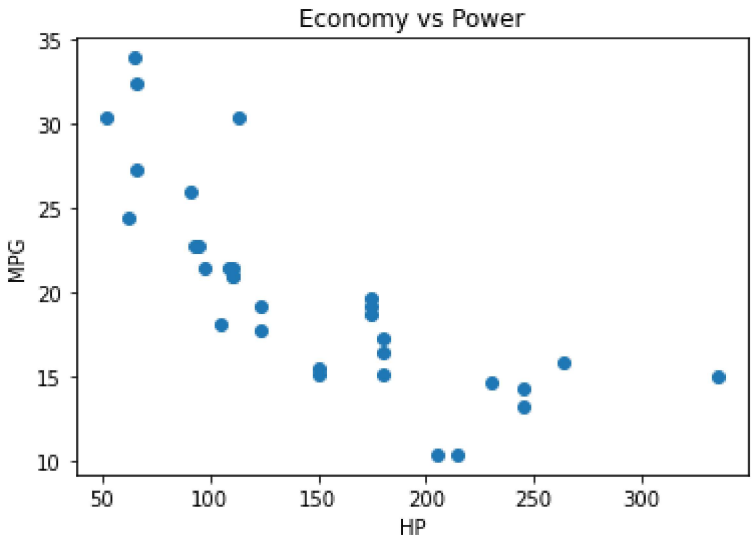
In [3]:
```
#1. Plot am-based histogram to compare mpg (20 points)

#Split am = 1 and am = 0, plot those two sets mpg as a histogram
#Set the alpha to less than 1 to make the histograms transparent
mtcars.groupby("am").mpg.hist(alpha = 0.4)



#Trying to do this with plotly... having a tough time of it
#Need to seperate out the 'am' data but I can't just pass the groupby
#y = mtcars.groupby("am").mpg
#plt.figure()
#plt.hist(y, histtype='barstacked')
```

Out[3]:
```
am
0    AxesSubplot(0.125,0.125;0.775x0.755)
1    AxesSubplot(0.125,0.125;0.775x0.755)
Name: mpg, dtype: object
```



