Chapter4

March 1, 2022

1 Chapter 4 - Practical Data Visualization

1.1 Segment 1 - Creating standard data graphics

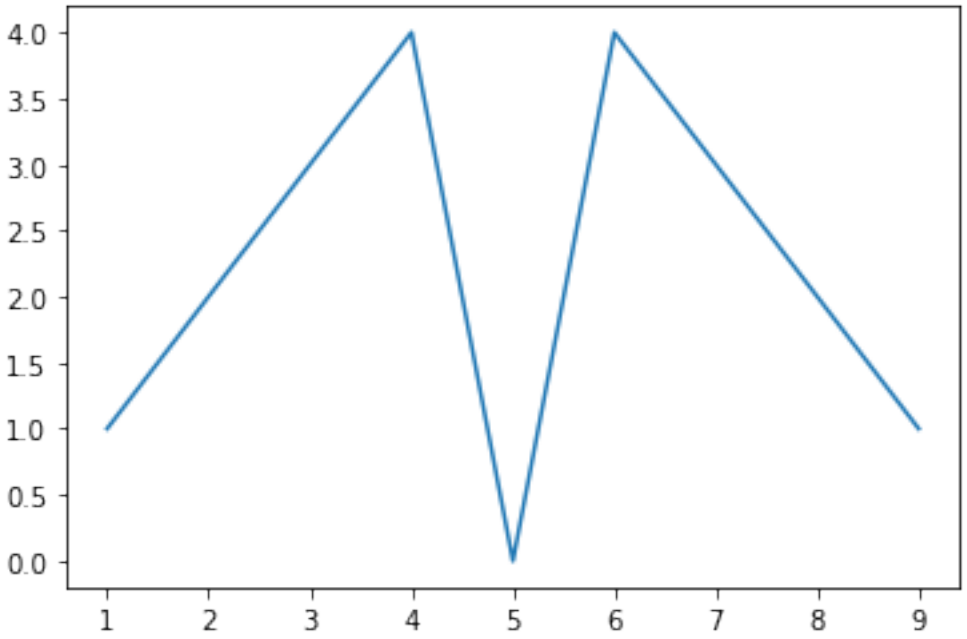
[1]:
```
import numpy as np
from numpy.random import randn
import pandas as pd
from pandas import Series, DataFrame

import matplotlib.pyplot as plt
from matplotlib import rcParams
```

1.1.1 Creating a line chart from a list object

Plotting a line chart in matplotlib

[2]:
```
x = range(1,10)
y = [1,2,3,4,0,4,3,2,1]

plt.plot(x,y)
```

[2]:
```
[<matplotlib.lines.Line2D at 0x26093e64880>]
```

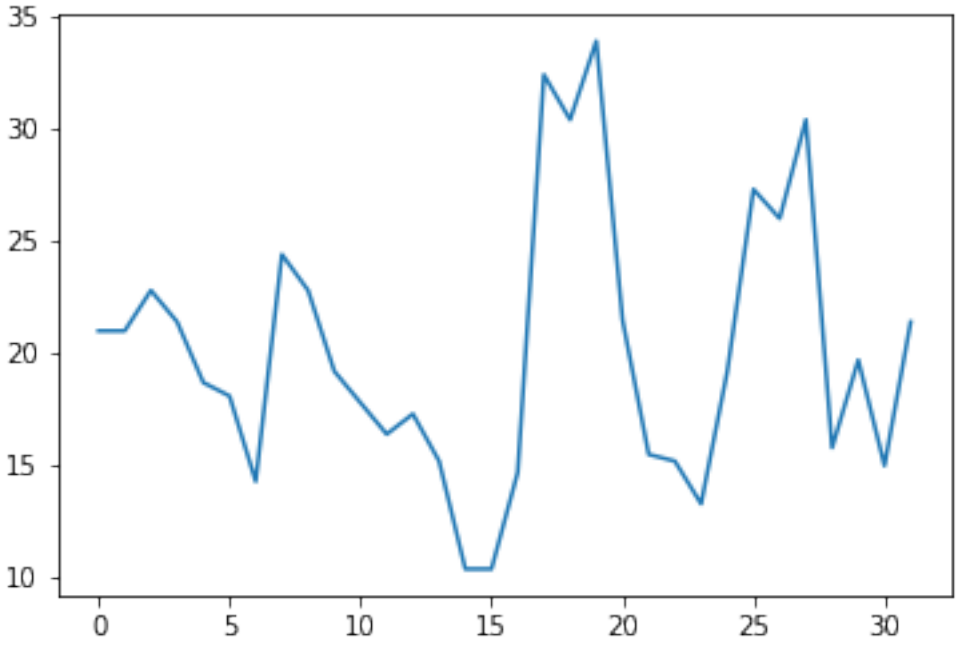


Plotting a line chart from a Pandas object

[3]:
```
address = './Data/mtcars.csv'

cars = pd.read_csv(address)
cars.columns = ['car_names','mpg','cyl','disp', 'hp', 'drat', 'wt', 'qsec',␣
↪'vs', 'am', 'gear', 'carb']

mpg = cars['mpg']
```

