In [7]:
```
import matplotlib.pyplot as plt
import pandas as pd
import os
%matplotlib widget
%matplotlib inline
%matplotlib notebook

os.chdir("C:\\Users\\micha\\Documents\\DAAN862\\")

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

```
Warning: Cannot change to a different GUI toolkit: notebook. Using widget instead.
```

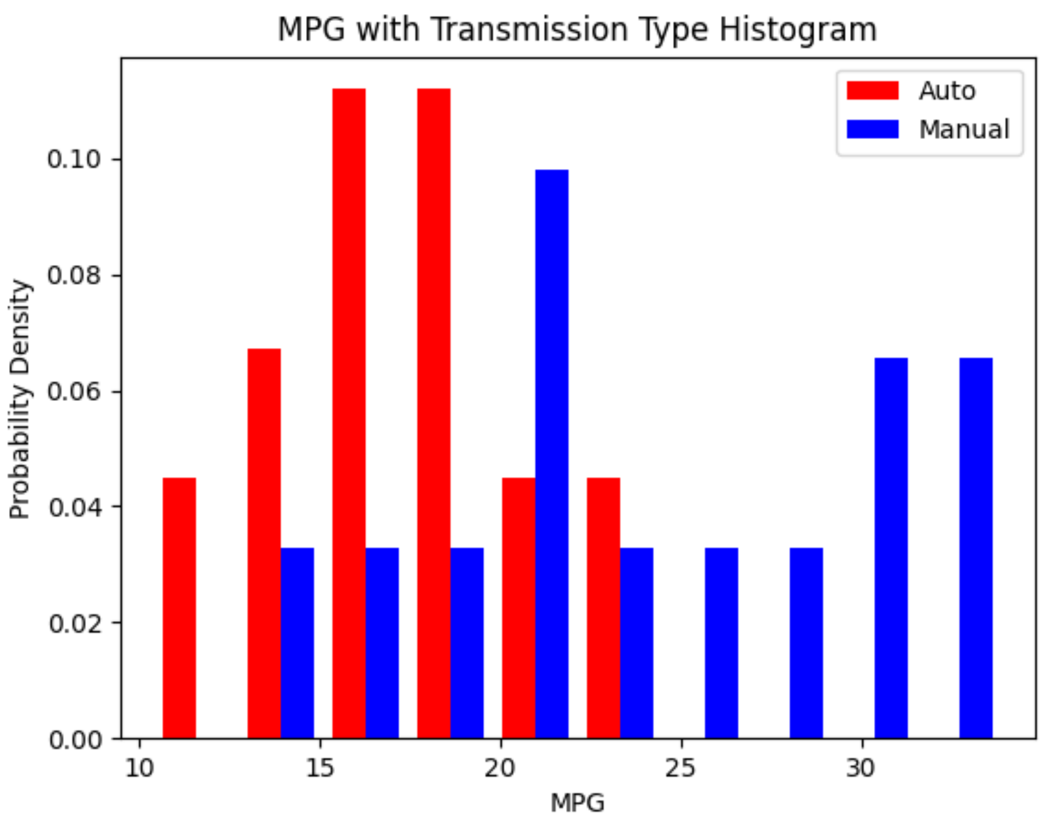
	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
1	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
2	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
3	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
4	Hornet Sportabout	18.7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

	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
30	Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8
31	Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

In [10]:
```
# Question 1 - Plot a am-based Histogram
# Split cars into two types based on 'am'
cars_split = cars.groupby(['am'])

# Plot the two groups in a histogram for mpg
plt.figure()
plt.hist([cars_split.get_group(0).mpg, cars_split.get_group(1).mpg], 10, density=1,
         color=['r', 'b'])
plt.xlabel('MPG')
plt.ylabel('Probability Density')
plt.legend(loc='upper right')
plt.title('MPG with Transmission Type Histogram')
```

Out[10]: `Text(0.5, 1.0, 'MPG with Transmission Type Histogram')`

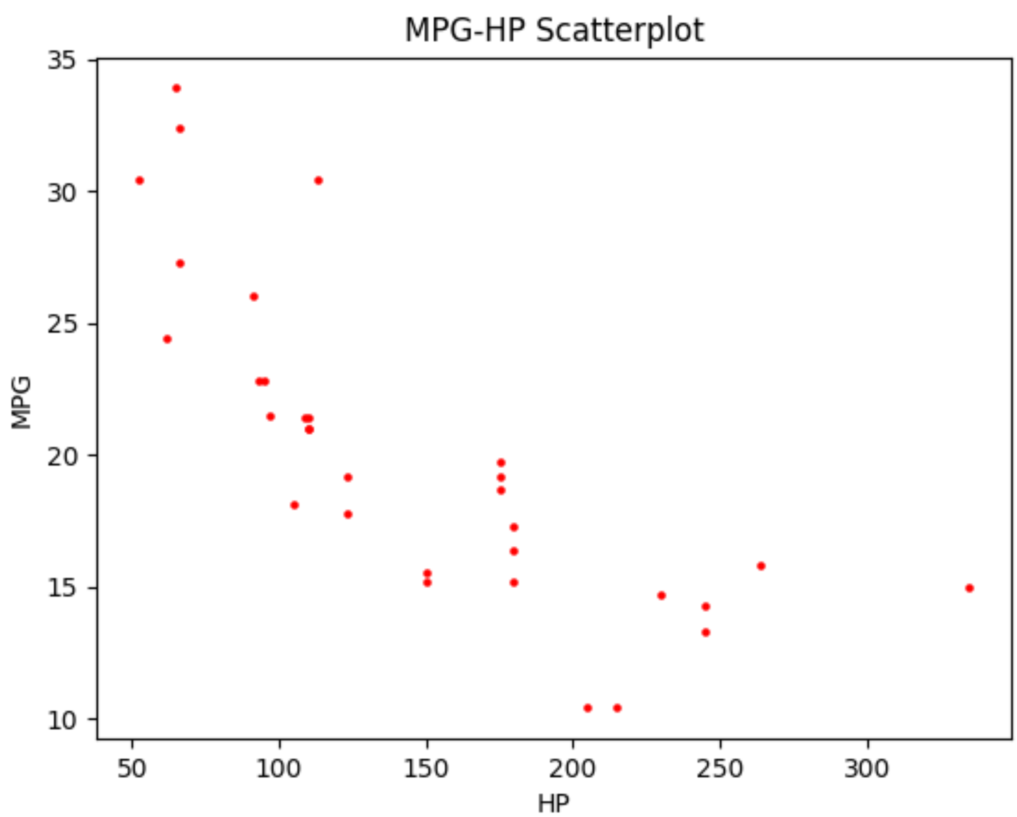
Figure



In [3]:
```
# Question 2 - Plot MPG vs HP on Scatterplot

