Descriptive Statistics

Imports

In [1]:
```
import numpy as np
import pandas as pd
import scipy.stats as stats
import matplotlib.pyplot as plt
import random
import math
import seaborn as sns
```

1. Descriptive Statistics

Measures of Center

Measures of center are statistics that give us a sense of the "middle" of a numeric variable. In other words, centrality measures give you a sense of a typical value you'd expect to see. Common measures of center include the mean, median and mode.

In [2]:
```
mtcars = pd.read_csv('data/mtcars.csv') # get an example data set
mtcars.head()
```

Out[2]:

	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

In [3]:
```
mtcars.shape
```

Out[3]:
```
(32, 12)
```

In [4]:
```
mtcars.set_index('model',inplace=True)
```

In [5]:
```
mtcars
```

Out[5]:

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

Sample Mean:

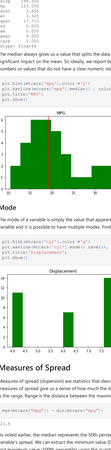
The mean is simply an average: the sum of the values divided by the total number of records. We can use *df.mean()* to get the mean of each column in a DataFrame:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

In [6]:
```
mtcars.mean()                # Get the mean of each column
```

Out[6]:
```
mpg      20.090625
cyl       6.187500
disp    230.721875
hp      146.687500
drat      3.596563
wt        3.217250
qsec     17.848750
vs        0.437500
am        0.406250
gear      3.687500
carb      2.812500
dtype: float64
```

In [7]:
```
plt.hist(mtcars['mpg'],color ='r')
plt.axvline(mtcars['mpg'].mean(), c='cyan')
plt.title('MPG')
plt.show()
```



Median

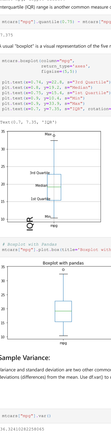
The median of a distribution is the value where 50% of the data lies below it and 50% lies above it. In essence, the median splits the data in half. The median is also known as the 50% percentile since 50% of the observations are found below it. You can get the median using the *df.median()* function:

In [8]:
```
mtcars.median()              # Get the median of each column
```

Out[8]:
```
mpg     19.200
cyl      6.000
disp   196.300
hp     123.000
drat     3.695
wt       3.325
qsec    17.710
vs       0.000
am       0.000
gear     4.000
carb     2.000
dtype: float64
```

The median always gives us a value that splits the data into two halves while the mean is a numeric average so extreme values can have a significant impact on the mean. So ideally, we report both. Furthermore, if you have data that is purely categorical values represented as numbers or values that do not have a clear numeric relation between values, then the mean is not really meaningful.

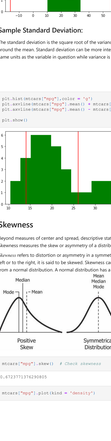
In [9]:
```
plt.hist(mtcars['mpg'],color ='r')
plt.axvline(mtcars['mpg'].median(), color='red')
plt.title('MPG')
plt.show()
```



Mode

The mode of a variable is simply the value that appears most frequently. Unlike mean and median, you can take the mode of a categorical variable and it is possible to have multiple modes. Find the mode of *cyl* variable:

In [10]:
```
plt.hist(mtcars['cyl'],color ='y')
plt.axvline(mtcars['cyl'].mode()[0],color='red')
plt.title('Displacement')
plt.show()
```



Measures of Spread

Measures of spread (dispersion) are statistics that describe how data varies. While measures of center give us an idea of the typical value, measures of spread give us a sense of how much the data tends to diverge from the typical value. One of the simplest measures of spread is the range. Range is the distance between the maximum and minimum observations:

In [11]:
```
max(mtcars['mpg']) - min(mtcars['mpg'])
```

Out[11]:
```
23.5
```

As noted earlier, the median represents the 50th percentile of a data set. A summary of several percentiles can be used to describe a variable's spread. We can extract the minimum value (0th percentile), first quartile (25th percentile), median, third quartile(75th percentile) and maximum value (100th percentile) using the *quantile()* function:

In [12]:
```
five_num = [mtcars['mpg'].quantile(0),
            mtcars['mpg'].quantile(0.25),
            mtcars['mpg'].quantile(0.50),
            mtcars['mpg'].quantile(0.75),
            mtcars['mpg'].quantile(1)]

five_num
```

Out[12]:
```
[10.4, 15.425, 19.2, 22.8, 33.9]
```

Since these values are so commonly used to describe data, they are known as the "five number summary". They are the same percentile values returned by *df.describe()*:

In [13]:
```
mtcars['mpg'].describe()
```

Out[13]:
```
count    32.000000
mean     20.090625
std       6.026948
min      10.400000
25%      15.425000
50%      19.200000
75%      22.800000
max      33.900000
Name: mpg, dtype: float64
```

The interquartile (IQR) range is another common measure of spread. IQR is the distance between the 3rd quartile and the 1st quartile:

In [14]:
```
mtcars['mpg'].quantile(0.75) - mtcars['mpg'].quantile(0.25)
```

Out[14]:
```
7.375
```

A usual "boxplot" is a visual representation of the five number summary and IQR.

In [15]:
```
mtcars.boxplot(column="mpg",
               return_type='axes',
               figsize=(5,5))

plt.text(x=0.74, y=22.8, s="3rd Quartile")
plt.text(x=0.8, y=19.2, s="Median")
plt.text(x=0.75, y=15.4, s="1st Quartile")
plt.text(x=0.9, y=10.4, s="Min")
plt.text(x=0.9, y=33.9, s="Max")
plt.text(x=0.74, y=11.05, s="IQR", fontsize=15, size=25)

plt.show()
```


In [16]:
```
# Boxplot with Pandas
mtcars['mpg'].plot.box(title='Boxplot with pandas');
```


Sample Variance:

Variance and standard deviation are two other common measures of spread. The variance of a distribution is the average of the squared deviations (differences) from the mean. Use *df.var()* to check variance.

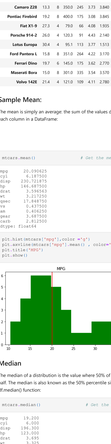
$$Var_x = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

In [17]:
```
mtcars['mpg'].var()
```

Out[17]:
```
36.3241028225806
```

In [18]:
```
plt.hist(mtcars['mpg'],color = 'g')
plt.axvline(mtcars['mpg'].mean()+ mtcars['mpg'].var(), color='red')
plt.axvline(mtcars['mpg'].mean()- mtcars['mpg'].var(), color='red')

plt.show()
```



Sample Standard Deviation:

The standard deviation is the square root of the variance.It is used to quantify the amount of variation or dispersion of a set of data values around the mean. Standard deviation can be more interpretable than variance, since the standard deviation is expressed in terms of the same units as the variable in question while variance is expressed in terms of units squared. Use *df.std()* to check the standard deviation:

$$Std_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

In [19]:
```
plt.hist(mtcars['mpg'],color = 'g')
plt.axvline(mtcars['mpg'].mean()+ mtcars['mpg'].std(), color='red')
plt.axvline(mtcars['mpg'].mean()- mtcars['mpg'].std(), color='red')

plt.show()
```


Skewness

Beyond measures of center and spread, descriptive statistics include measures that give you a sense of the shape of a distribution. Skewness measures the skew or asymmetry of a distribution while kurtosis measures the "peakedness" of a distribution.

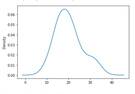
Skewness refers to distortion or asymmetry in a symmetrical bell curve, or normal distribution, in a set of data. If the curve is shifted to the left or to the right, it is said to be skewed. Skewness can be quantified as a representation of the extent to which a given distribution varies from a normal distribution. A normal distribution has a skew of zero.

In [20]:
```
mtcars['mpg'].skew()  # Check skewness
```

Out[20]:
```
0.6723771376290805
```

In [21]:
```
mtcars['mpg'].plot(kind = 'density')
```

Out[25]: <Axes:ylabel="Density">



* Happy Learning ***