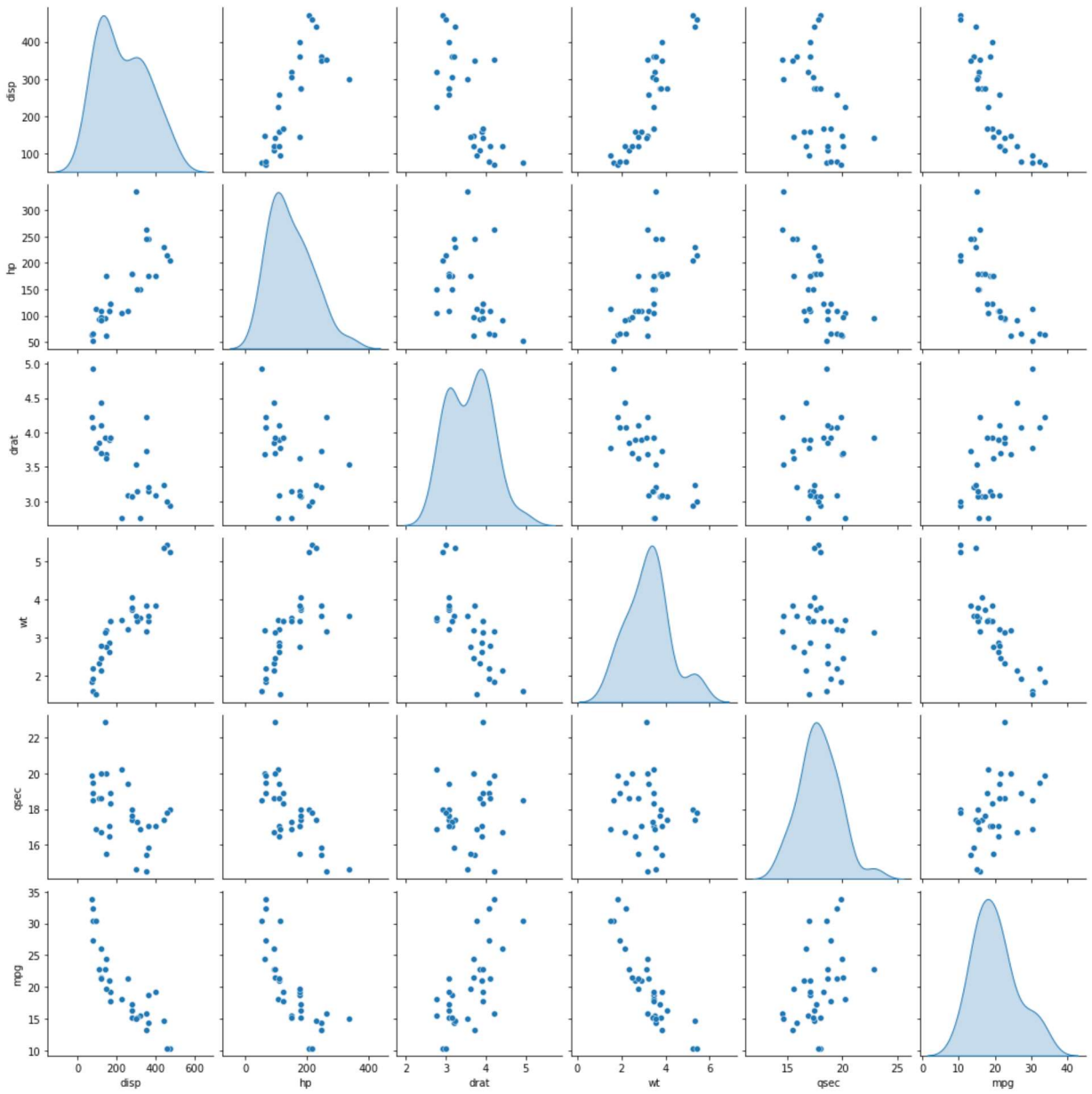
In [4]:
```
#2. Use scatterplot to plot mpg VS. hp (20 points)

plt.figure()
plt.scatter(mtcars.hp, mtcars.mpg)
plt.ylabel("MPG")
plt.xlabel("HP")
plt.title("Economy vs Power")
```

