In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from ggplot import mtcars
%matplotlib inline
```

```
C:\Anaconda\lib\site-packages\ggplot\utils.py:81: FutureWarning: pandas.tsli
b is deprecated and will be removed in a future version.
You can access Timestamp as pandas.Timestamp
  pd.tslib.Timestamp,
C:\Anaconda\lib\site-packages\ggplot\stats\smoothers.py:4: FutureWarning: Th
e pandas.lib module is deprecated and will be removed in a future version. T
hese are private functions and can be accessed from pandas._libs.lib instead
  from pandas.lib import Timestamp
C:\Anaconda\lib\site-packages\statsmodels\compat\pandas.py:56: FutureWarnin
g: The pandas.core.datetools module is deprecated and will be removed in a f
uture version. Please use the pandas.tseries module instead.
  from pandas.core import datetools
```

Mean(Average)

- Mean is defined as the sum of all the observations divided by number of observations. It tells us how our each data point approx look like. The main disadvantage of analysing distribution of the data by mean is, it gets effected by the outliers.

In [3]:

```
mtcars.index = mtcars['name']
mtcars.mean()
```

Out[3]:

```
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]:

```
mtcars.head()
```

Out[7]:

	name	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
name												
Mazda RX4	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2

In [8]:

```
mtcars.mean(axis = 1)
```

Out[8]:

```
name
Mazda RX4              29.907273
Mazda RX4 Wag          29.981364
Datsun 710             23.598182
Hornet 4 Drive         38.739545
Hornet Sportabout      53.664545
Valiant                35.049091
Duster 360             59.720000
Merc 240D              24.634545
Merc 230               27.233636
Merc 280               31.860000
Merc 280C              31.787273
Merc 450SE             46.430909
Merc 450SL             46.500000
Merc 450SLC            46.350000
Cadillac Fleetwood     66.232727
Lincoln Continental    66.058545
Chrysler Imperial      65.972273
Fiat 128               19.440909
Honda Civic            17.742273
Toyota Corolla         18.814091
Toyota Corona          24.888636
Dodge Challenger       47.240909
AMC Javelin            46.007727
Camaro Z28             58.752727
Pontiac Firebird       57.379545
Fiat X1-9              18.928636
Porsche 914-2          24.779091
Lotus Europa           24.880273
Ford Pantera L         60.971818
Ferrari Dino           34.508182
Maserati Bora          63.155455
Volvo 142E             26.262727
dtype: float64
```

Median -

- Median is a middle value of a sorted distribution. The median splits the data in half.

In [10]:

```
mtcars.median()
```

Out[10]:

```
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
```

In [15]:

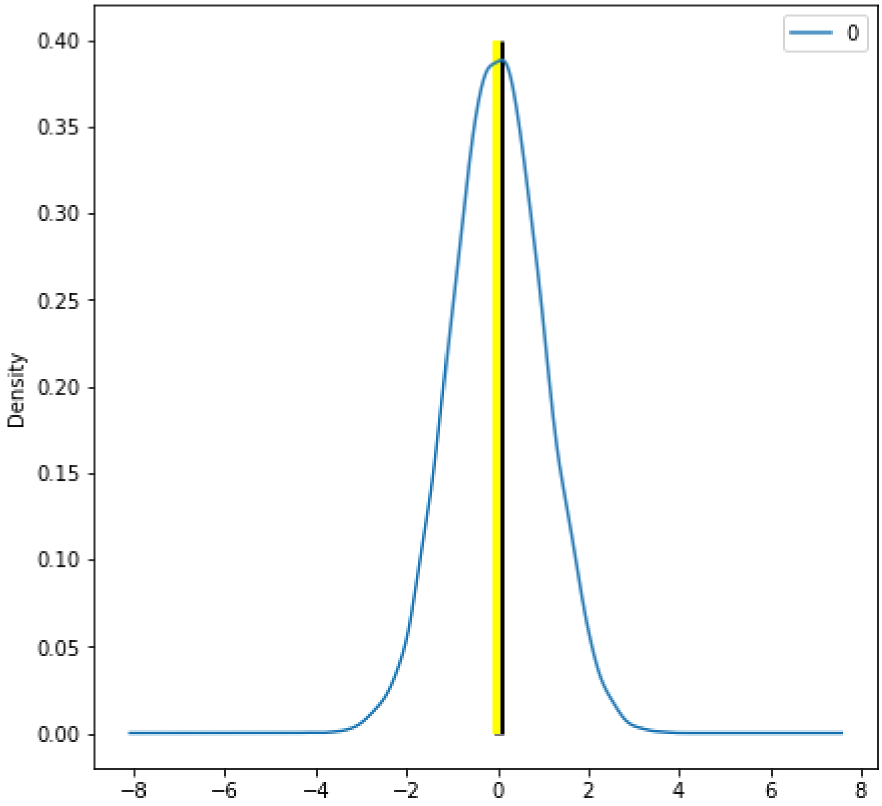
```
# Density Plot

norm_data = pd.DataFrame(np.random.normal(size=10000))
```

In [20]:

```
norm_data.plot(kind= 'density', figsize = (7,7))
plt.vlines(norm_data.mean(), ymin=0, ymax=0.4, linewidth = 5.0)
plt.vlines(norm_data.median(), ymin=0, ymax=0.4, linewidth = 4.0, color='yellow')
```

Out[20]:

```
<matplotlib.collections.LineCollection at 0x195066d6a58>
```



In this plot mean and median both are on the top of each other because of the symmetric destribution of of the data.

In Skewed distribution the mean tends to sift towards the skewness and median tries to resist the effect of the skewness.

In [34]:

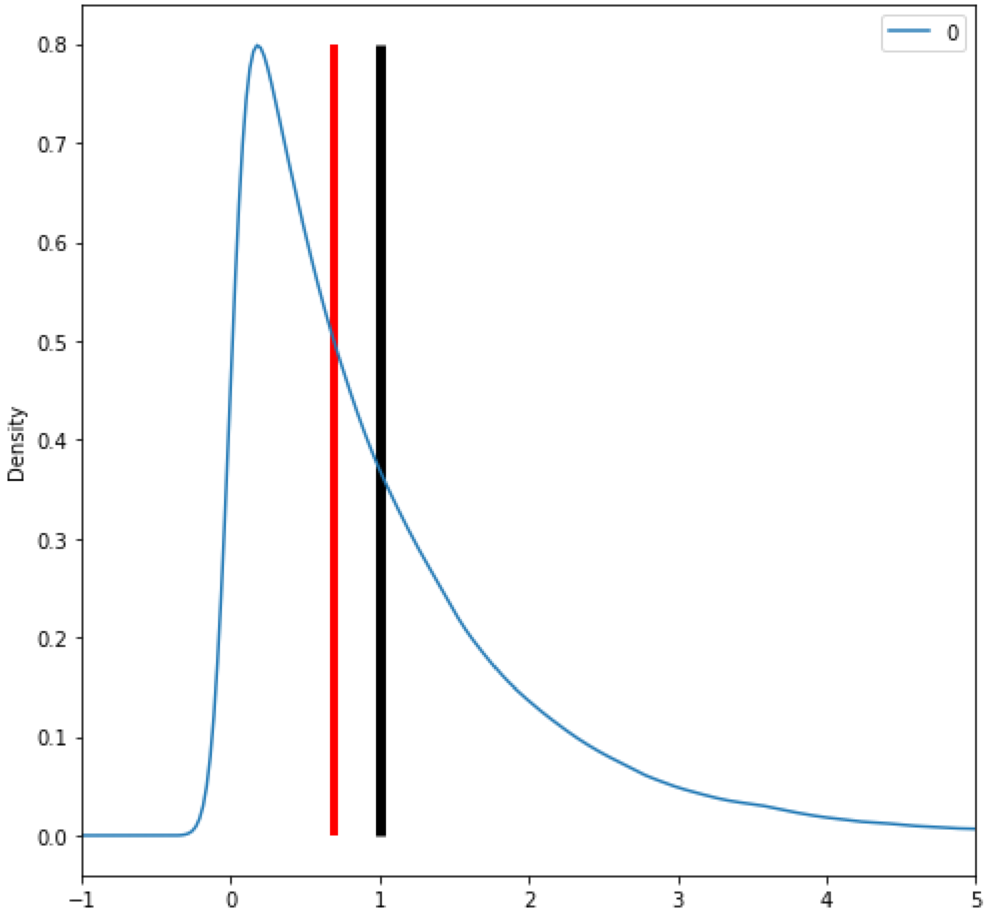
```
skewed_data = pd.DataFrame(np.random.exponential(size=100000))
```

In [37]:

```
skewed_data.plot(kind='density', figsize=(8,8), xlim = (-1, 5))
plt.vlines(skewed_data.mean(),ymin=0,ymax=0.8, linewidth=5)
plt.vlines(skewed_data.median(), ymin=0,ymax=0.8, linewidth = 4, color = 'Red')
```

Out[37]:

```
<matplotlib.collections.LineCollection at 0x19507097978>
```



In [40]:

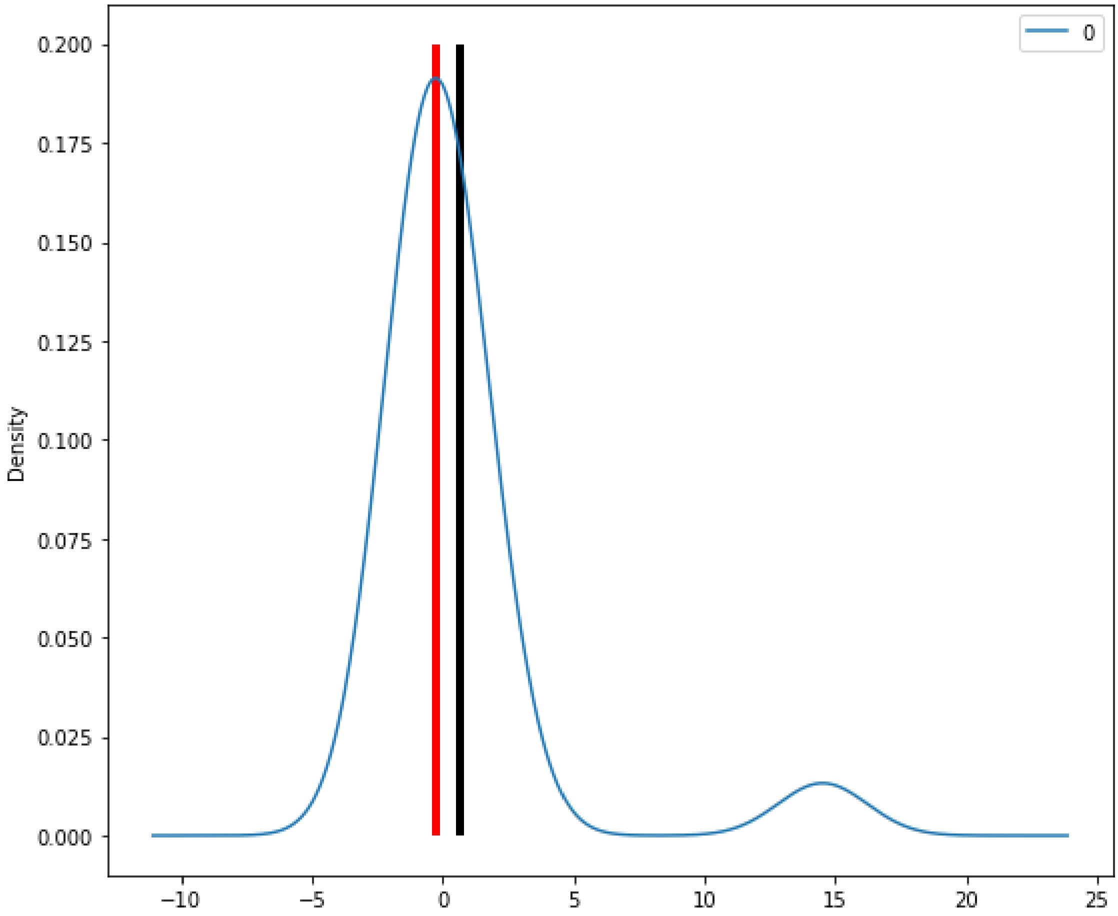
```
normal_data = np.random.normal(size= 50)
outlier_data = np.random.normal(15, size=3)
combined_data = pd.DataFrame(np.concatenate((normal_data, outlier_data), axis = 0))
```

In [42]:

```
combined_data.plot(kind = 'density', figsize=(9,8))
plt.vlines(combined_data.mean(), ymin=0, ymax=0.2, linewidth = 4 )
plt.vlines(combined_data.median(), ymin=0, ymax=0.2,linewidth = 4, color='red')
```

Out[42]:

```
<matplotlib.collections.LineCollection at 0x195078c0e48>
```



In [ ]:

```

```