plt.figure()
plt.scatter(cars.hp, cars.mpg, s=5, c='r')
plt.xlabel('HP')
plt.ylabel('MPG')
plt.title('MPG-HP Scatterplot')
```

Out[3]:
```
Text(0.5, 1.0, 'MPG-HP Scatterplot')
```

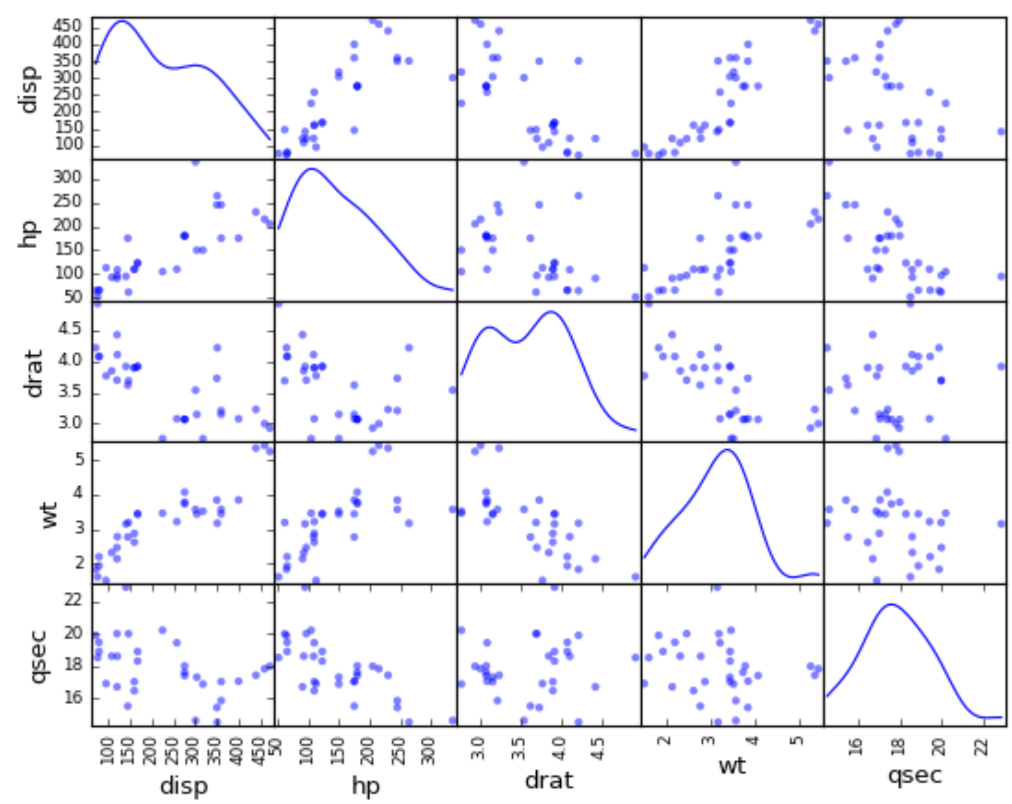
Figure



In [12]:
```
# Question 3 - Create a Scatter Matrix with disp, hp, drat, wt, and qsec
cars_scatter = cars[['disp', 'hp', 'drat', 'wt', 'qsec']].copy()

plt.style.use('classic')

pd.plotting.scatter_matrix(cars_scatter, c='b', s=60, diagonal='kde')
```

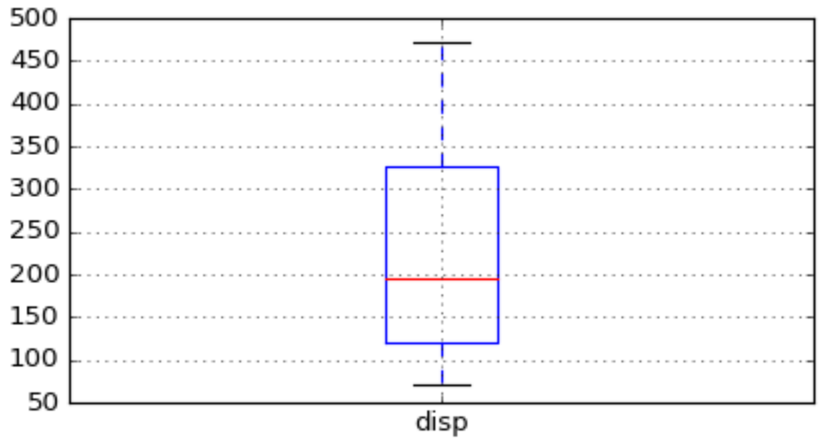
Out[12]:
```
array([[<AxesSubplot:xlabel='disp', ylabel='disp'>,
        <AxesSubplot:xlabel='hp', ylabel='disp'>,
        <AxesSubplot:xlabel='drat', ylabel='disp'>,
        <AxesSubplot:xlabel='wt', ylabel='disp'>,
        <AxesSubplot:xlabel='qsec', ylabel='disp'>],
       [<AxesSubplot:xlabel='disp', ylabel='hp'>,
        <AxesSubplot:xlabel='hp', ylabel='hp'>,
        <AxesSubplot:xlabel='drat', ylabel='hp'>,
        <AxesSubplot:xlabel='wt', ylabel='hp'>,
        <AxesSubplot:xlabel='qsec', ylabel='hp'>],
       [<AxesSubplot:xlabel='disp', ylabel='drat'>,
        <AxesSubplot:xlabel='hp', ylabel='drat'>,
        <AxesSubplot:xlabel='drat', ylabel='drat'>,
        <AxesSubplot:xlabel='wt', ylabel='drat'>,
        <AxesSubplot:xlabel='qsec', ylabel='drat'>],
       [<AxesSubplot:xlabel='disp', ylabel='wt'>,
        <AxesSubplot:xlabel='hp', ylabel='wt'>,
        <AxesSubplot:xlabel='drat', ylabel='wt'>,
        <AxesSubplot:xlabel='wt', ylabel='wt'>,
        <AxesSubplot:xlabel='qsec', ylabel='wt'>],
       [<AxesSubplot:xlabel='disp', ylabel='qsec'>,
        <AxesSubplot:xlabel='hp', ylabel='qsec'>,
        <AxesSubplot:xlabel='drat', ylabel='qsec'>,
        <AxesSubplot:xlabel='wt', ylabel='qsec'>,
        <AxesSubplot:xlabel='qsec', ylabel='qsec'>]], dtype=object)
```

Figure

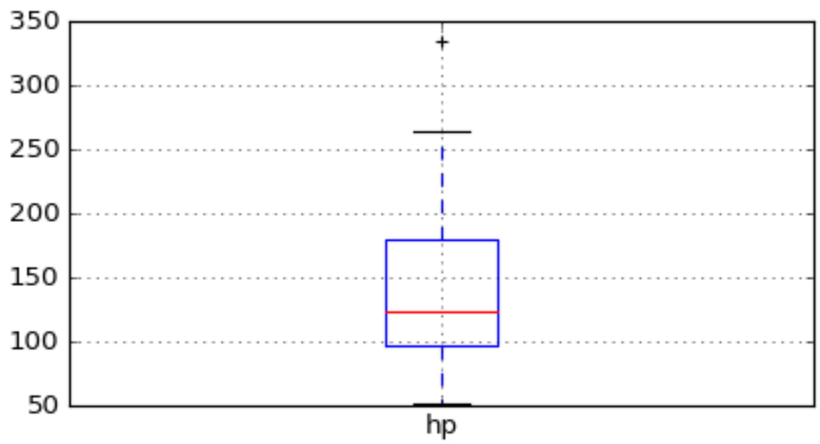


In [47]:
```
# Question 4 - Plot the same data in Boxplots