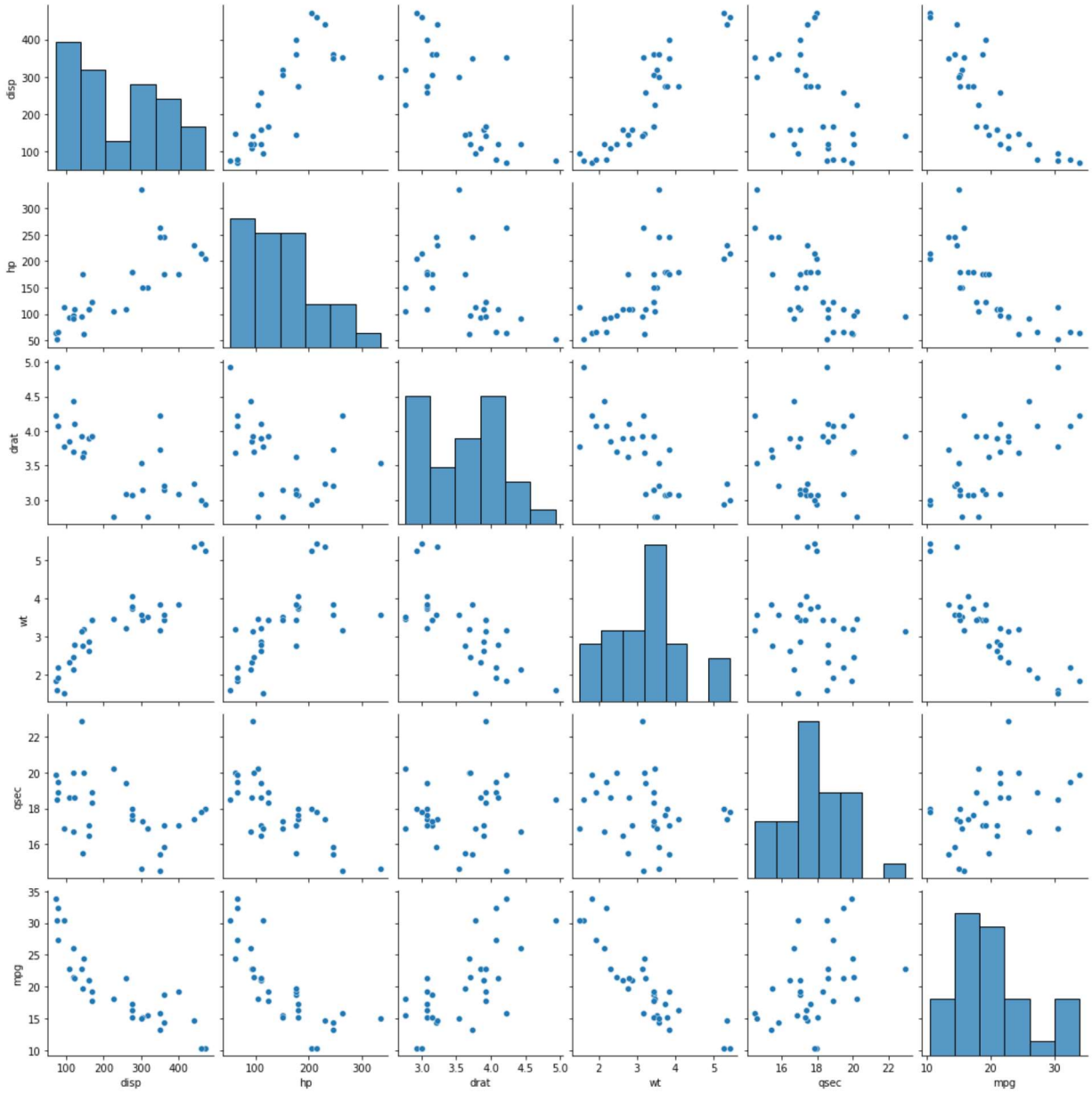
Out[4]:
```
Text(0.5, 1.0, 'Economy vs Power')
```