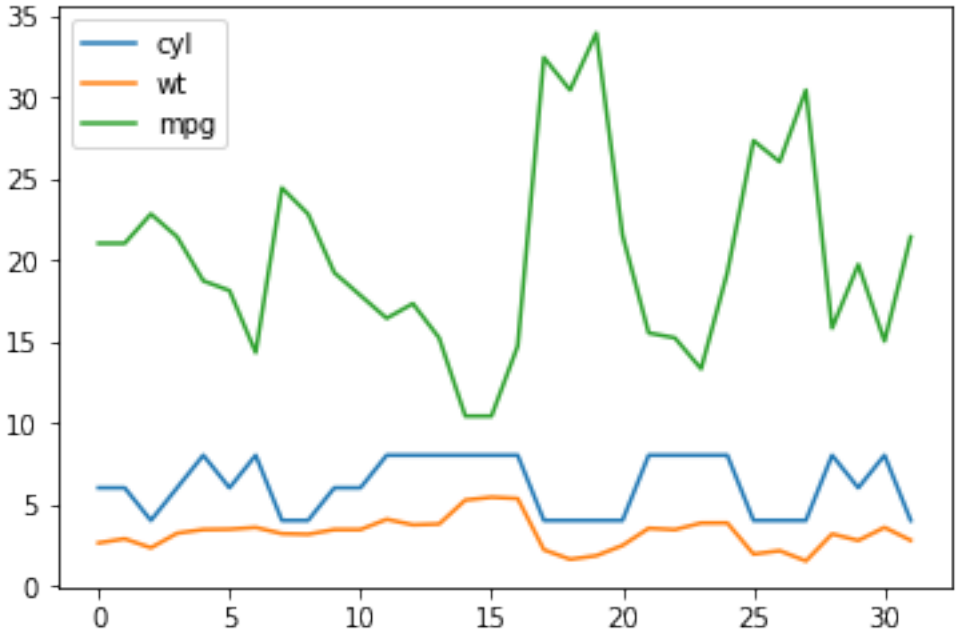
[4]:
```
mpg.plot()
```

[4]: <AxesSubplot:>



[5]:
```
df = cars[['cyl','wt','mpg']]
df.plot()
```

[5]: <AxesSubplot:>

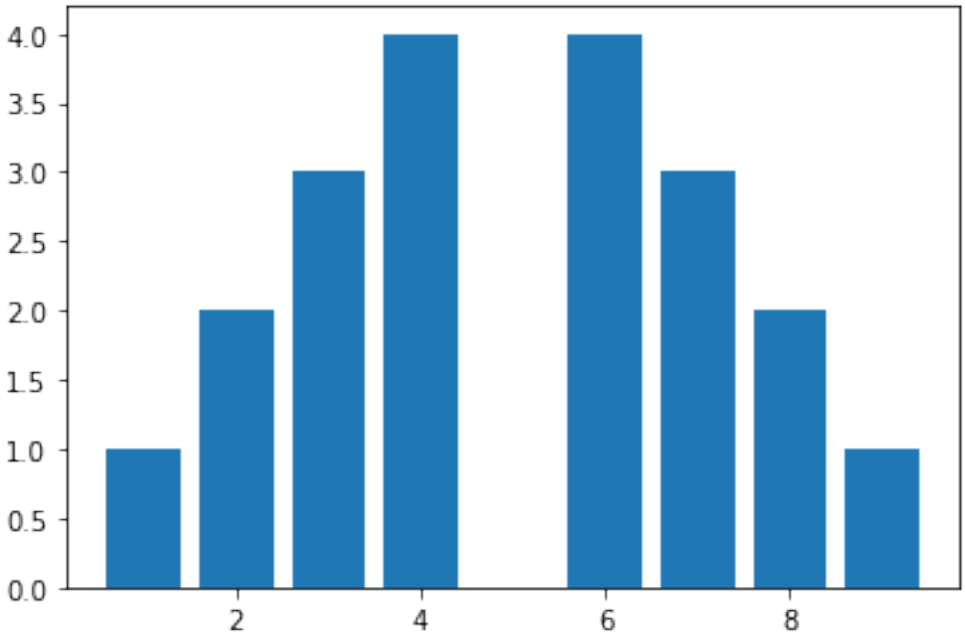


1.1.2 Creating bar charts

Creating a bar chart from a list

[6]:
```
plt.bar(x, y)
```

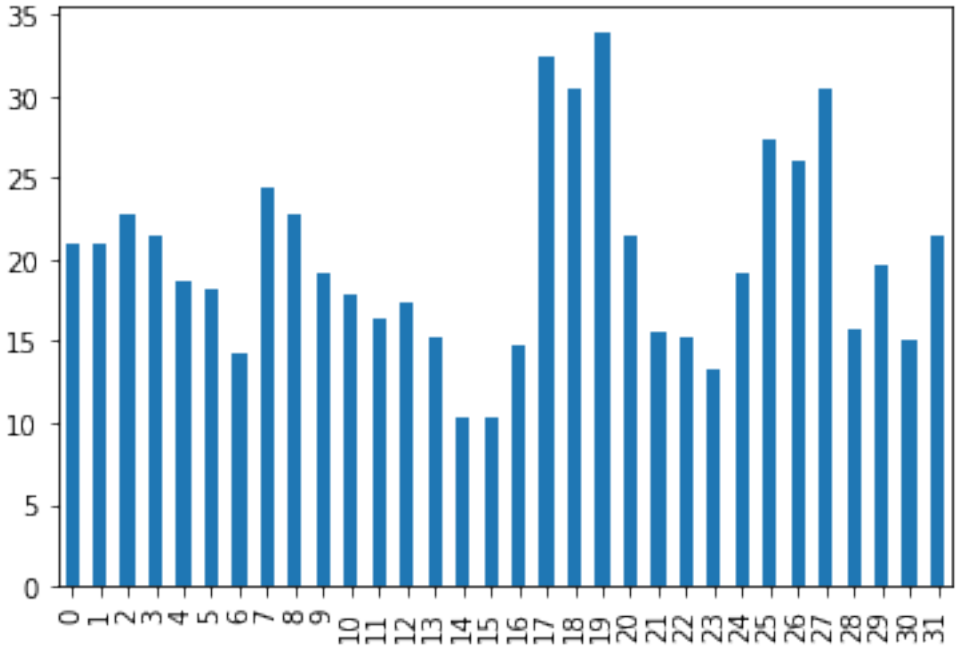
[6]: <BarContainer object of 9 artists>



Creating bar charts from Pandas objects

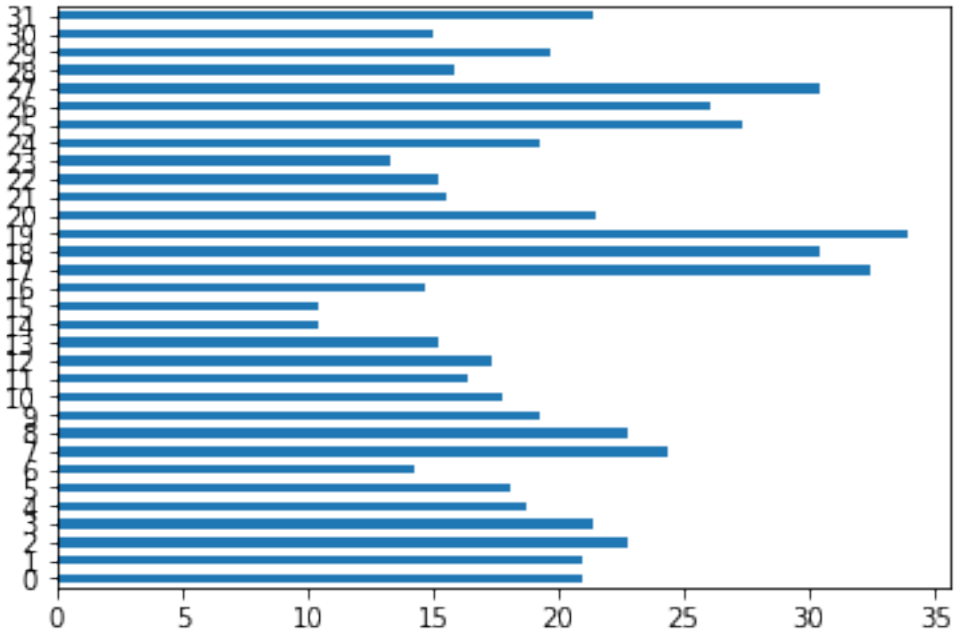
[7]:
```
mpg.plot(kind="bar")
```

[7]: <AxesSubplot:>



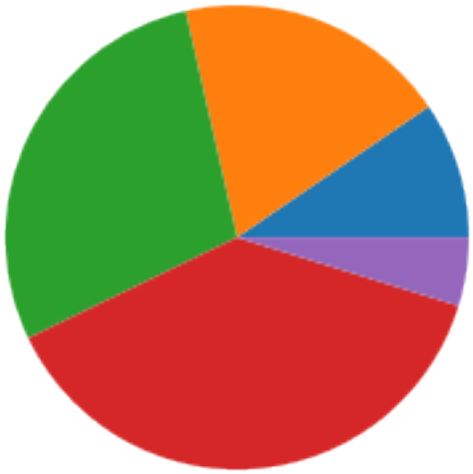
[8]:
```
mpg.plot(kind="barh")
```

[8]: <AxesSubplot:>



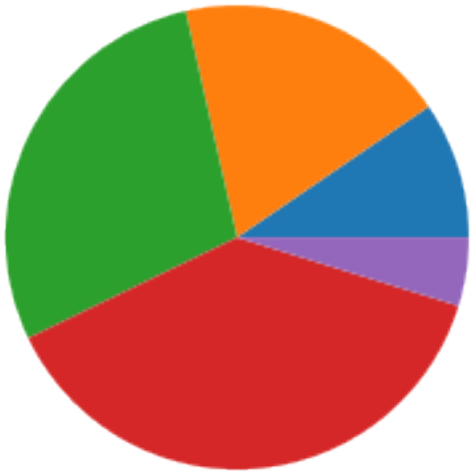
1.1.3 Creating a pie chart

[9]:
```
x = [1,2,3,4,0.5]
plt.pie(x)
plt.show()
```



1.1.4 Saving a plot

[10]:
```
plt.pie(x)
plt.savefig('pie_chart.png')
plt.show()
```



[11]:
```
%pwd
```

[11]:
```
'C:\\Users\\aadar\\Documents\\TERM2\\BDM 1034 - Application Design for Big
Data\\Week6\\Assignment'
```

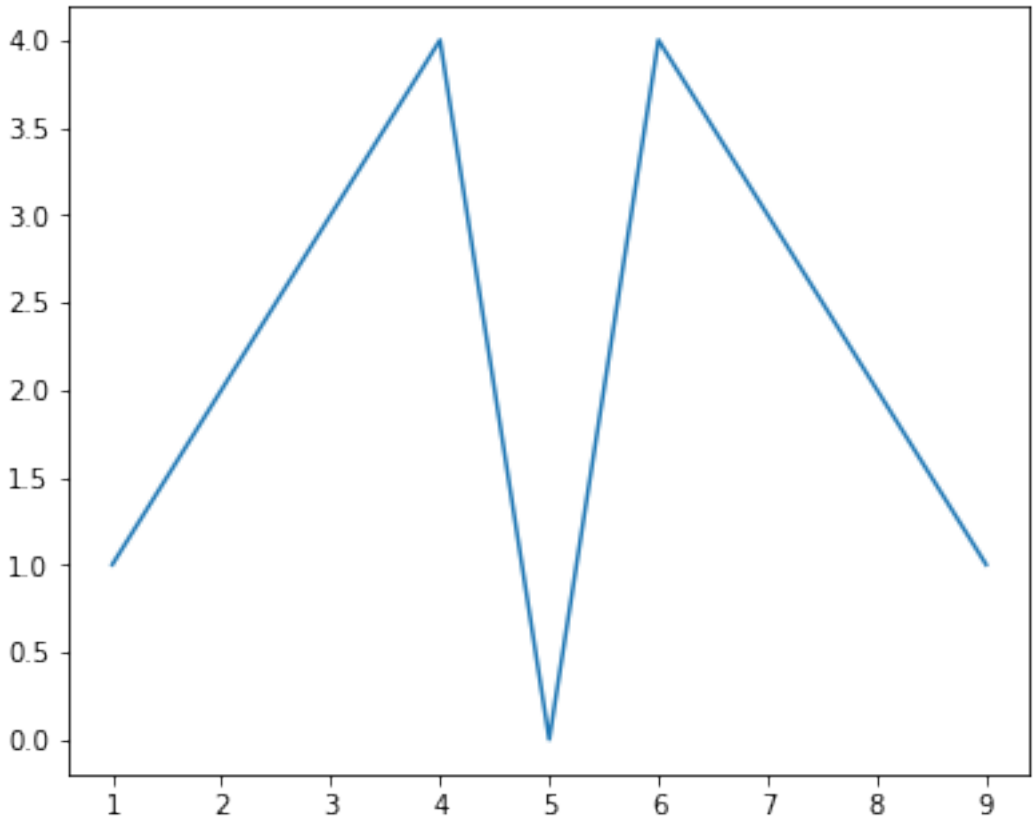
1.2 Segment 2 - Defining elements of a plot

[12]:
```
%matplotlib inline
rcParams['figure.figsize']= 5,4
```

1.2.1 Defining axes, ticks, and grids

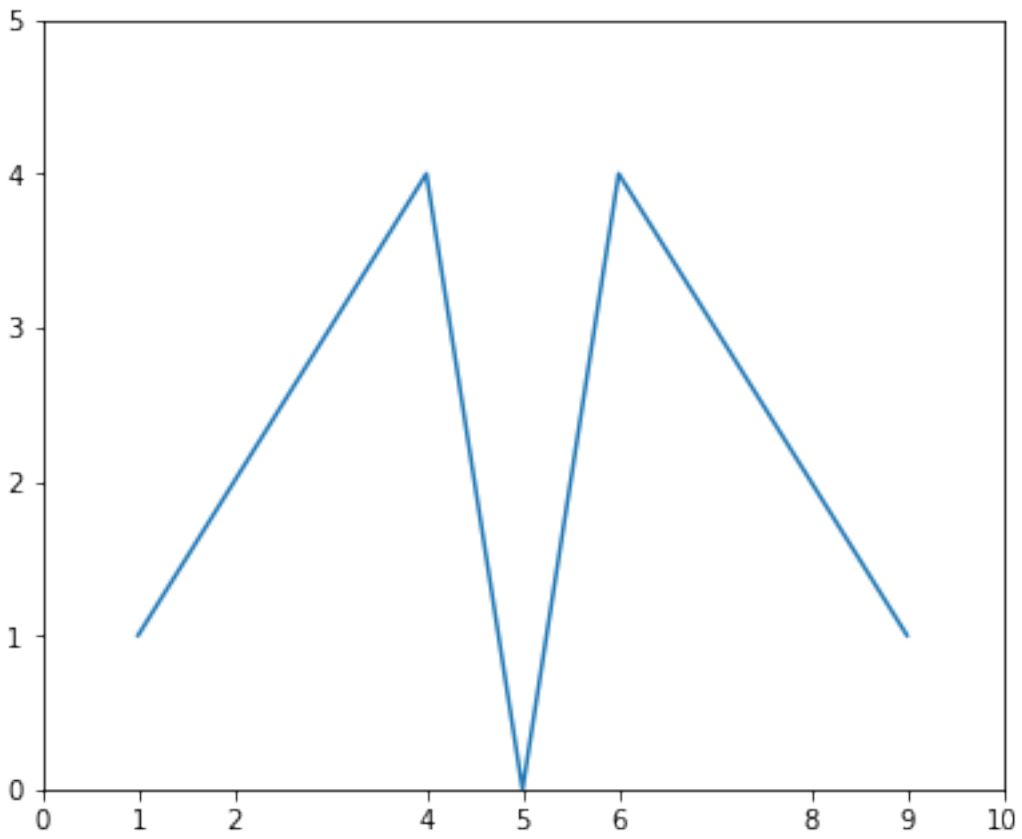
[13]:
```
x = range(1,10)
y = [1,2,3,4,0,4,3,2,1]

fig = plt.figure()
ax = fig.add_axes([.1,.1,1,1])

ax.plot(x,y)
```