for col in cars_scatter:
    plt.figure(figsize=(6,3))
    cars_scatter.boxplot([col])
```

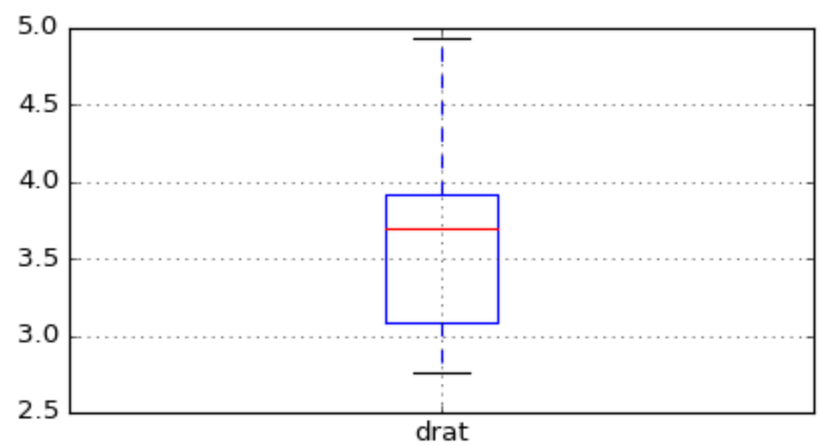
Figure



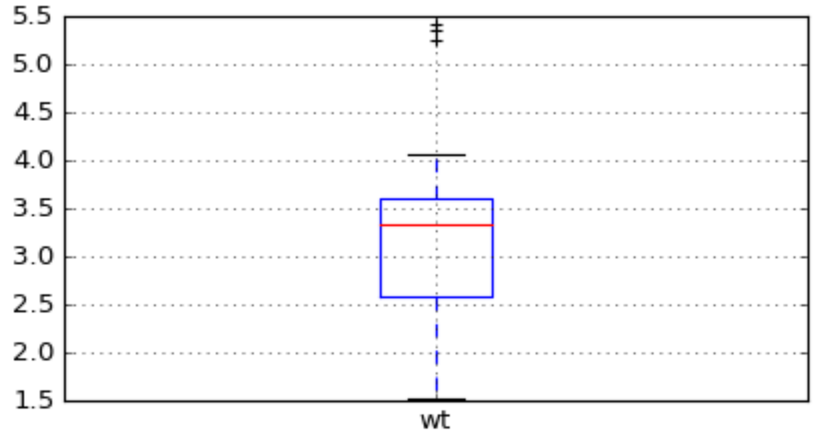
Figure



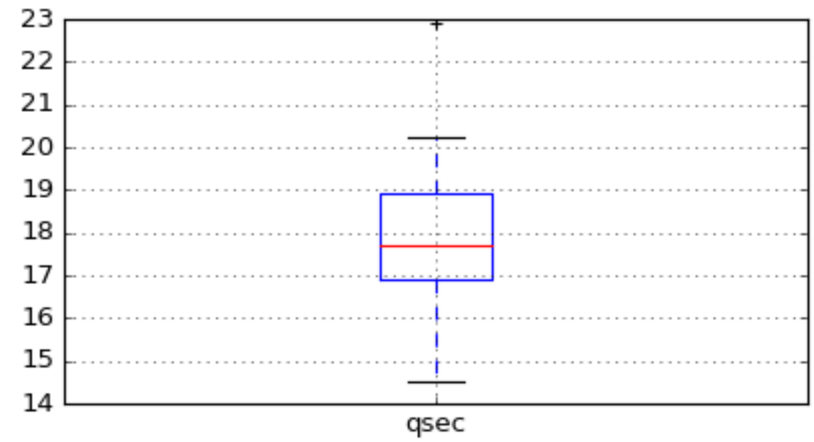
Figure



Figure

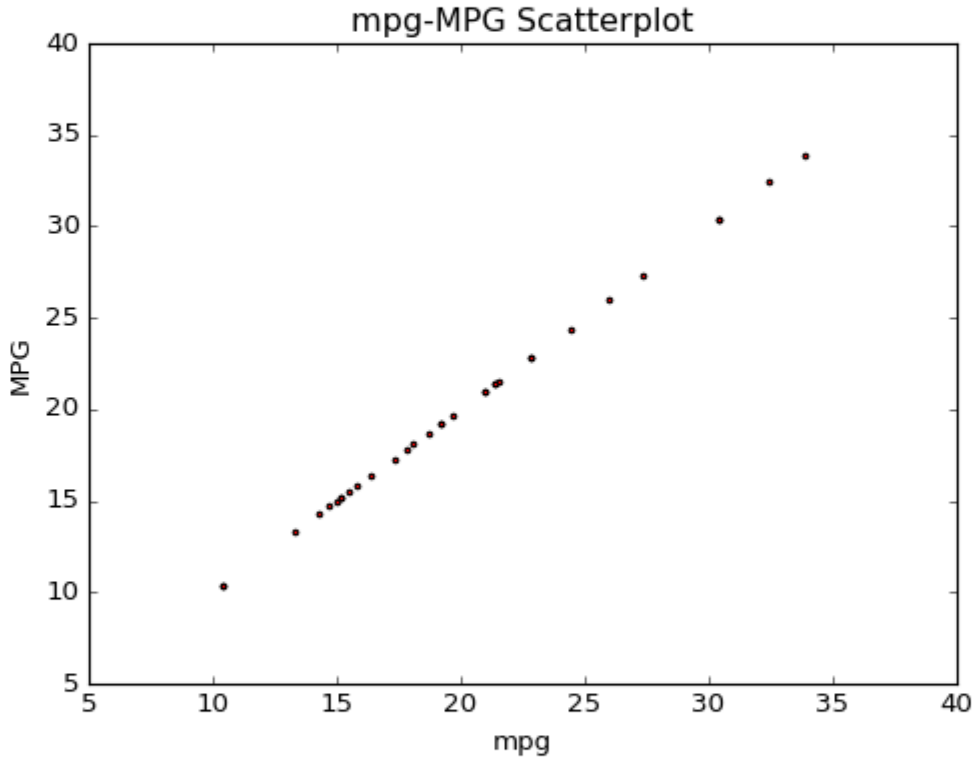


Figure

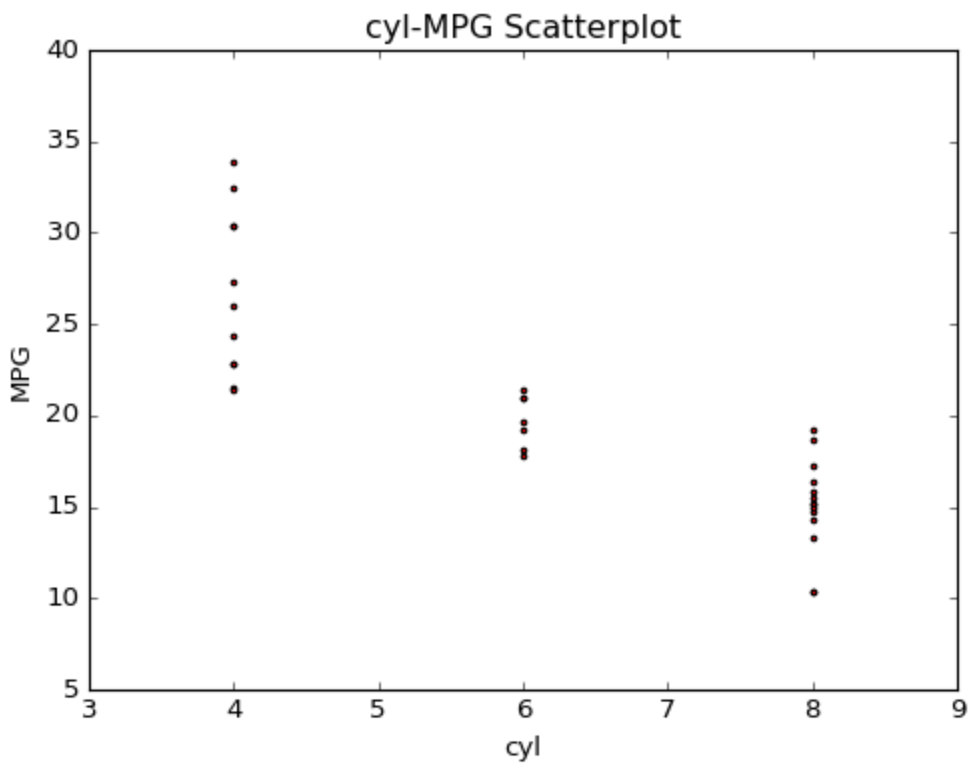


In [82]:
```
# Question 5 - Utilize plots to determine which variable has the largest impact on 

# First, let us create a scatter matrix with all the variables besides model

# Start with producing a correlation heatmap by getting all variables
# correlated agsint mpg
cars_dropped = cars.copy()
cars_dropped.pop('model')

for col in cars_dropped:
    plt.figure(figsize=(7, 5))
    plt.scatter(cars[col], cars.mpg, s=5, c='r')
    plt.xlabel(col)
    plt.ylabel('MPG')
    plt.title(col + '-MPG Scatterplot')

# As shown from the produced scatter plots, there are a few variables that seem
# to strong correlate to MPG, such as disp, hp, drat, and wt
# From here, we can further analyze the data by using heatmaps of correlation
cars_corr = cars_dropped.corrwith(cars.mpg)
```

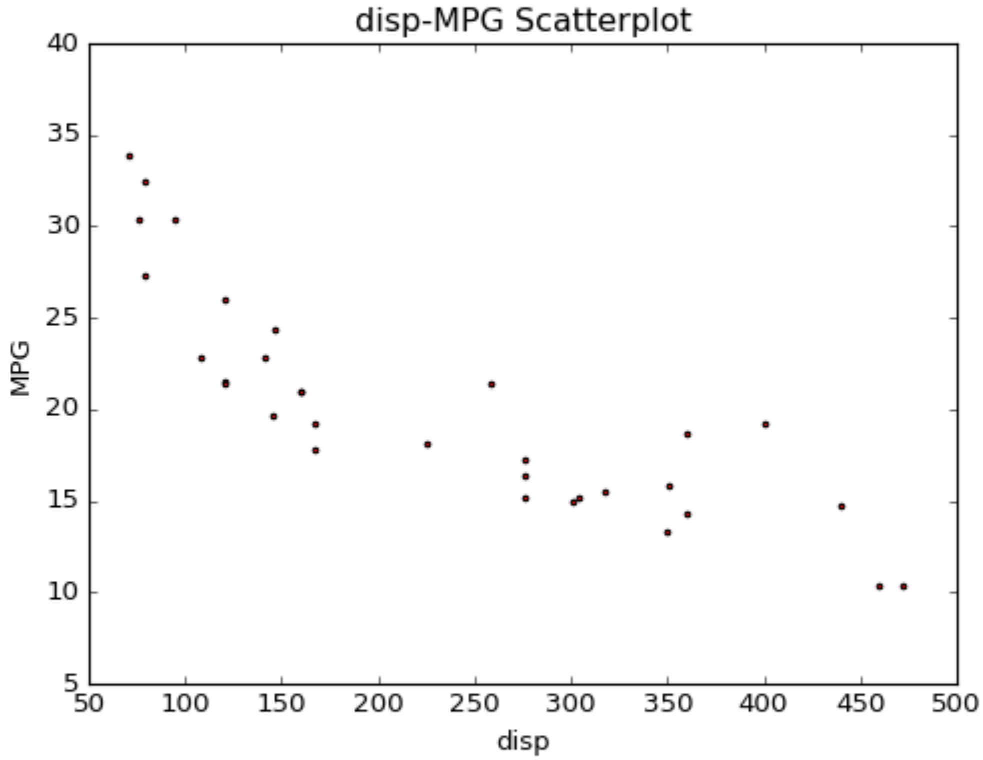
Figure



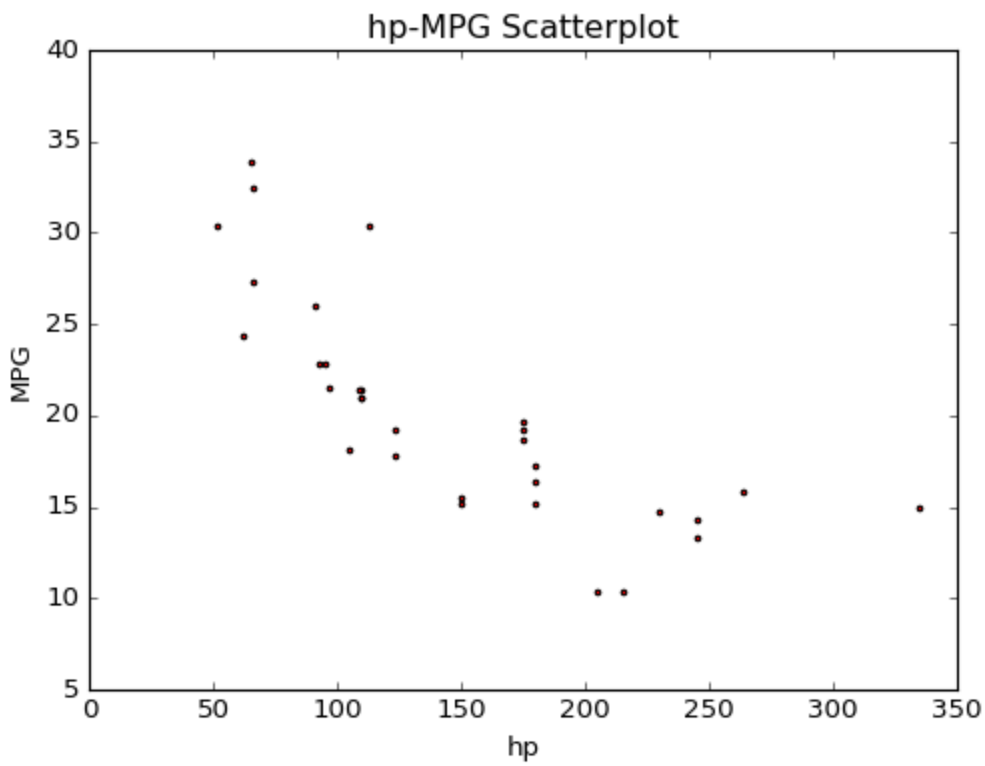
Figure



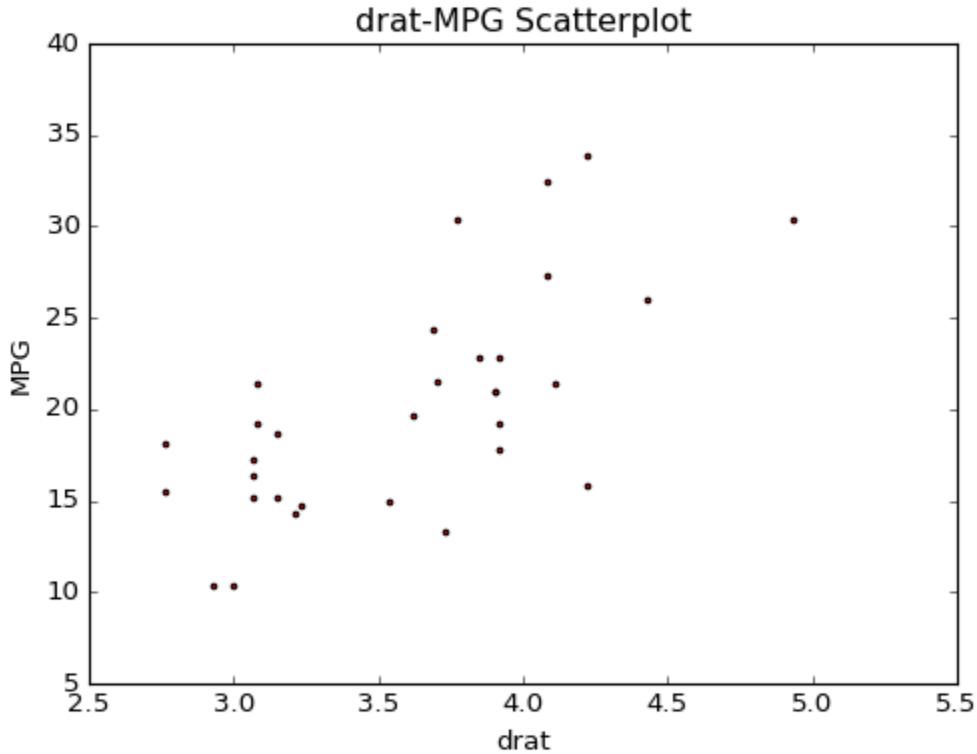
Figure



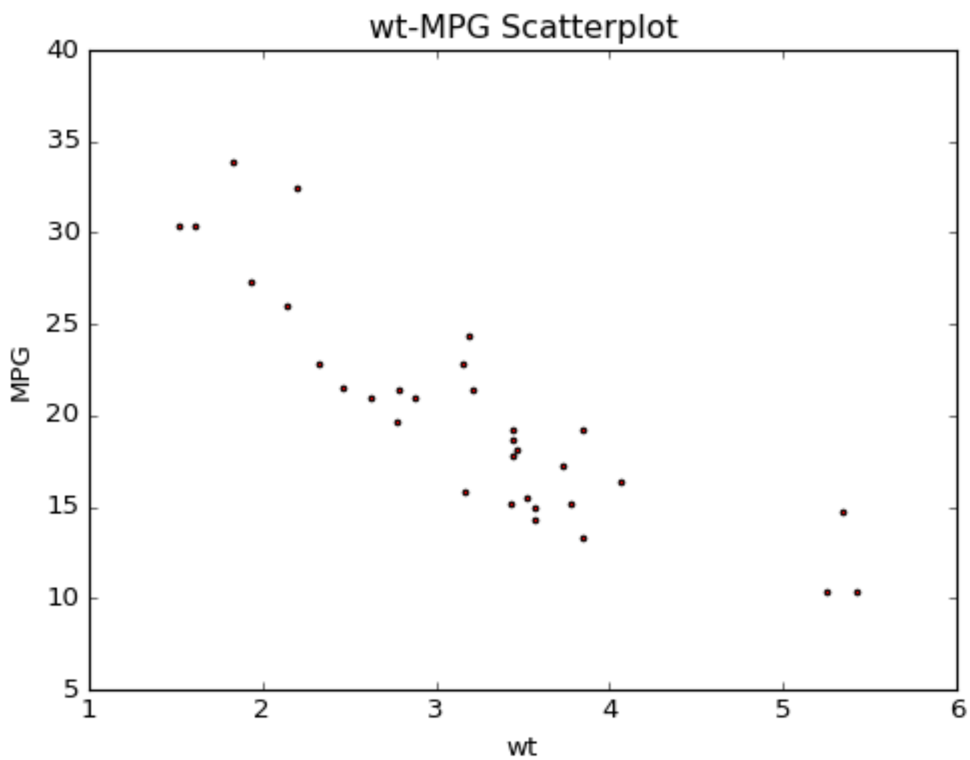
Figure



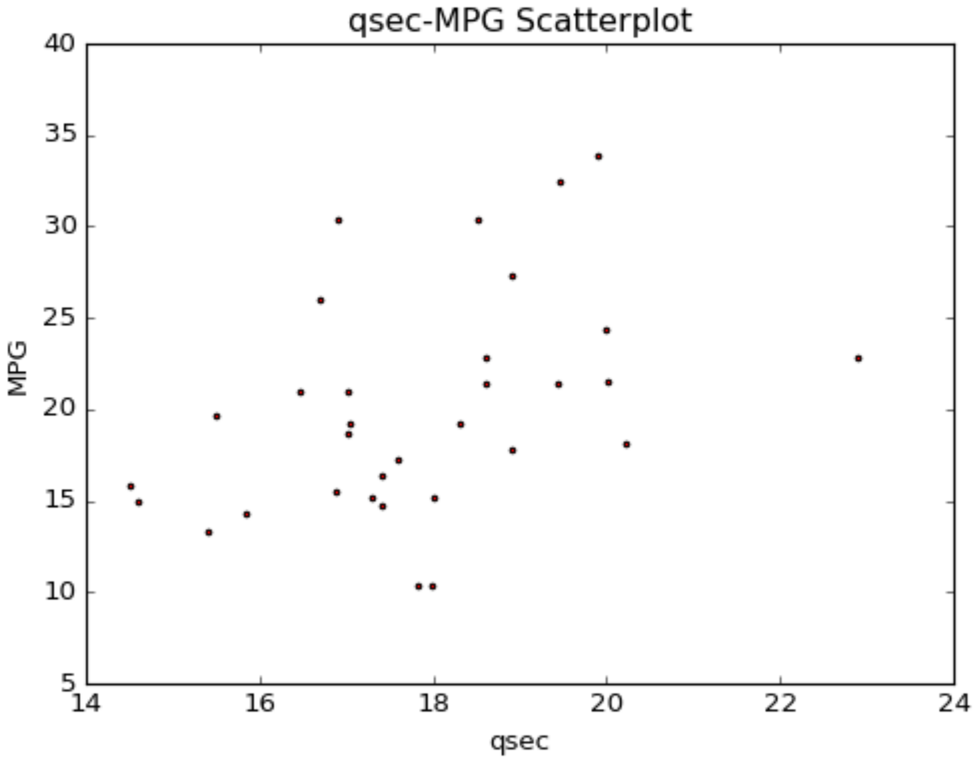
Figure



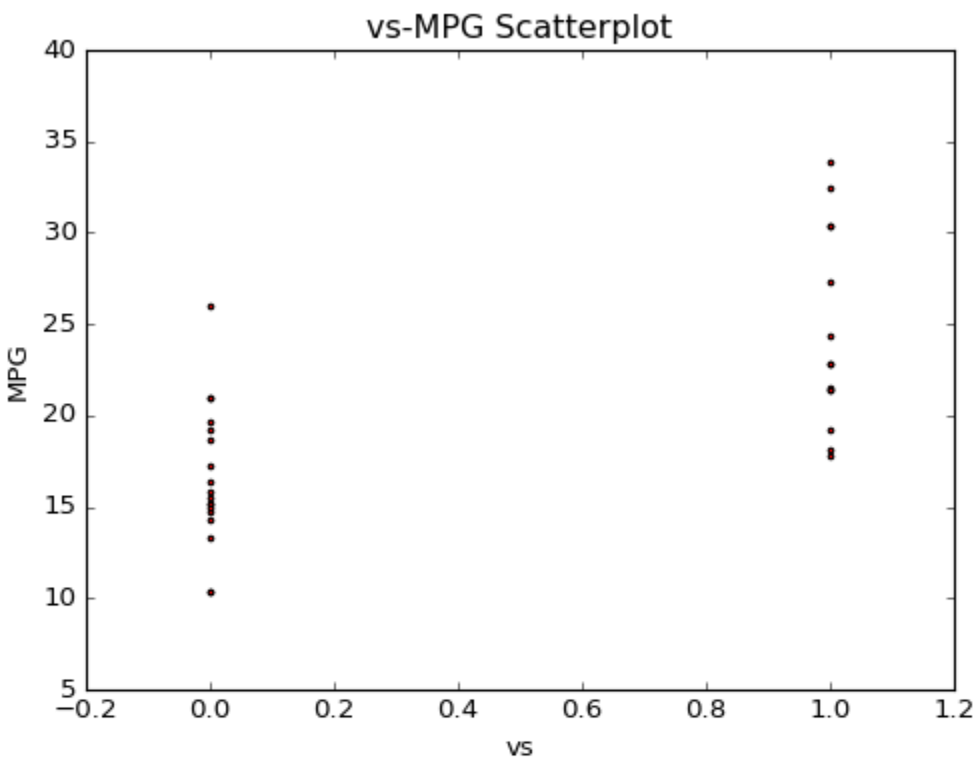
Figure



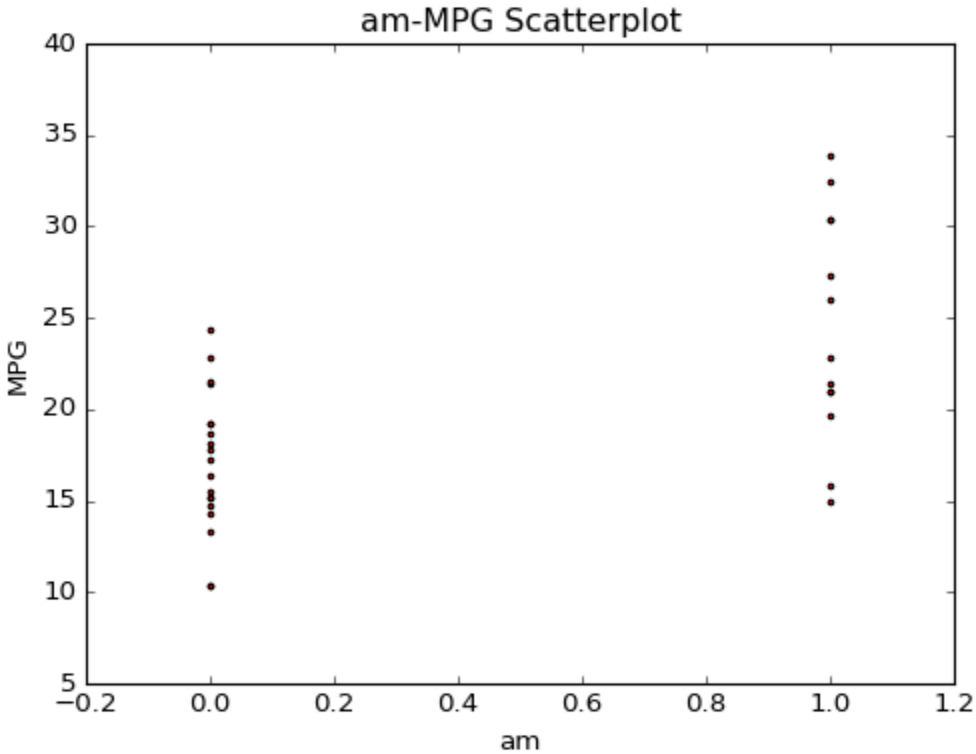
Figure



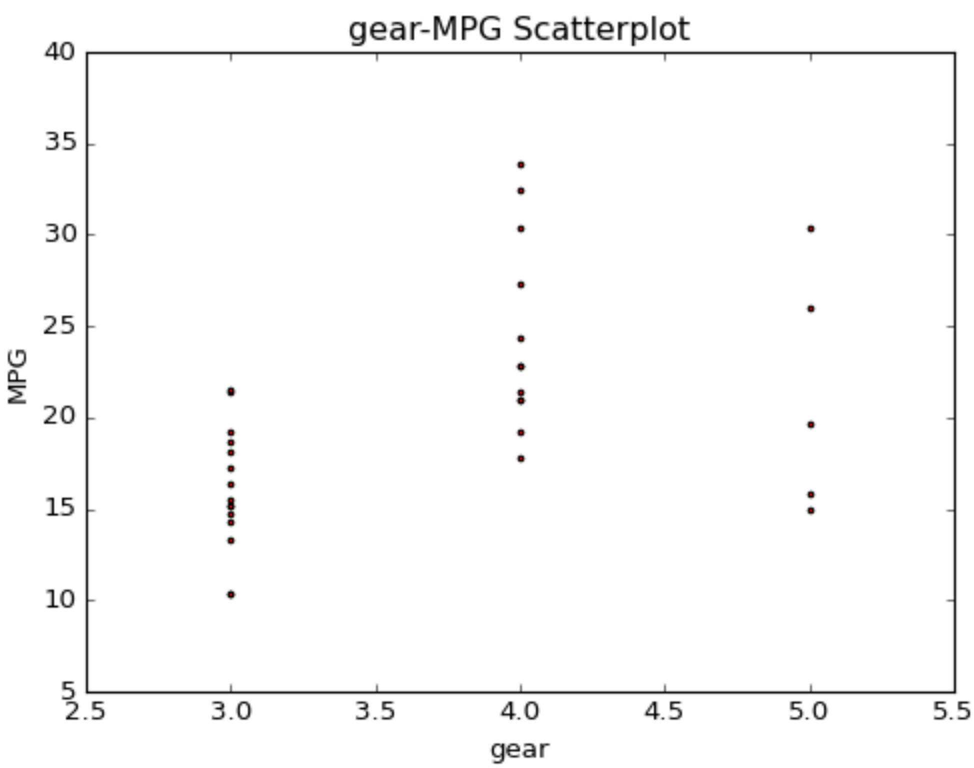
Figure



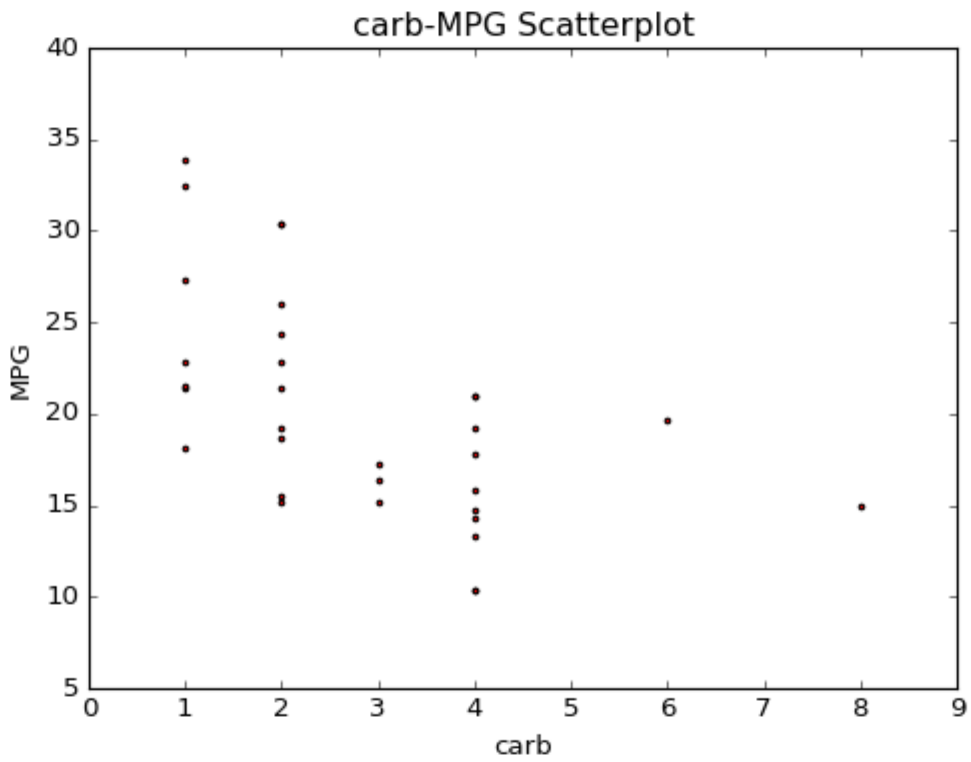
Figure



Figure



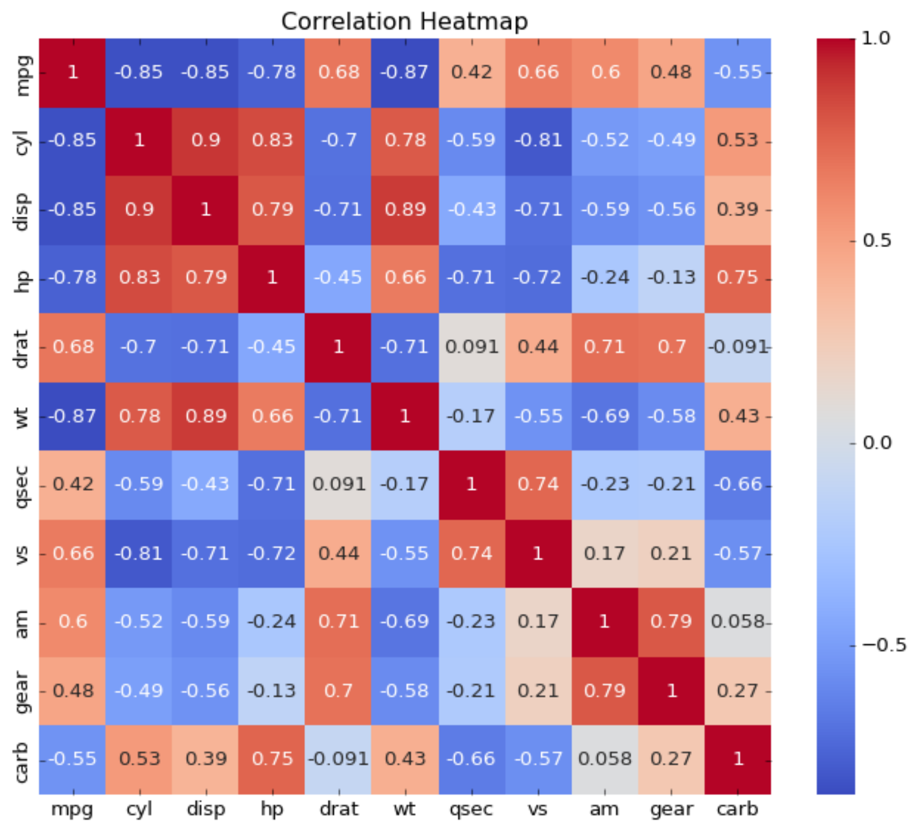
Figure



In [84]:

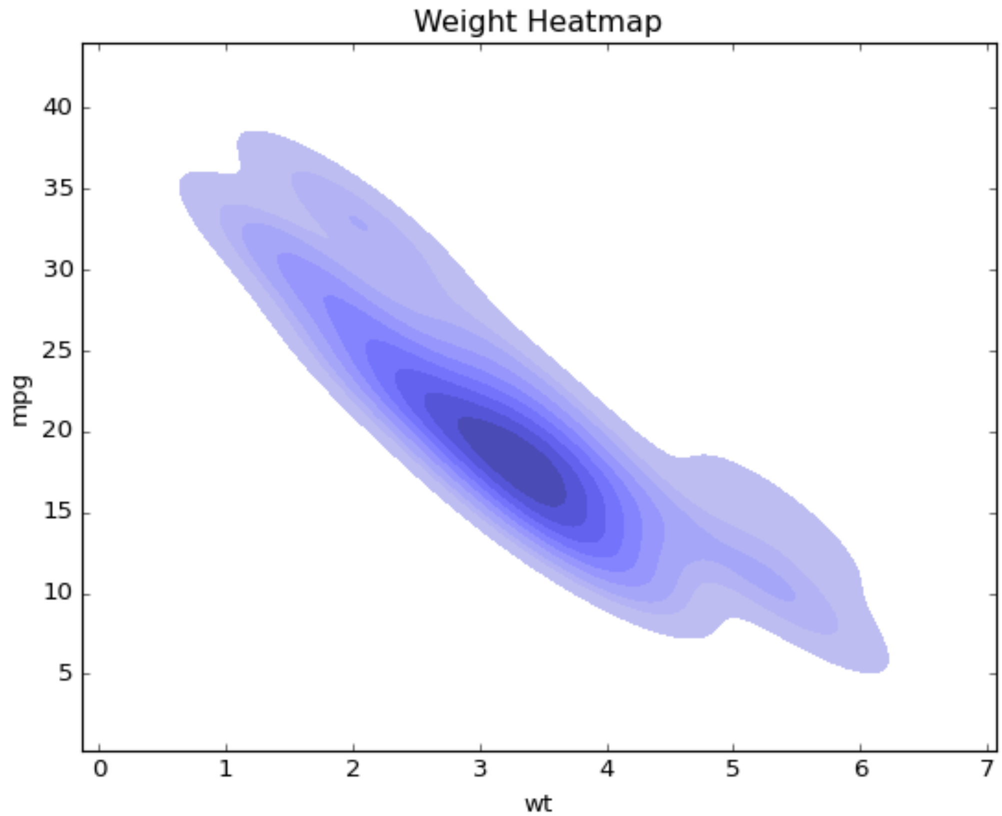
```
# Using seaborn package, we can create a correlation heatmap now
import seaborn as sns

corr_heat_map = cars_dropped.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(corr_heat_map, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()

# We can see that our initial observation may be true as wt and disp
# as well as cyl seem to correlate the strongest to MPG. We can replot these
# values again to verify with individual heatmaps
```

Figure

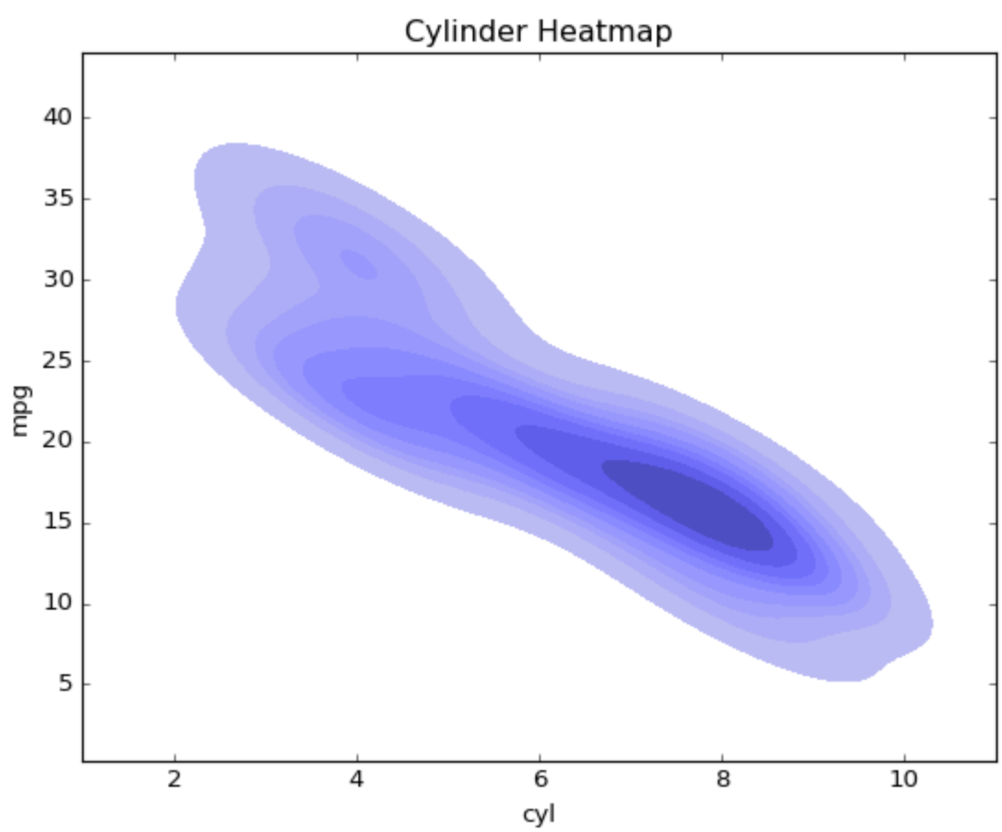

Correlation Heatmap
mpg
cyl
disp
hp
drat
wt
qsec
vs
am
gear
carb
1.0
0.5
0.0
−0.5

In [92]:
```
# Top association plots
plt.figure()
sns.kdeplot(x=cars_dropped.wt, y=cars_dropped.mpg, fill=True)
plt.title('Weight Heatmap')
plt.show()

plt.figure()
sns.kdeplot(x=cars_dropped.cyl, y=cars_dropped.mpg, fill=True)
plt.title('Cylinder Heatmap')
plt.show()

plt.figure()
sns.kdeplot(x=cars_dropped.disp, y=cars_dropped.mpg, fill=True)
plt.title('Displacement Heatmap')
plt.show()

# In contrast, poor association
plt.figure()
sns.kdeplot(x=cars_dropped.qsec, y=cars_dropped.mpg, fill=True)
plt.title('Quarter Sec Heatmap')
plt.show()

# A more narrow, concentration of values with deeper color correspond to a
# a stronger correlation

# Without mathematical analysis, it is difficult to determine with 100%
# accuracy which variable impacts another the most, but by using scatter plots
# we were able to identify several variables that are likely to be the strongest as
# and then by utilizing mathematical means, we can verify our observations to deter
# Weight has the biggest impact followed by Displacement and Cylinder count
```

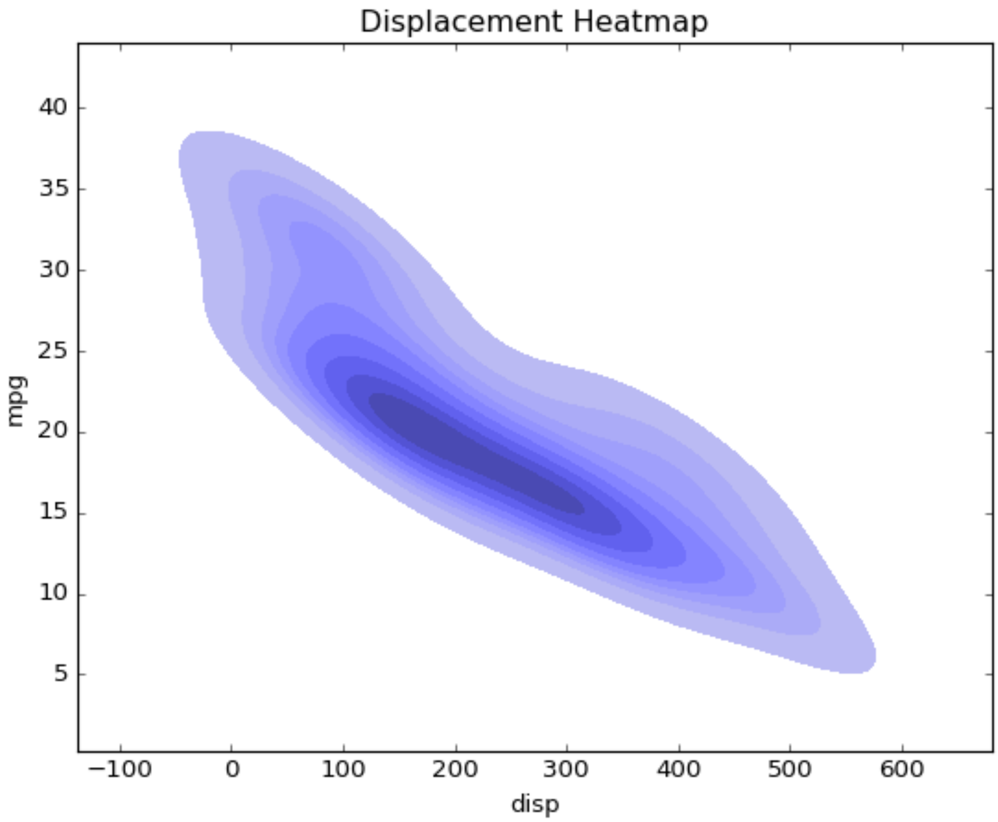
Figure



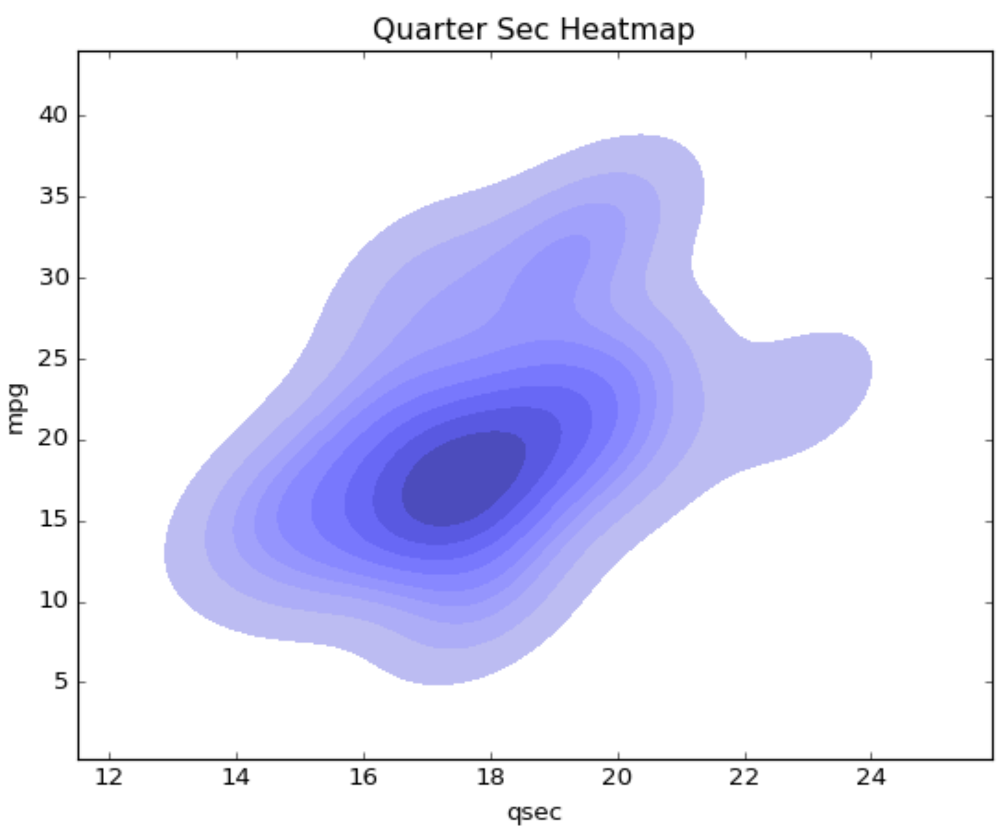
Figure



Figure



Figure



In [ ]:

```

```