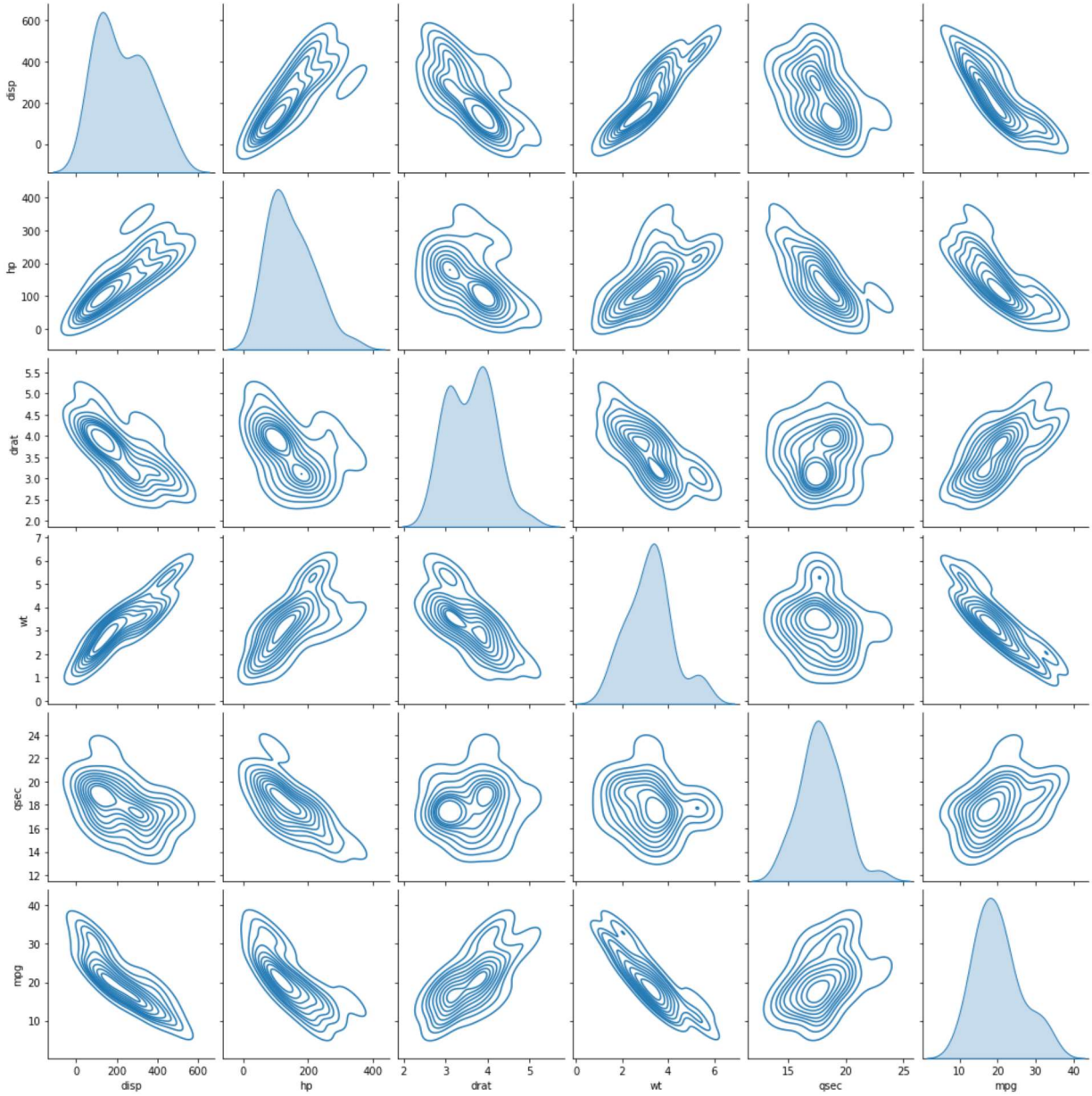


In [5]:
```
#3. Create a scatterplot matrix for new data consisting of columns [disp, hp, dra

#I was playing around with different pairplots here.

#point data with diagonal kde
sns.pairplot(mtcars, vars = ['disp', 'hp', 'drat', 'wt', 'qsec', 'mpg'], diag_kin

#point data only
sns.pairplot(mtcars, vars = ['disp', 'hp', 'drat', 'wt', 'qsec', 'mpg'])

#All data kde (contors)
sns.pairplot(mtcars, vars = ['disp', 'hp', 'drat', 'wt', 'qsec', 'mpg'], kind='kd

#ugly plot and not useful
#sns.pairplot(mtcars, vars = ['disp', 'hp', 'drat', 'wt', 'qsec', 'mpg'], kind='h
```

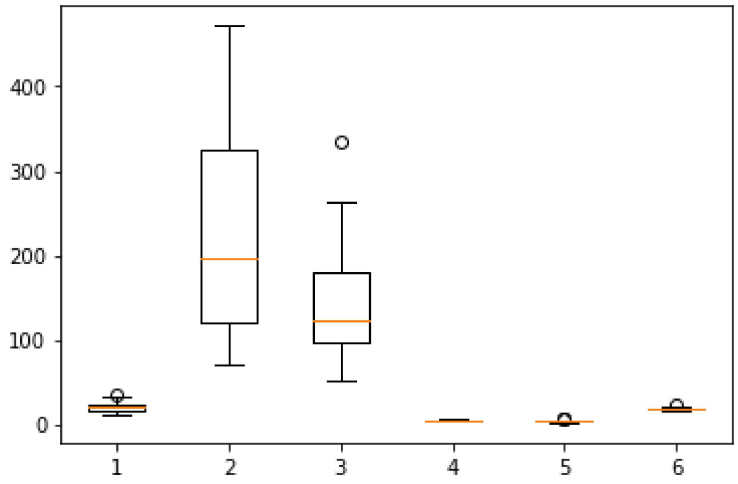
Out[5]:
```
<seaborn.axisgrid.PairGrid at 0x1cc5bd5f850>
```