[13]:
```
[<matplotlib.lines.Line2D at 0x26094a6d3a0>]
```



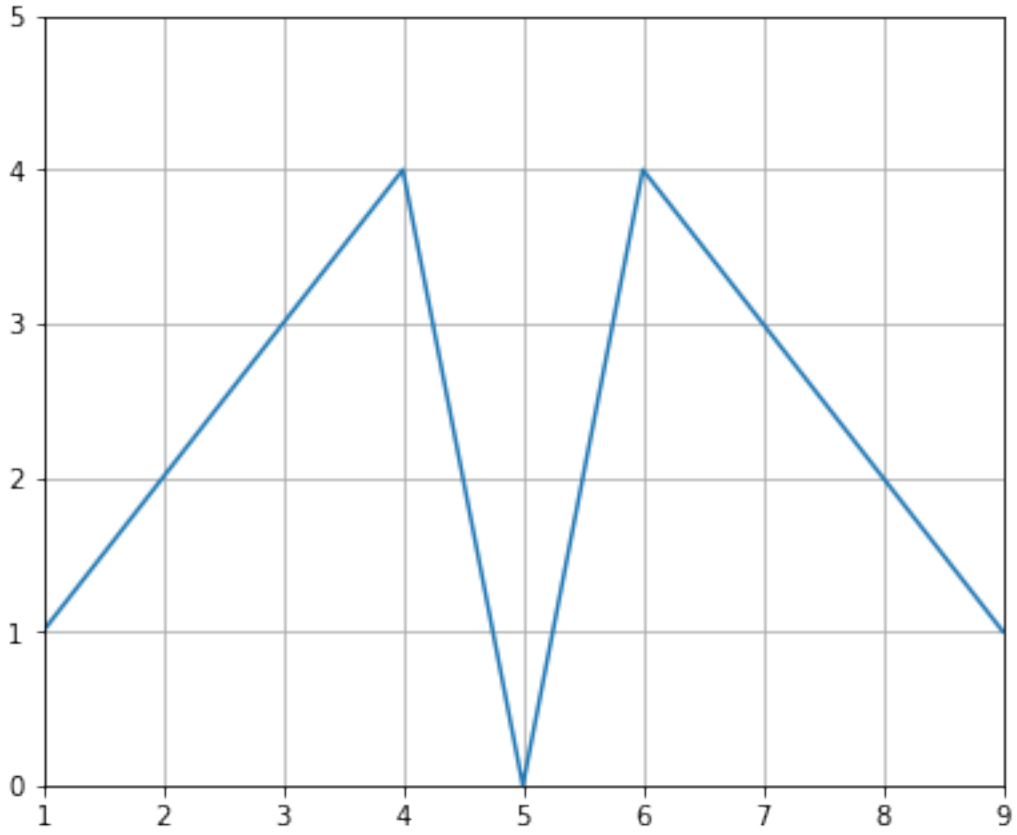
[14]:
```
fig = plt.figure()
ax = fig.add_axes([.1,.1,1,1])

ax.set_xlim([1,9])
ax.set_ylim([0,5])

ax.set_xticks([0,1,2,4,5,6,8,9,10])
ax.set_yticks([0,1,2,3,4,5])

ax.plot(x,y)
```

[14]: [<matplotlib.lines.Line2D at 0x26094ae4ee0>]



[15]:
```
fig = plt.figure()
ax = fig.add_axes([.1,.1,1,1])

ax.set_xlim([1,9])
ax.set_ylim([0,5])

ax.grid()
ax.plot(x,y)
```

[15]: [<matplotlib.lines.Line2D at 0x26095af8880>]

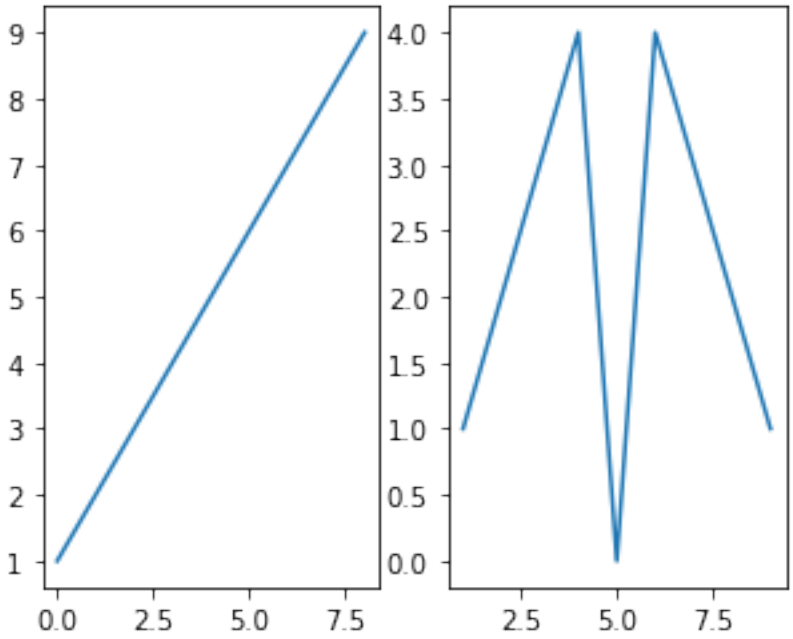


1.2.2 Generating multiple plots in one figure with subplots

[16]:
```
fig = plt.figure()
fig, (ax1, ax2) = plt.subplots(1,2)

ax1.plot(x)
ax2.plot(x,y)
```

[16]: [<matplotlib.lines.Line2D at 0x26095b96280>]

<Figure size 360x288 with 0 Axes>

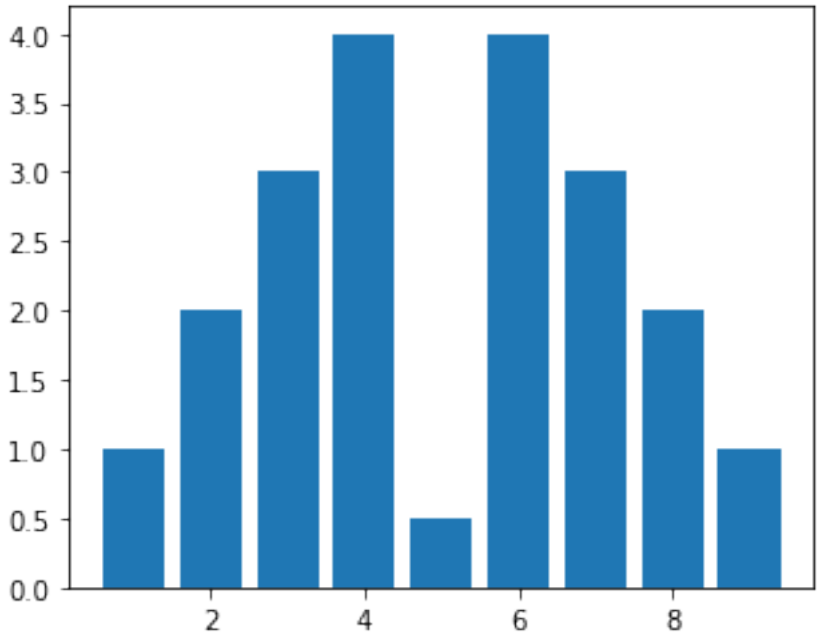


1.3 Segment 3 - Plot formatting

1.3.1 Defining plot color

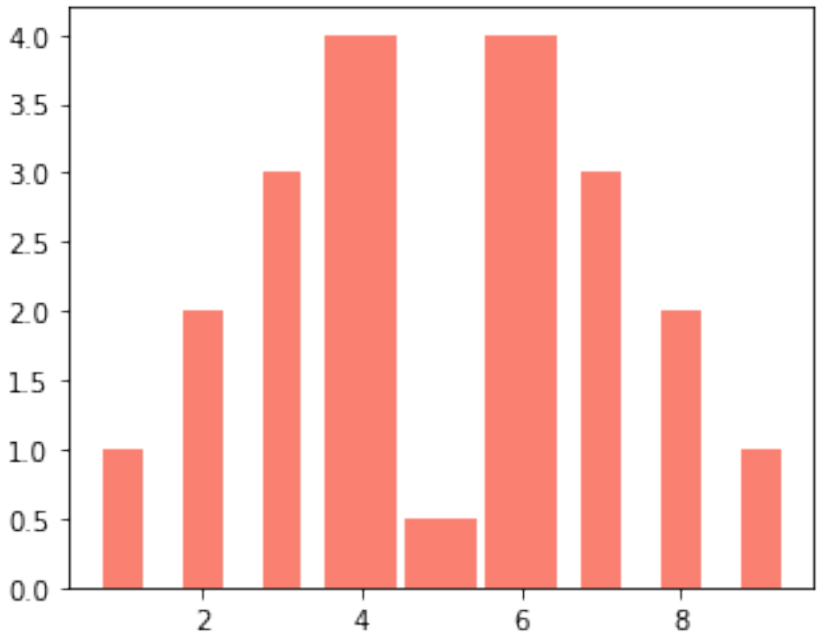
[17]:
```
x = range(1,10)
y = [1,2,3,4,0.5,4,3,2,1]

plt.bar(x,y)
```

[17]: <BarContainer object of 9 artists>



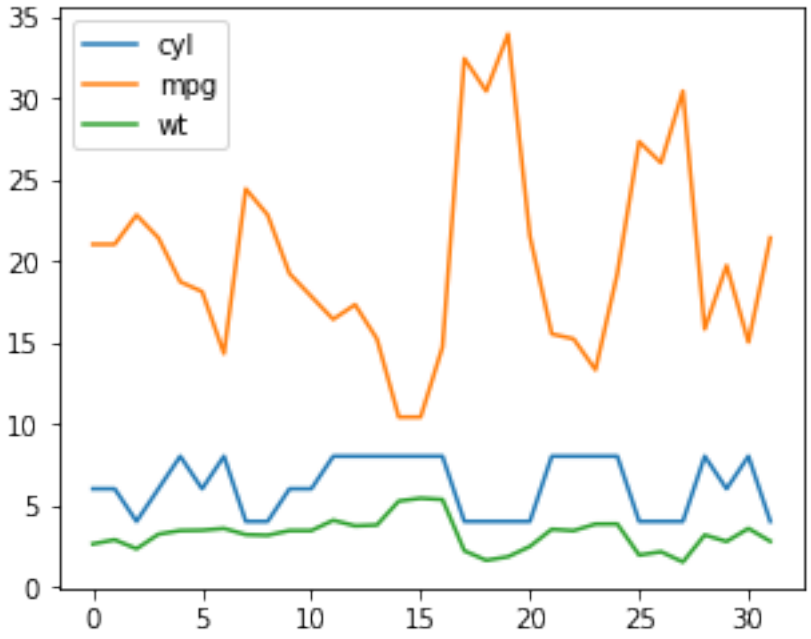
[18]:
```
wide = [.5,.5,.5,.9,.9,.9,.5,.5,.5]
color = ['salmon']
plt.bar(x, y, width=wide, color=color, align='center')
```

[18]: <BarContainer object of 9 artists>



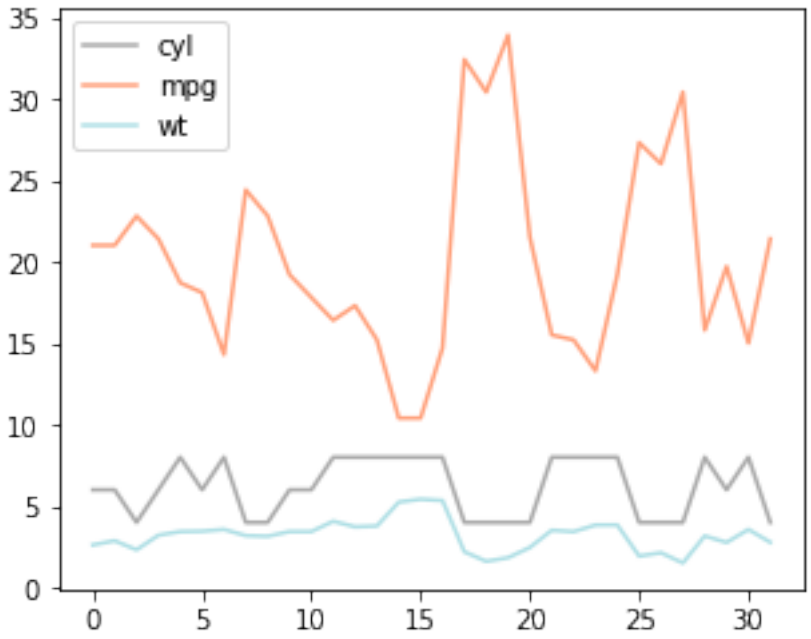
[19]:
```
address = './Data/mtcars.csv'

cars = pd.read_csv(address)
cars.columns = ['car_names','mpg','cyl','disp', 'hp', 'drat', 'wt', 'qsec',␣
↪'vs', 'am', 'gear', 'carb']

df = cars[['cyl', 'mpg', 'wt']]
df.plot()
```

[19]: <AxesSubplot:>

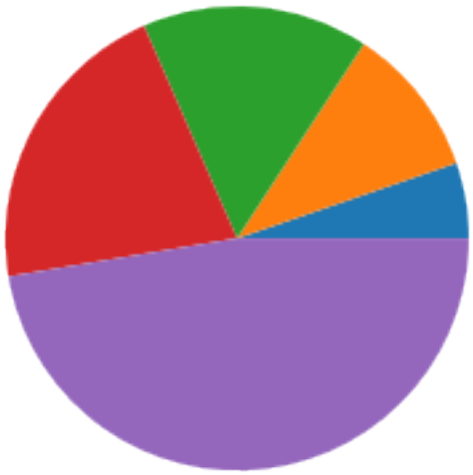


[20]:
```
color_theme = ['darkgray', 'lightsalmon', 'powderblue']
df.plot(color=color_theme)
```

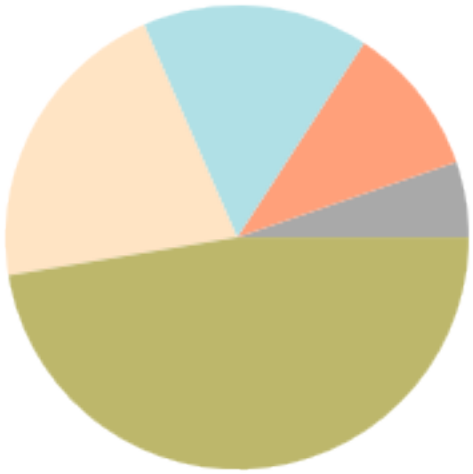
[20]: <AxesSubplot:>



[21]:
```
z = [1,2,3,4,9]
plt.pie(z)
plt.show()
```



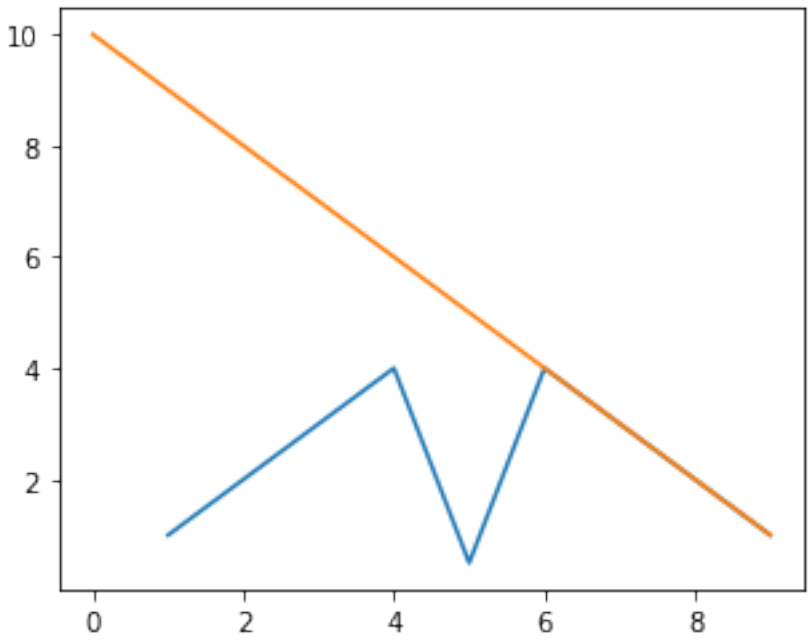
[22]:
```
color_theme = ['#A9A9A9', '#FFA07A', '#B0E0E6', '#FFE4C4', '#BDB76B']
plt.pie(z, colors=color_theme)
plt.show()
```



1.3.2 Customizing line styles

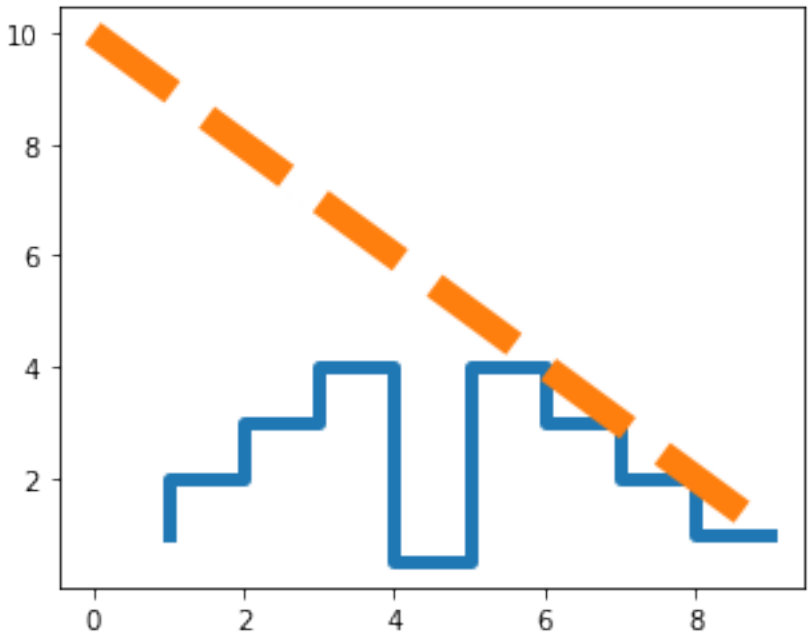
[23]:
```
x1 = range(0,10)
y1 = [10,9,8,7,6,5,4,3,2,1]

plt.plot(x, y)
plt.plot(x1, y1)
```

[23]:
```
[<matplotlib.lines.Line2D at 0x26095db2400>]
```



[24]:
```
plt.plot(x, y, ds='steps', lw=5)
plt.plot(x1, y1, ls='--', lw=10)
```

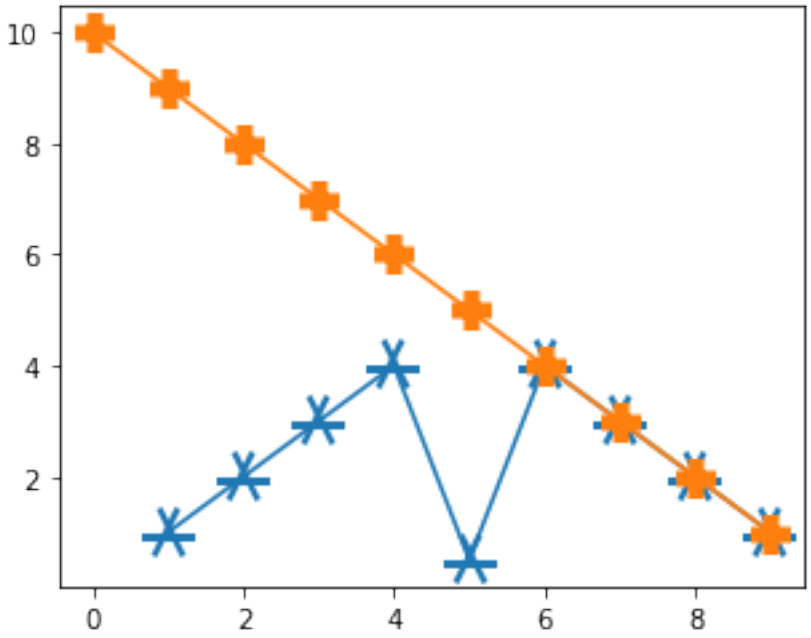
[24]: [<matplotlib.lines.Line2D at 0x26095db2fa0>]



1.3.3 Setting plot markers

[25]:
```
plt.plot(x, y, marker='1', mew=20)
plt.plot(x1, y1, marker='+', mew=15)
```

[25]:
```
[<matplotlib.lines.Line2D at 0x26095e5a460>]
```



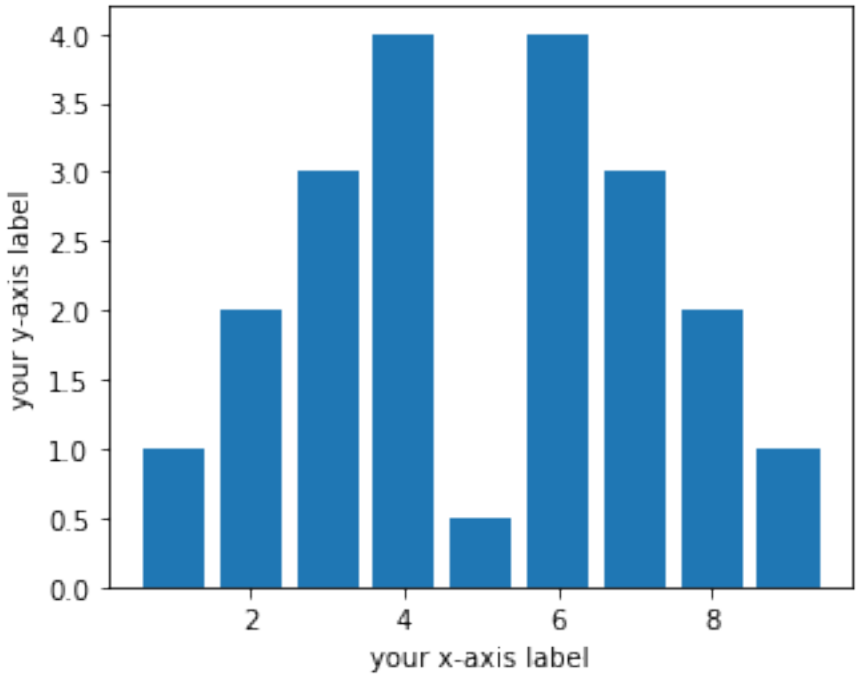
1.4 Segment 4 - Creating labels and annotations

1.4.1 Labeling plot features

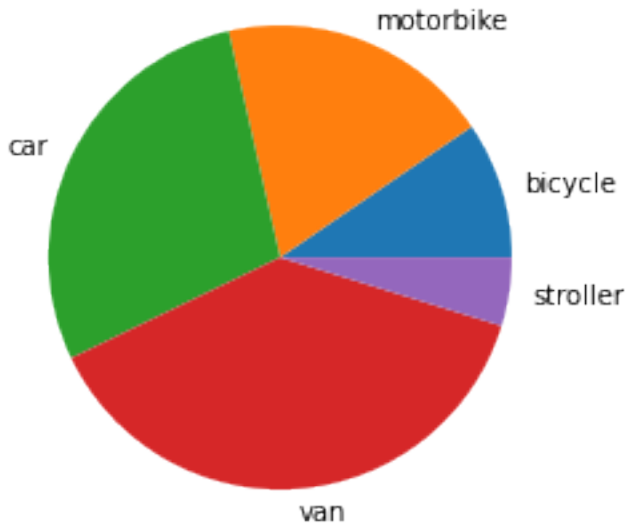
The functional method

[26]:
```
x = range(1,10)
y = [1,2,3,4,.5,4,3,2,1]
plt.bar(x,y)

plt.xlabel('your x-axis label')
plt.ylabel('your y-axis label')
```

[26]:
```
Text(0, 0.5, 'your y-axis label')
```



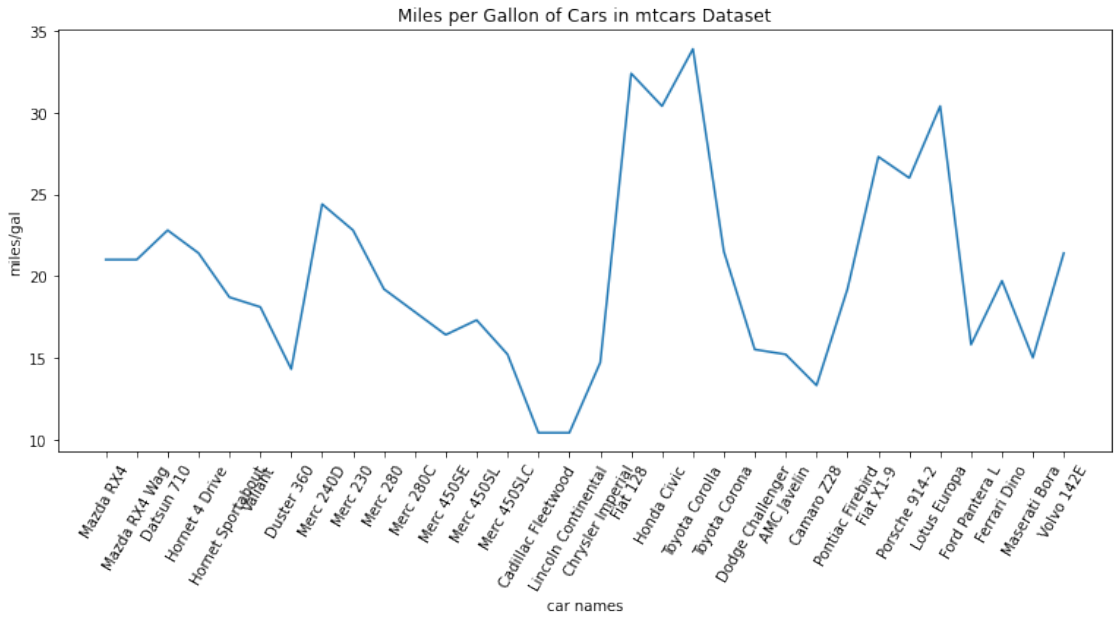
[27]:
```
z = [1,2,3,4,.5]
veh_type = ['bicycle', 'motorbike', 'car', 'van', 'stroller']

plt.pie(z, labels=veh_type)
plt.show()
```



The object-oriented method

[37]:
```
rcParams['figure.figsize']= 10,4
address = './Data/mtcars.csv'

cars = pd.read_csv(address)
cars.columns = ['car_names','mpg','cyl','disp', 'hp', 'drat', 'wt', 'qsec',␣
↪'vs', 'am', 'gear', 'carb']

mpg = cars.mpg

fig = plt.figure()
ax = fig.add_axes([.1,.1,1,1])

mpg.plot()

ax.set_xticks(range(32))

ax.set_xticklabels(cars.car_names, rotation=60, fontsize='medium')
ax.set_title('Miles per Gallon of Cars in mtcars Dataset')

ax.set_xlabel('car names')
ax.set_ylabel('miles/gal')
```

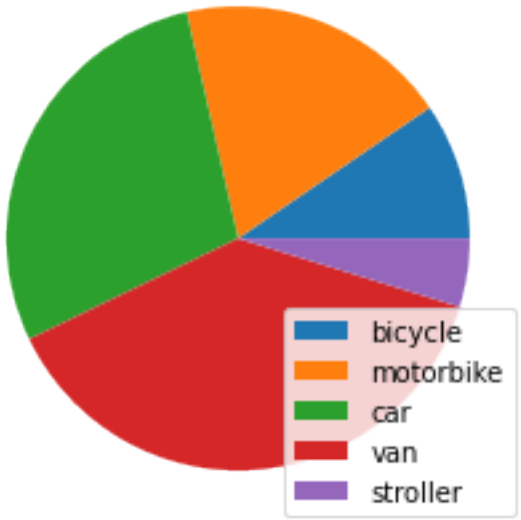
[37]:
```
Text(0, 0.5, 'miles/gal')
```



1.4.2 Adding a legend to your plot

The functional method

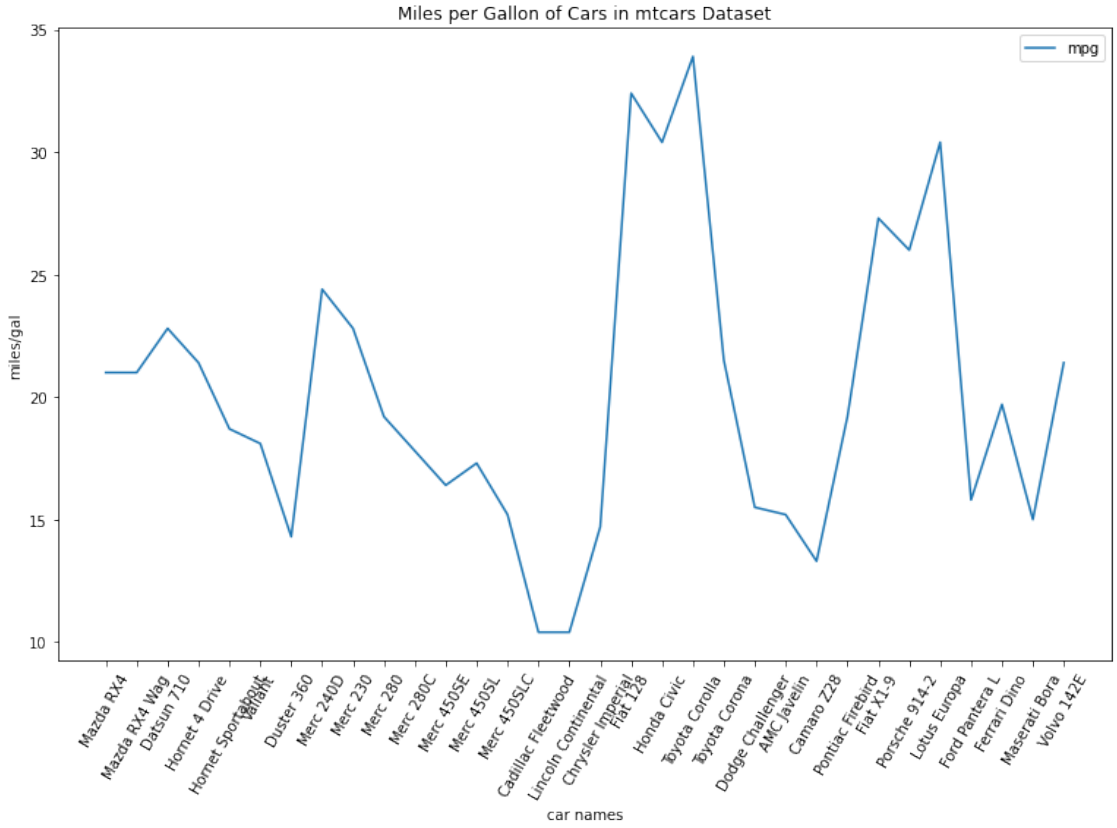
[29]:
```
plt.pie(z)
plt.legend(veh_type, loc='best')
plt.show()
```



The object-oriented method

[38]:
```
rcParams['figure.figsize']= 10,6
fig = plt.figure()
ax = fig.add_axes([.1,.1,1,1])

mpg.plot()

ax.set_xticks(range(32))

ax.set_xticklabels(cars.car_names, rotation=60, fontsize='medium')
ax.set_title('Miles per Gallon of Cars in mtcars Dataset')

ax.set_xlabel('car names')
ax.set_ylabel('miles/gal')

ax.legend(loc='best')
```

[38]:
```
<matplotlib.legend.Legend at 0x260963bdf40>
```



1.4.3 Annotating your plot

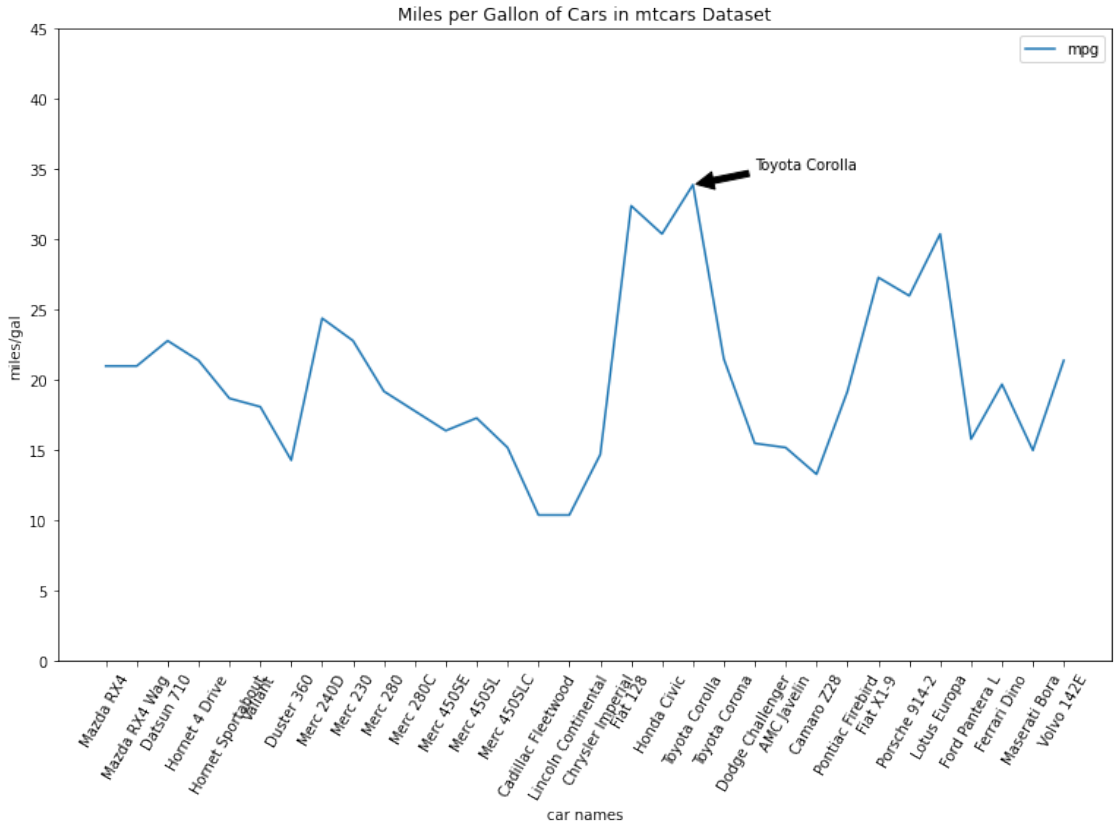
[31]:
```
mpg.max()
```

[31]: 33.9

[39]:
```
rcParams['figure.figsize']= 10,6
fig = plt.figure()
ax = fig.add_axes([.1,.1,1,1])

mpg.plot()

ax.set_xticks(range(32))

ax.set_xticklabels(cars.car_names, rotation=60, fontsize='medium')
ax.set_title('Miles per Gallon of Cars in mtcars Dataset')

ax.set_xlabel('car names')
ax.set_ylabel('miles/gal')

ax.legend(loc='best')
```

```
ax.set_ylim([0,45])

ax.annotate('Toyota Corolla', xy= (19,33.9), xytext=(21,35),
            arrowprops=dict(facecolor='black', shrink=0.05))
```

[39]:
```
Text(21, 35, 'Toyota Corolla')
```



1.4.4 The simplest time series plot

[43]:
```
rcParams['figure.figsize'] = 5, 4
address = './Data/Superstore-Sales.csv'

df = pd.read_csv(address, index_col='Order Date', encoding='cp1252',␣
 ↪parse_dates=True)
df.head()
```

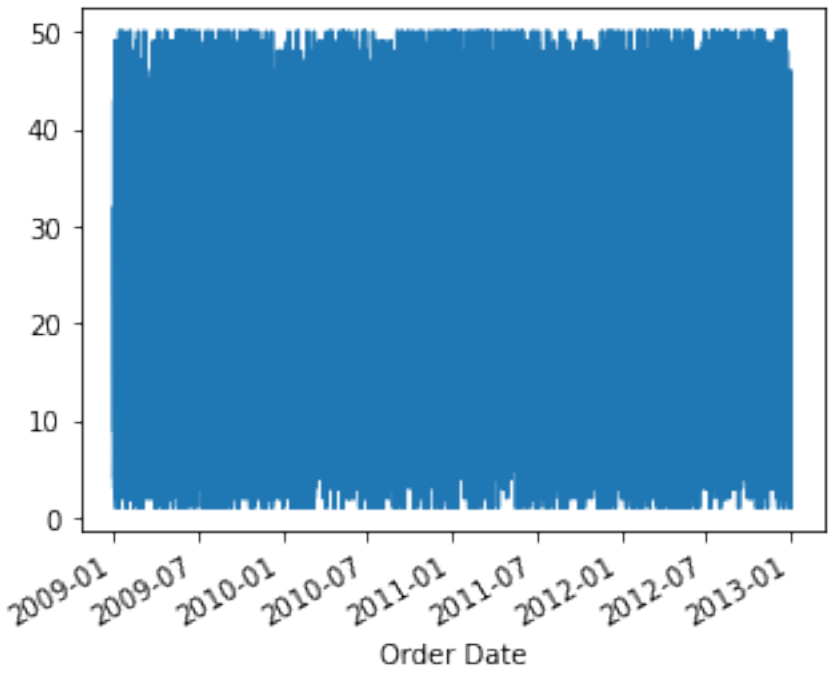
[43]:

Order Date	Row ID	Order ID	Order Priority	Order Quantity	Sales
2010-10-13	1	3	Low	6	261.5400
2012-10-01	49	293	High	49	10123.0200

```
2012-10-01      50       293           High              27    244.5700
2011-07-10      80       483           High              30   4965.7595
2010-08-28      85       515  Not Specified              19    394.2700

             Discount       Ship Mode   Profit  Unit Price  Shipping Cost  \
Order Date
2010-10-13       0.04     Regular Air  -213.25       38.94          35.00
2012-10-01       0.07  Delivery Truck   457.81      208.16          68.02
2012-10-01       0.01     Regular Air    46.71        8.69           2.99
2011-07-10       0.08     Regular Air  1198.97      195.99           3.99
2010-08-28       0.08     Regular Air    30.94       21.78           5.94

                  Customer Name Province   Region Customer Segment  \
Order Date
2010-10-13  Muhammed MacIntyre  Nunavut  Nunavut   Small Business
2012-10-01        Barry French  Nunavut  Nunavut         Consumer
2012-10-01        Barry French  Nunavut  Nunavut         Consumer
2011-07-10       Clay Rozendal  Nunavut  Nunavut        Corporate
2010-08-28      Carlos Soltero  Nunavut  Nunavut         Consumer

            Product Category            Product Sub-Category  \
Order Date
2010-10-13  Office Supplies          Storage & Organization
2012-10-01  Office Supplies                      Appliances
2012-10-01  Office Supplies  Binders and Binder Accessories
2011-07-10       Technology    Telephones and Communication
2010-08-28  Office Supplies                      Appliances

                                                 Product Name  \
Order Date
2010-10-13   Eldon Base for stackable storage shelf, platinum
2012-10-01  1.7 Cubic Foot Compact "Cube" Office Refrigera…
2012-10-01   Cardinal Slant-D® Ring Binder, Heavy Gauge Vinyl
2011-07-10                                              R380
2010-08-28                          Holmes HEPA Air Purifier

            Product Container  Product Base Margin   Ship Date
Order Date
2010-10-13        Large Box                 0.80  10/20/2010
2012-10-01       Jumbo Drum                 0.58   10/2/2012
2012-10-01        Small Box                 0.39   10/3/2012
2011-07-10        Small Box                 0.58   7/12/2011
2010-08-28       Medium Box                 0.50   8/30/2010
```

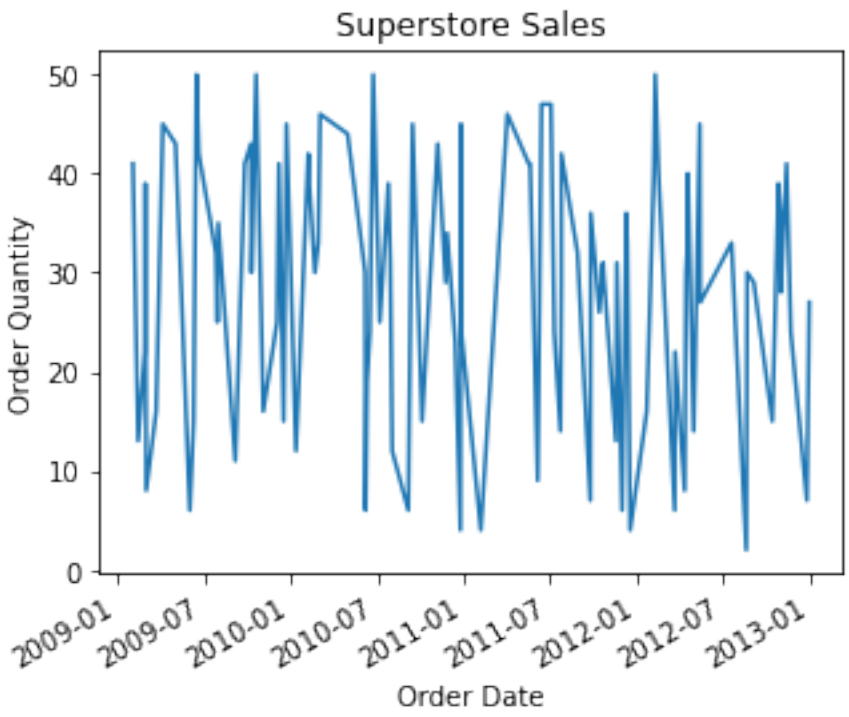
[44]:
```
df['Order Quantity'].plot()
```

[44]:
```
<AxesSubplot:xlabel='Order Date'>
```



[45]:
```
df2 = df.sample(n=100, random_state=25, axis=0)

plt.xlabel('Order Date')
plt.ylabel('Order Quantity')
plt.title('Superstore Sales')

df2['Order Quantity'].plot()
```

[45]:
```
<AxesSubplot:title={'center':'Superstore Sales'}, xlabel='Order Date',
ylabel='Order Quantity'>
```



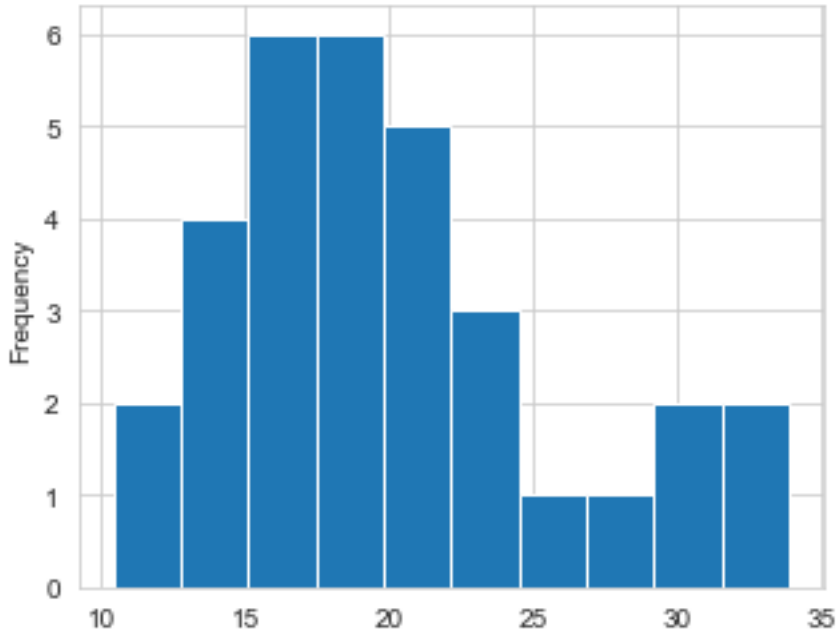
1.5 Segment 6 - Creating statistical data graphics|

[46]:
```
%matplotlib inline
rcParams['figure.figsize'] = 5, 4
import seaborn as sb
sb.set_style('whitegrid')
```

1.5.1 Eyeballing dataset distributions with histograms

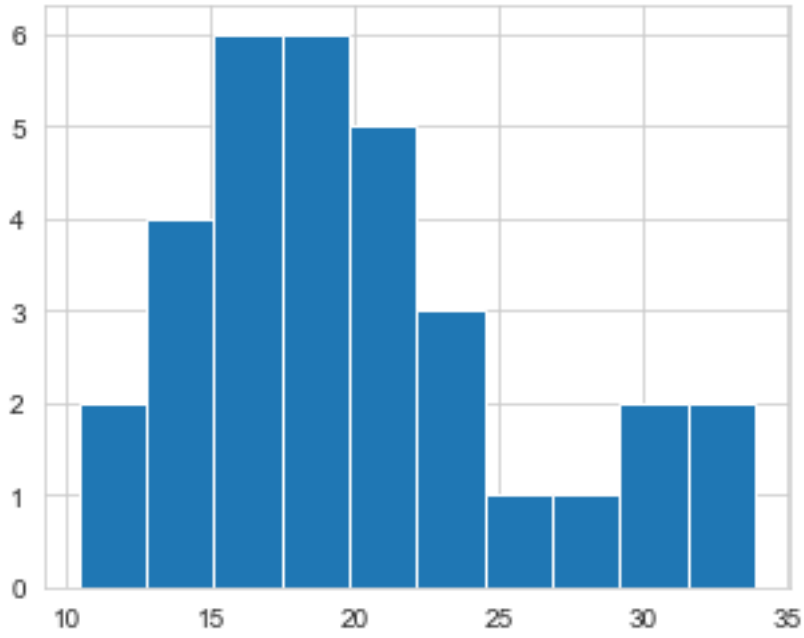
[48]:
```
address = './Data/mtcars.csv'

cars = pd.read_csv(address)

cars.columns = ['car_names','mpg','cyl','disp', 'hp', 'drat', 'wt', 'qsec',␣
↪'vs', 'am', 'gear', 'carb']

cars.index = cars.car_names

mpg = cars['mpg']

mpg.plot(kind='hist')
```

[48]:
```
<AxesSubplot:ylabel='Frequency'>
```



[49]:
```
plt.hist(mpg)
plt.plot()
```

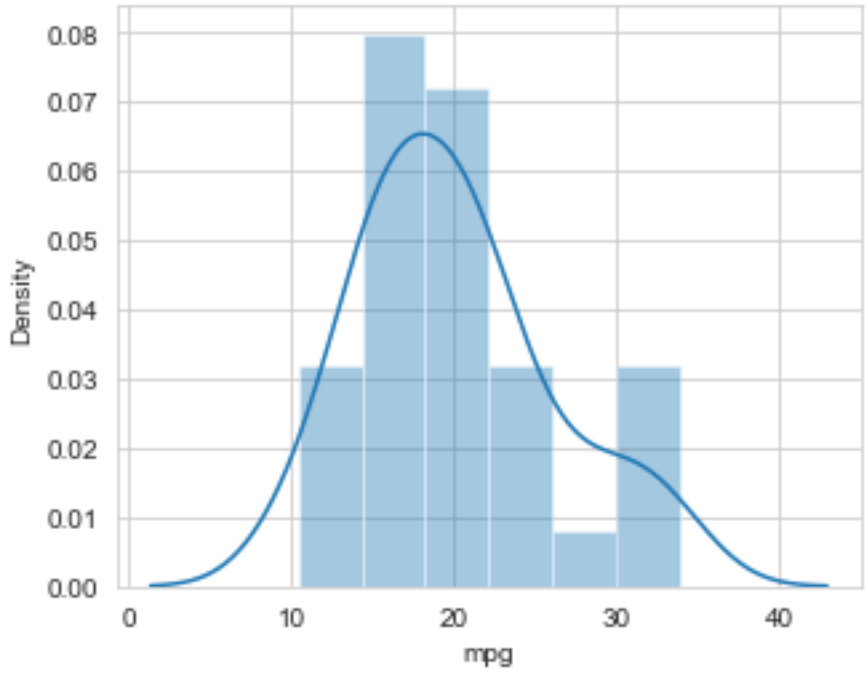
[49]: []



[50]:
```
sb.distplot(mpg)
```

```
E:\Anaconda\lib\site-packages\seaborn\distributions.py:2557: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version.
Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

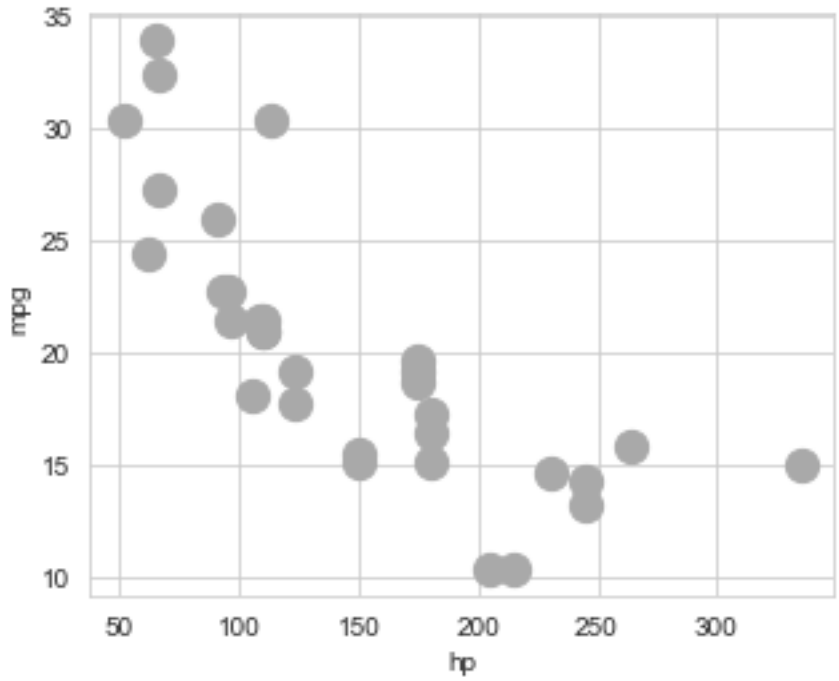
[50]:
```
<AxesSubplot:xlabel='mpg', ylabel='Density'>
```



1.5.2 Seeing scatterplots in action

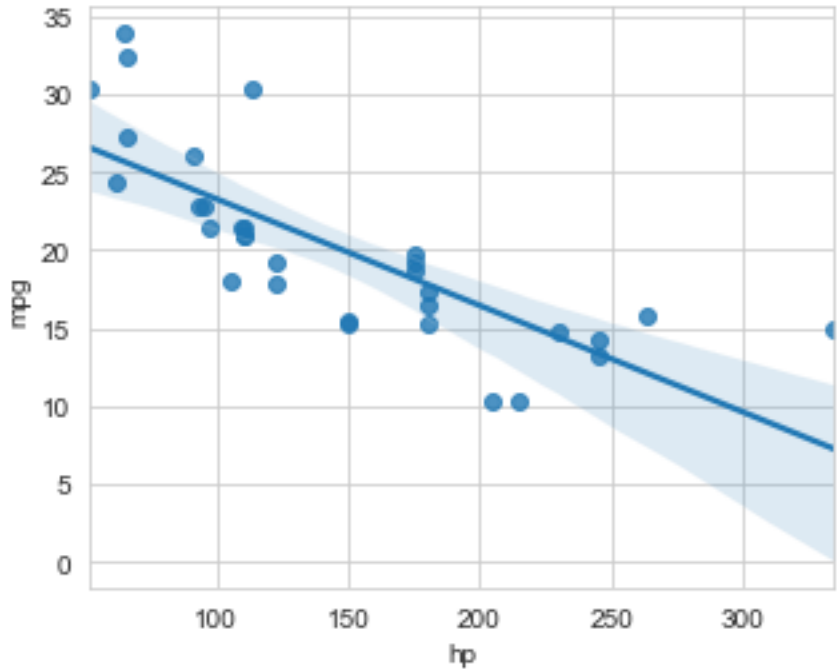
[52]:
```
cars.plot(kind='scatter', x='hp', y='mpg', c=['darkgray'], s=150)
```

[52]:
```
<AxesSubplot:xlabel='hp', ylabel='mpg'>
```



[53]:
```
sb.regplot(x='hp', y='mpg', data=cars, scatter=True)
```

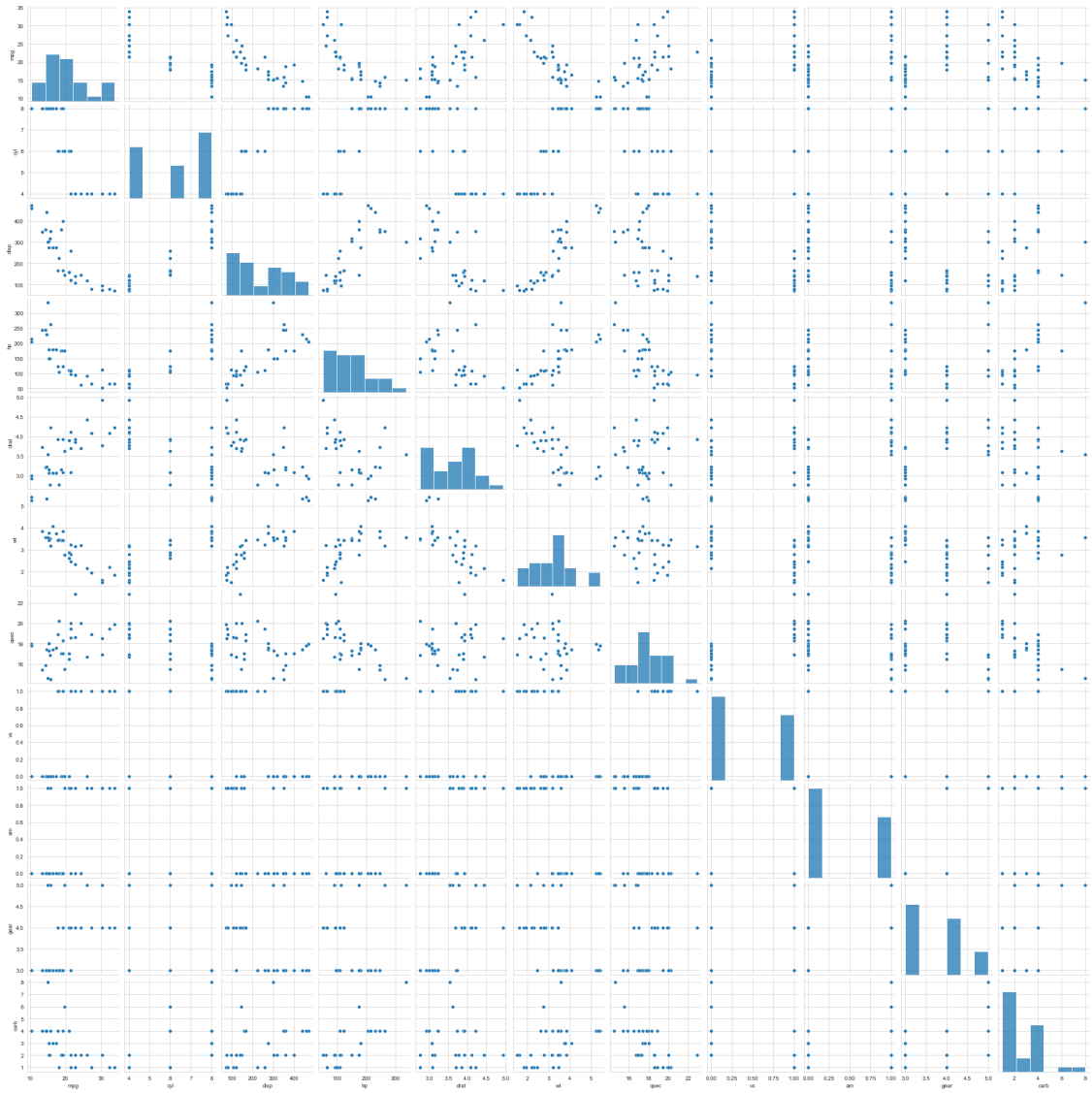
[53]:
```
<AxesSubplot:xlabel='hp', ylabel='mpg'>
```



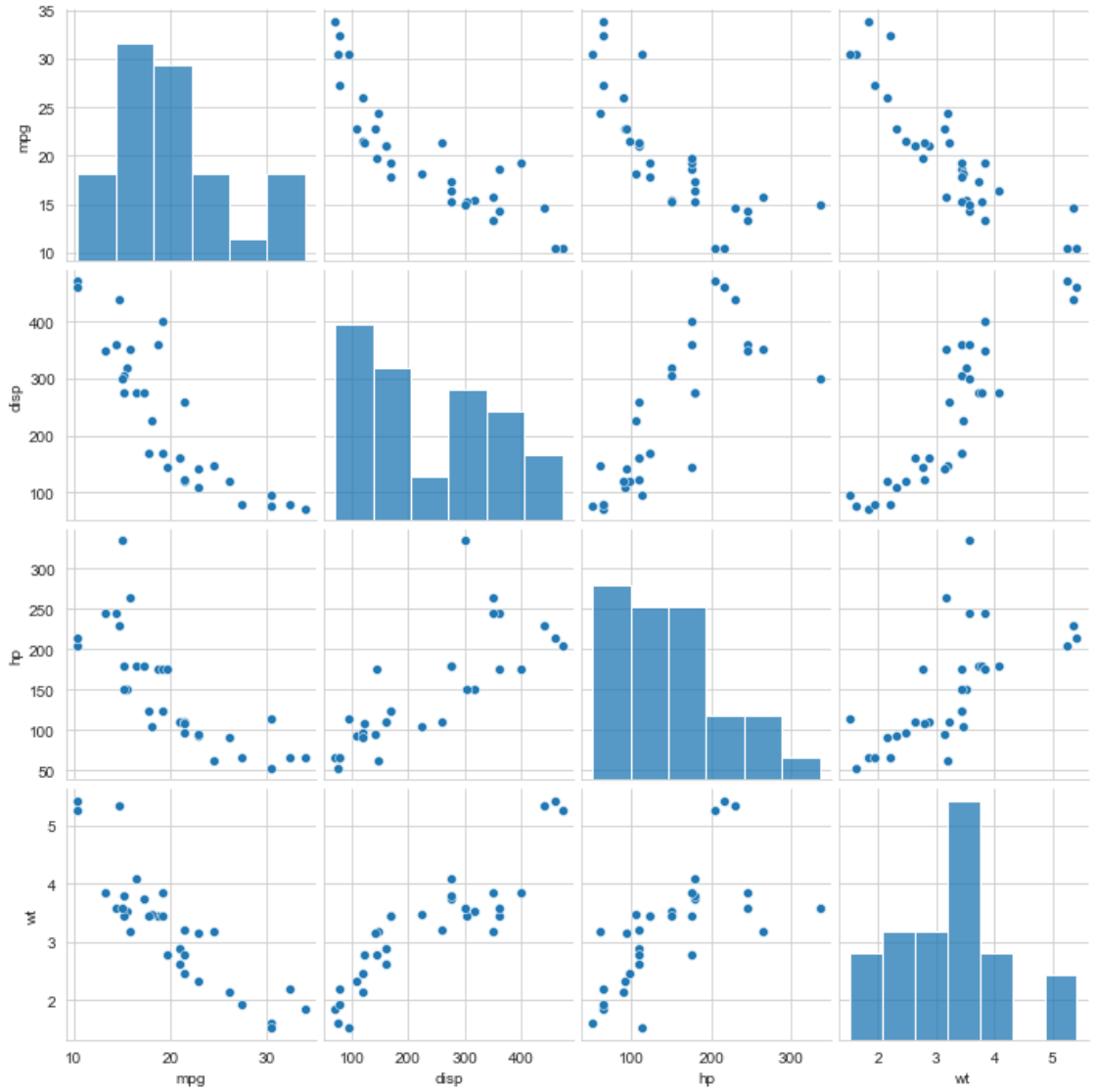
1.5.3 Generating a scatter plot matrix

[54]:
```
sb.pairplot(cars)
```

[54]:
```
<seaborn.axisgrid.PairGrid at 0x2609bee8d30>
```



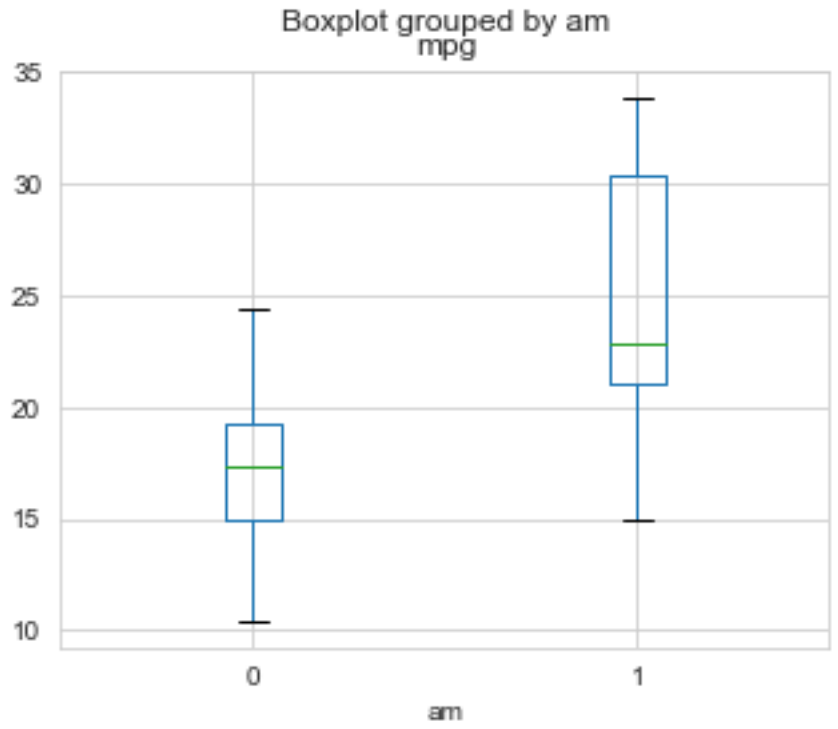
[56]:
```
cars_subset = cars[['mpg', 'disp', 'hp', 'wt']]
sb.pairplot(cars_subset)
plt.show()
```

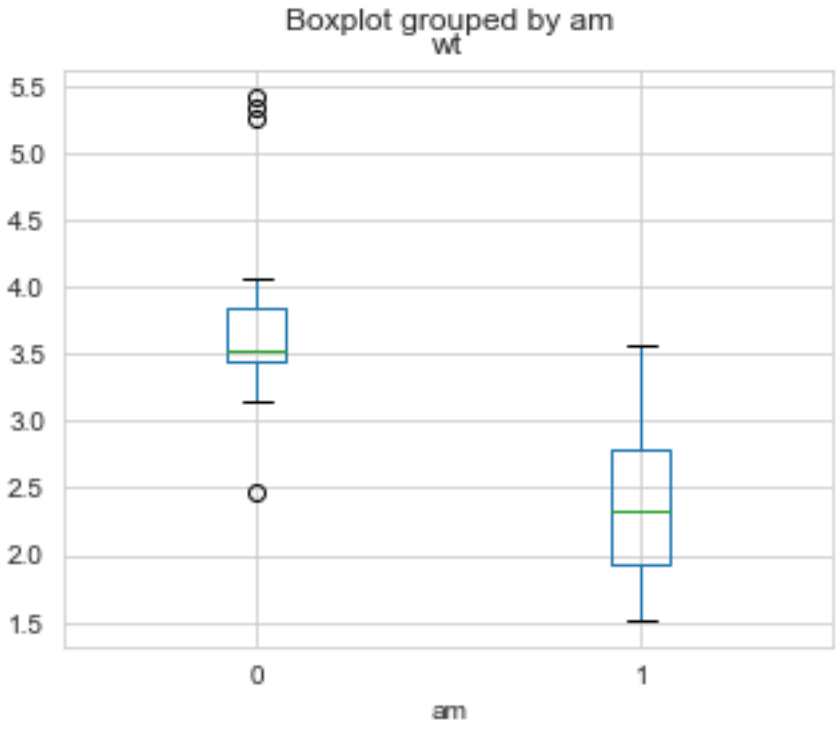


1.5.4 Building boxplots

[57]:
```
cars.boxplot(column='mpg', by='am')
cars.boxplot(column='wt', by='am')
```

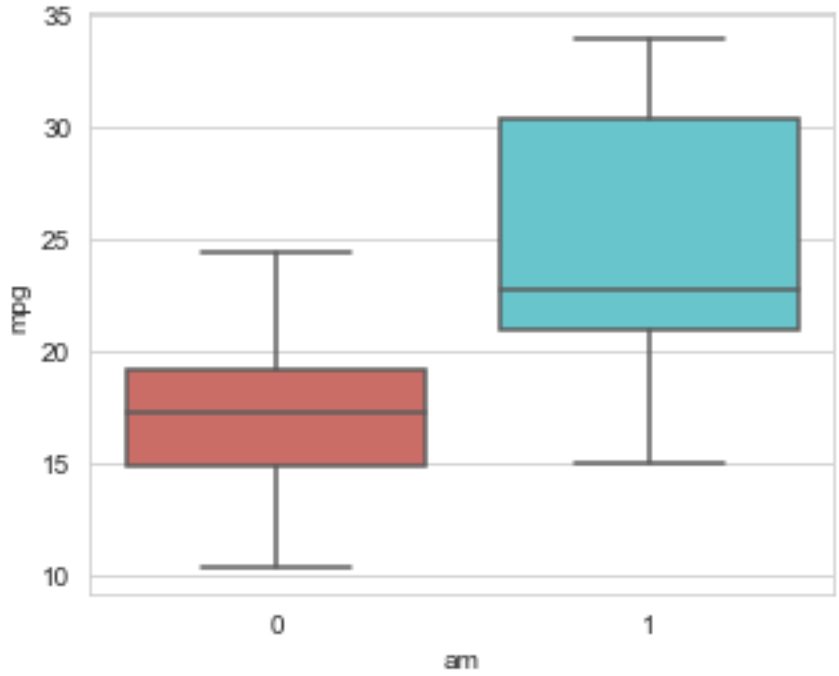
[57]:
```
<AxesSubplot:title={'center':'wt'}, xlabel='am'>
```


Boxplot grouped by am
mpg
35
30
25
20
15
10
0
1
am


Boxplot grouped by am
wt
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
0
1
am

[58]:
```
sb.boxplot(x='am', y='mpg', data=cars, palette='hls')
```

[58]: <AxesSubplot:xlabel='am', ylabel='mpg'>



[ ]:
```

```