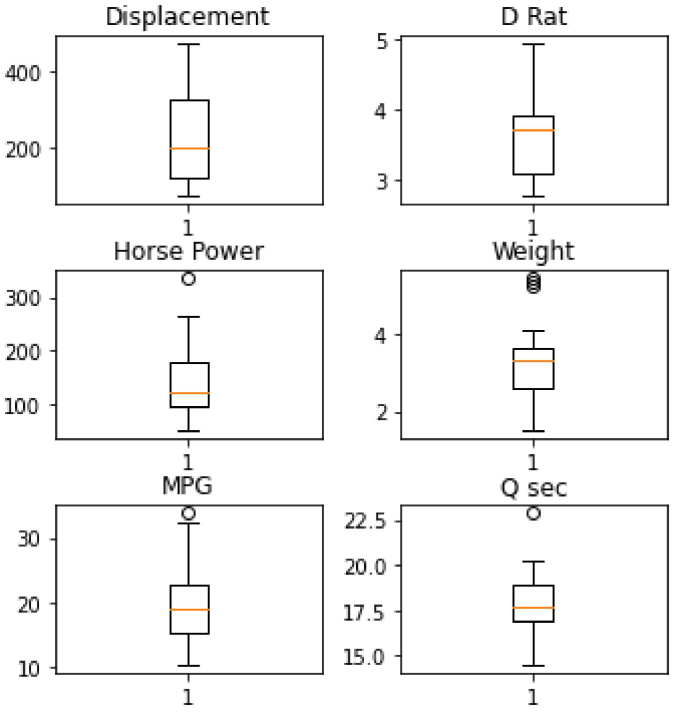



disp
hp
drat
wt
qsec
mpg


disp
hp
drat
wt
qsec
mpg

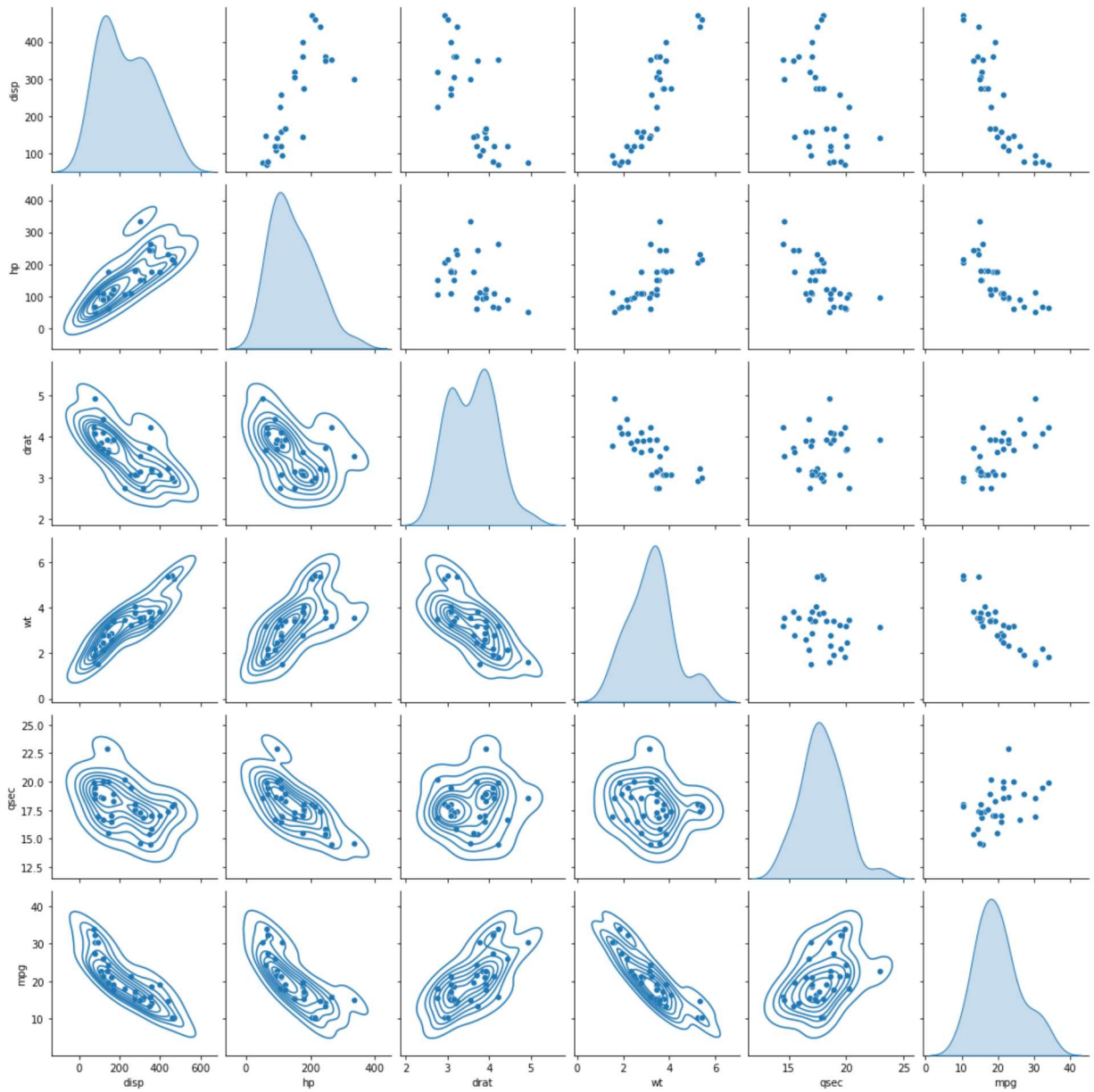
In [6]:
```
#4. Create boxplots for new data consisting of columns [disp, hp, drat, wt, qsect

#get just the relevant columns
test = mtcars[['mpg','disp', 'hp', 'drat', 'wt', 'qsec']]

#One boxplot with all of the column values through it
plt.figure()
plt.boxplot(test)


#get just the relevant columns
test = mtcars[['mpg','disp', 'hp', 'drat', 'wt', 'qsec']]
#Plot six individual boxplots as the scaling is too wide on the previous
fig, axs = plt.subplots(3, 2, figsize=(5, 5))
fig.tight_layout(w_pad = 1)
axs[0, 0].boxplot(test.disp)
axs[0, 0].set_title('Displacement')
axs[1, 0].boxplot(test.hp)
axs[1, 0].set_title("Horse Power")
axs[0, 1].boxplot(test.drat)
axs[0, 1].set_title("D Rat")
axs[1, 1].boxplot(test.wt)
axs[1, 1].set_title("Weight")
axs[2, 1].boxplot(test.qsec)
axs[2, 1].set_title("Q sec")
axs[2, 0].boxplot(test.mpg)
axs[2, 0].set_title("MPG")
plt.subplots_adjust(wspace = 0.3, hspace = 0.4)
```




Displacement
400
200
1
D Rat
5
4
3
1
Horse Power
300
200
100
1
Weight
4
2
1
MPG
30
20
10
1
Q sec
22.5
20.0
17.5
15.0
1

In [7]:
```
#5. Use plots to answer which variable has the most impact on mpg. (20 points)

#Plot a pairplot of the 6 variables with diagonals being kde
g = sns.pairplot(mtcars, vars = ['disp', 'hp', 'drat', 'wt', 'qsec', 'mpg'], diag
#set the lower grids to also have kde contors
g.map_lower(sns.kdeplot, levels = 8)

#Based on the graphs, we can see that the 'wt' variable
#has the tightest contors with the 'mpg' variable
#Thus, there should be the highest correlation between them.
print("""\n\nBased on the graphs, we can see that the 'wt' variable has the tight
      Thus, there should be the highest correlation between them.""")
```

```
Based on the graphs, we can see that the 'wt' variable has the tightest contors
with the 'mpg' variable.

      Thus, there should be the highest correlation between them.
```



In [ ]:

In [ ]: