In [3]:

```
#Import Python Libraries
import numpy as np
import pandas as pd
```

In [28]:

```
# Example of creating Pandas series :
mys1 = pd.Series( [-3,-1,1,3,5,8,9,11] )
print(s1)
```

```
0    -3
1    -1
2     1
3     3
4     5
5     8
6     9
7    11
dtype: int64
```

In [29]:

```
# View index values
print(mys1.index)
```

```
RangeIndex(start=0, stop=8, step=1)
```

In [32]:

```
# Creating Pandas series with index:
mys2 = pd.Series( np.random.randn(5), index=['a', 'b', 'k', 'd', 'z'] )
print(mys2)
```

```
a    0.028006
b   -1.010352
k   -1.087207
d   -1.235484
z   -0.931226
dtype: float64
```

In [33]:

```
# View index values
print(mys2.index)
```

```
Index(['a', 'b', 'k', 'd', 'z'], dtype='object')
```

In [34]:

```
# Create a Series from dictionary
data = {'pi': 3.1415, 'e': 2.71828}  # dictionary
print(data)
print()
mys3 = pd.Series( data )
print(mys3)
```

```
{'pi': 3.1415, 'e': 2.71828}

pi    3.14150
e     2.71828
dtype: float64
```

In [35]:

```
# reordering the elements
mys4 = pd.Series( data, index=['e', 'pi', 'tau'] )
print(mys4)
```

```
e      2.71828
pi     3.14150
tau        NaN
dtype: float64
```

In [36]:

```
print(mys1)
```

```
0    -3
1    -1
2     1
3     3
4     5
5     8
6     9
7    11
dtype: int64
```

In [37]:

```
print(mys1[:2]) # First 2 elements
```

```
0   -3
1   -1
dtype: int64
```

In [38]:

```
print( mys1[[4,2,0]])  # Elements out of order
```

```
4    5
2    1
0   -3
dtype: int64
```

In [39]:

```
# Series can be used as ndarray:
print("Median:" , mys1.median())
```

```
Median: 4.0
```

In [40]:

```
mys1[mys1 > 0]
```

Out[40]:

```
2     1
3     3
4     5
5     8
6     9
7    11
dtype: int64
```

In [41]:

```
# numpy functions can be used on series as usual:
mys1[mys1 < mys1.median()]
```

Out[41]:

```
0   -3
1   -1
2    1
3    3
dtype: int64
```

In [42]:

```
d1 =  pd.DataFrame( {'Name': pd.Series(['Alice','Bob','Chris']),
                      'Age': pd.Series([ 21,25,23]) } )
```

In [43]:

```
d1
```

Out[43]:

	Name	Age
0	Alice	21
1	Bob	25
2	Chris	23

In [44]:

```
d2 = pd.DataFrame( np.array([['Alice','Bob','Chris'], [21,25,23]]).T, columns=['Name','Age'
```

In [45]:

```
d2
```

Out[45]:

	Name	Age
0	Alice	21
1	Bob	25
2	Chris	23

In [46]:

```
#Add a new column:
d2['height'] = pd.Series([5.2,6.0,5.6])
d2
```

Out[46]:

	Name	Age	height
0	Alice	21	5.2
1	Bob	25	6.0
2	Chris	23	5.6

In [48]:

```
#Read csv file
df = pd.read_csv("C:/Users/ABHISHEK/Desktop/GITAM ML/GITAM ML LAB/redwinequality.csv")
```

In [49]:

```
df.head()
```

Out[49]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcoh
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9

In [50]:

```
df
```

Out[50]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56
5	7.4	0.660	0.00	1.8	0.075	13.0	40.0	0.99780	3.51	0.56
6	7.9	0.600	0.06	1.6	0.069	15.0	59.0	0.99640	3.30	0.46
7	7.3	0.650	0.00	1.2	0.065	15.0	21.0	0.99460	3.39	0.47
8	7.8	0.580	0.02	2.0	0.073	9.0	18.0	0.99680	3.36	0.57
9	7.5	0.500	0.36	6.1	0.071	17.0	102.0	0.99780	3.35	0.80
10	6.7	0.580	0.08	1.8	0.097	15.0	65.0	0.99590	3.28	0.54
11	7.5	0.500	0.36	6.1	0.071	17.0	102.0	0.99780	3.35	0.80
12	5.6	0.615	0.00	1.6	0.089	16.0	59.0	0.99430	3.58	0.52
13	7.8	0.610	0.29	1.6	0.114	9.0	29.0	0.99740	3.26	1.56
14	8.9	0.620	0.18	3.8	0.176	52.0	145.0	0.99860	3.16	0.88
15	8.9	0.620	0.19	3.9	0.170	51.0	148.0	0.99860	3.17	0.93
16	8.5	0.280	0.56	1.8	0.092	35.0	103.0	0.99690	3.30	0.75
17	8.1	0.560	0.28	1.7	0.368	16.0	56.0	0.99680	3.11	1.28
18	7.4	0.590	0.08	4.4	0.086	6.0	29.0	0.99740	3.38	0.50
19	7.9	0.320	0.51	1.8	0.341	17.0	56.0	0.99690	3.04	1.08
20	8.9	0.220	0.48	1.8	0.077	29.0	60.0	0.99680	3.39	0.53
21	7.6	0.390	0.31	2.3	0.082	23.0	71.0	0.99820	3.52	0.65
22	7.9	0.430	0.21	1.6	0.106	10.0	37.0	0.99660	3.17	0.91
23	8.5	0.490	0.11	2.3	0.084	9.0	67.0	0.99680	3.17	0.53
24	6.9	0.400	0.14	2.4	0.085	21.0	40.0	0.99680	3.43	0.63
25	6.3	0.390	0.16	1.4	0.080	11.0	23.0	0.99550	3.34	0.56
26	7.6	0.410	0.24	1.8	0.080	4.0	11.0	0.99620	3.28	0.59
27	7.9	0.430	0.21	1.6	0.106	10.0	37.0	0.99660	3.17	0.91
28	7.1	0.710	0.00	1.9	0.080	14.0	35.0	0.99720	3.47	0.55
29	7.8	0.645	0.00	2.0	0.082	8.0	16.0	0.99640	3.38	0.59
...	...	...	...	...	...	...	...	...	...	...
1569	6.2	0.510	0.14	1.9	0.056	15.0	34.0	0.99396	3.48	0.57
1570	6.4	0.360	0.53	2.2	0.230	19.0	35.0	0.99340	3.37	0.93

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates
1571	6.4	0.380	0.14	2.2	0.038	15.0	25.0	0.99514	3.44	0.65
1572	7.3	0.690	0.32	2.2	0.069	35.0	104.0	0.99632	3.33	0.51
1573	6.0	0.580	0.20	2.4	0.075	15.0	50.0	0.99467	3.58	0.67
1574	5.6	0.310	0.78	13.9	0.074	23.0	92.0	0.99677	3.39	0.48
1575	7.5	0.520	0.40	2.2	0.060	12.0	20.0	0.99474	3.26	0.64
1576	8.0	0.300	0.63	1.6	0.081	16.0	29.0	0.99588	3.30	0.78
1577	6.2	0.700	0.15	5.1	0.076	13.0	27.0	0.99622	3.54	0.60
1578	6.8	0.670	0.15	1.8	0.118	13.0	20.0	0.99540	3.42	0.67
1579	6.2	0.560	0.09	1.7	0.053	24.0	32.0	0.99402	3.54	0.60
1580	7.4	0.350	0.33	2.4	0.068	9.0	26.0	0.99470	3.36	0.60
1581	6.2	0.560	0.09	1.7	0.053	24.0	32.0	0.99402	3.54	0.60
1582	6.1	0.715	0.10	2.6	0.053	13.0	27.0	0.99362	3.57	0.50
1583	6.2	0.460	0.29	2.1	0.074	32.0	98.0	0.99578	3.33	0.62
1584	6.7	0.320	0.44	2.4	0.061	24.0	34.0	0.99484	3.29	0.80
1585	7.2	0.390	0.44	2.6	0.066	22.0	48.0	0.99494	3.30	0.84
1586	7.5	0.310	0.41	2.4	0.065	34.0	60.0	0.99492	3.34	0.85
1587	5.8	0.610	0.11	1.8	0.066	18.0	28.0	0.99483	3.55	0.66
1588	7.2	0.660	0.33	2.5	0.068	34.0	102.0	0.99414	3.27	0.78
1589	6.6	0.725	0.20	7.8	0.073	29.0	79.0	0.99770	3.29	0.54
1590	6.3	0.550	0.15	1.8	0.077	26.0	35.0	0.99314	3.32	0.82
1591	5.4	0.740	0.09	1.7	0.089	16.0	26.0	0.99402	3.67	0.56
1592	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75
1593	6.8	0.620	0.08	1.9	0.068	28.0	38.0	0.99651	3.42	0.82
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66

1599 rows × 12 columns

In [51]:

```
type(df)
```

Out[51]:

```
pandas.core.frame.DataFrame
```

In [52]:

```
df.columns
```

Out[52]:

```
Index(['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',
       'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'densit
y',
       'pH', 'sulphates', 'alcohol', 'quality'],
      dtype='object')
```

In [53]:

```
df.dtypes
```

Out[53]:

```
fixed acidity           float64
volatile acidity        float64
citric acid             float64
residual sugar          float64
chlorides               float64
free sulfur dioxide     float64
total sulfur dioxide    float64
density                 float64
pH                      float64
sulphates               float64
alcohol                 float64
quality                   int64
dtype: object
```

In [54]:

```
df['quality'].dtype
```

Out[54]:

```
dtype('int64')
```

In [55]:

```
df.axes
```

Out[55]:

```
[RangeIndex(start=0, stop=1599, step=1),
 Index(['fixed acidity', 'volatile acidity', 'citric acid', 'residual suga
r',
        'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'densit
y',
        'pH', 'sulphates', 'alcohol', 'quality'],
       dtype='object')]
```

In [56]:

```
df.ndim
```

Out[56]:

```
2
```

In [57]:

```
df.shape
```

Out[57]:

```
(1599, 12)
```

In [58]:

```
df.size
```

Out[58]:

```
19188
```

In [59]:

```
df.describe()
```

Out[59]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total su dio:
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.00(
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.00(
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.00(
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.00(
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.00(
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.00(

In [60]:

```
df.mean()
```

Out[60]:

```
fixed acidity            8.319637
volatile acidity         0.527821
citric acid              0.270976
residual sugar           2.538806
chlorides                0.087467
free sulfur dioxide     15.874922
total sulfur dioxide    46.467792
density                  0.996747
pH                       3.311113
sulphates                0.658149
alcohol                 10.422983
quality                  5.636023
dtype: float64
```

In [62]:

```
df['quality'].head()
```

Out[62]:

```
0    5
1    5
2    5
3    6
4    5
Name: quality, dtype: int64
```

In [63]:

```
#Extract a column name (method 2)
df.quality.head()
```

Out[63]:

```
0    5
1    5
2    5
3    6
4    5
Name: quality, dtype: int64
```

In [66]:

```
df_rank = df.groupby('chlorides')
```

In [67]:

```
df_rank.mean()
```

Out[67]:

chlorides	fixed acidity	volatile acidity	citric acid	residual sugar	free sulfur dioxide	total sulfur dioxide	density	pH
0.012	6.700000	0.280000	0.280000	2.400000	36.000000	100.000000	0.990640	3.260000
0.034	5.100000	0.470000	0.020000	1.300000	18.000000	44.000000	0.992100	3.900000
0.038	7.850000	0.375000	0.290000	1.900000	18.000000	33.500000	0.995200	3.340000
0.039	7.575000	0.605000	0.447500	1.850000	8.000000	33.500000	0.994765	3.437500
0.041	6.100000	0.661250	0.097500	1.625000	13.250000	45.000000	0.993120	3.742500
0.042	6.366667	0.426667	0.206667	1.733333	8.333333	41.333333	0.994073	3.430000
0.043	6.900000	0.450000	0.110000	2.400000	6.000000	12.000000	0.993540	3.300000
0.044	6.160000	0.493000	0.132000	1.820000	20.200000	76.200000	0.992662	3.480000
0.045	5.725000	0.682500	0.095000	1.650000	32.250000	89.000000	0.993627	3.530000
0.046	6.125000	0.463750	0.320000	2.650000	19.250000	71.500000	0.993160	3.457500
0.047	8.025000	0.637500	0.292500	1.900000	14.750000	44.000000	0.995580	3.337500
0.048	5.925000	0.471250	0.133750	2.012500	17.250000	52.375000	0.992787	3.520000
0.049	6.700000	0.382500	0.200000	1.700000	22.250000	71.625000	0.993180	3.398750
0.050	7.100000	0.497500	0.273333	3.233333	18.833333	79.000000	0.994055	3.359167
0.051	6.500000	0.390000	0.230000	8.300000	28.000000	91.000000	0.995200	3.440000
0.052	8.030000	0.444000	0.292000	2.110000	7.500000	20.900000	0.994675	3.338000
0.053	7.160000	0.558000	0.188000	2.000000	20.400000	46.400000	0.995272	3.352000
0.054	7.669231	0.474615	0.241538	2.823077	11.384615	42.769231	0.995686	3.395385
0.055	8.012500	0.461250	0.223750	2.393750	12.125000	21.875000	0.994631	3.315000
0.056	7.522222	0.408333	0.256667	2.011111	9.777778	26.000000	0.994790	3.366667
0.057	7.320000	0.485000	0.166000	1.920000	17.000000	31.800000	0.994765	3.348000
0.058	8.035714	0.537143	0.228571	1.925000	13.285714	31.714286	0.995383	3.315714
0.059	8.082353	0.538235	0.305882	2.029412	13.647059	25.941176	0.995532	3.320000
0.060	8.037500	0.492812	0.283750	2.856250	16.062500	42.000000	0.995634	3.334375
0.061	7.890909	0.537273	0.254545	2.181818	16.000000	29.909091	0.995879	3.338182
0.062	7.754167	0.432500	0.270833	2.208333	14.833333	29.708333	0.995902	3.315000
0.063	8.222727	0.452500	0.284545	2.127273	17.136364	34.727273	0.996024	3.316364
0.064	8.025000	0.515000	0.199500	2.400000	19.450000	34.500000	0.995603	3.333000
0.065	8.008696	0.540000	0.218696	2.073913	14.086957	29.260870	0.995677	3.366957
0.066	8.309375	0.446094	0.283125	2.278125	14.140625	38.000000	0.996319	3.341875
...	...	...	...	...	...	...	...	...
0.222	8.300000	0.490000	0.360000	1.800000	6.000000	16.000000	0.998000	3.180000

chlorides	fixed acidity	volatile acidity	citric acid	residual sugar	free sulfur dioxide	total sulfur dioxide	density	pH
0.226	8.700000	0.700000	0.240000	2.500000	5.000000	15.000000	0.999100	3.320000
0.230	6.400000	0.360000	0.530000	2.200000	19.000000	35.000000	0.993400	3.370000
0.235	6.900000	0.630000	0.330000	6.700000	66.000000	115.000000	0.997870	3.220000
0.236	7.600000	0.490000	0.260000	1.600000	10.000000	88.000000	0.996800	3.110000
0.241	6.900000	0.635000	0.170000	2.400000	6.000000	18.000000	0.996100	3.400000
0.243	6.900000	0.765000	0.180000	2.400000	5.500000	48.000000	0.996120	3.400000
0.250	9.800000	0.500000	0.490000	2.600000	5.000000	20.000000	0.999000	3.310000
0.263	8.900000	0.635000	0.370000	1.700000	5.000000	62.000000	0.997100	3.000000
0.267	6.800000	0.815000	0.000000	1.200000	16.000000	29.000000	0.994710	3.320000
0.270	8.900000	0.610000	0.490000	2.000000	23.000000	110.000000	0.997200	3.120000
0.332	7.500000	0.490000	0.200000	2.600000	8.000000	14.000000	0.996800	3.210000
0.337	8.900000	0.590000	0.500000	2.000000	27.000000	81.000000	0.996400	3.040000
0.341	7.900000	0.320000	0.510000	1.800000	17.000000	56.000000	0.996900	3.040000
0.343	11.000000	0.200000	0.480000	2.000000	6.000000	18.000000	0.997900	3.300000
0.358	7.700000	0.270000	0.680000	3.500000	5.000000	10.000000	0.997200	3.250000
0.360	7.500000	0.705000	0.240000	1.800000	15.000000	63.000000	0.996400	3.000000
0.368	8.100000	0.560000	0.280000	1.700000	16.000000	56.000000	0.996800	3.110000
0.369	8.500000	0.440000	0.500000	1.900000	15.000000	38.000000	0.996340	3.010000
0.387	9.500000	0.550000	0.660000	2.300000	12.000000	37.000000	0.998200	3.170000
0.401	7.300000	0.670000	0.260000	1.800000	16.000000	51.000000	0.996900	3.160000
0.403	8.600000	0.635000	0.680000	1.800000	19.000000	56.000000	0.996320	3.020000
0.413	8.400000	0.370000	0.530000	1.800000	9.000000	26.000000	0.997900	3.060000
0.414	8.800000	0.610000	0.635000	1.550000	17.000000	54.500000	0.996770	2.965000
0.415	8.400000	0.680000	0.566667	1.700000	12.666667	54.666667	0.996340	3.030000
0.422	8.600000	0.490000	0.510000	2.000000	16.000000	62.000000	0.997900	3.030000
0.464	7.800000	0.430000	0.700000	1.900000	22.000000	67.000000	0.997400	3.130000
0.467	7.800000	0.410000	0.680000	1.700000	18.000000	69.000000	0.997300	3.080000
0.610	9.200000	0.520000	1.000000	3.400000	32.000000	69.000000	0.999600	2.740000
0.611	7.700000	0.410000	0.760000	1.800000	8.000000	45.000000	0.996800	3.060000

153 rows × 11 columns

In [68]:

```
df.groupby('density').mean()
```

Out[68]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	pH
density								
0.99007	8.000000	0.180000	0.370000	0.900000	0.049000	36.000000	109.000000	2.890000
0.99020	5.000000	0.400000	0.500000	4.300000	0.046000	29.000000	80.000000	3.490000
0.99064	6.700000	0.280000	0.280000	2.400000	0.012000	36.000000	100.000000	3.260000
0.99080	5.500000	0.490000	0.030000	1.800000	0.044000	28.000000	87.000000	3.500000
0.99084	5.000000	0.380000	0.010000	1.600000	0.048000	26.000000	60.000000	3.700000
0.99120	6.100000	0.210000	0.400000	1.400000	0.066000	40.500000	165.000000	3.250000
0.99150	5.800000	0.290000	0.260000	1.700000	0.063000	3.000000	11.000000	3.390000
0.99154	4.900000	0.420000	0.000000	2.100000	0.048000	16.000000	42.000000	3.710000
0.99157	5.100000	0.420000	0.000000	1.800000	0.044000	18.000000	88.000000	3.680000
0.99160	5.200000	0.340000	0.000000	1.800000	0.050000	27.000000	63.000000	3.680000
0.99162	6.400000	0.690000	0.000000	1.650000	0.055000	7.000000	12.000000	3.470000
0.99170	5.000000	0.420000	0.240000	2.000000	0.060000	19.000000	50.000000	3.720000
0.99182	5.300000	0.470000	0.110000	2.200000	0.048000	16.000000	89.000000	3.540000
0.99191	5.700000	0.600000	0.000000	1.400000	0.063000	11.000000	18.000000	3.450000
0.99210	5.100000	0.470000	0.020000	1.300000	0.034000	18.000000	44.000000	3.900000
0.99220	6.450000	0.430000	0.245000	1.550000	0.062500	25.000000	91.000000	3.285000
0.99235	7.900000	0.540000	0.340000	2.500000	0.076000	8.000000	17.000000	3.200000
0.99236	6.000000	0.540000	0.060000	1.800000	0.050000	38.000000	89.000000	3.300000
0.99240	5.366667	0.731667	0.103333	1.566667	0.044333	13.666667	94.000000	3.530000
0.99242	7.100000	0.750000	0.010000	2.200000	0.059000	11.000000	18.000000	3.390000
0.99252	7.100000	0.390000	0.120000	2.100000	0.065000	14.000000	24.000000	3.300000
0.99256	5.600000	0.660000	0.000000	2.500000	0.066000	7.000000	15.000000	3.520000
0.99258	5.333333	0.695000	0.016667	1.600000	0.051667	17.000000	39.000000	3.860000
0.99264	5.100000	0.585000	0.000000	1.700000	0.044000	14.000000	86.000000	3.560000
0.99270	5.200000	0.480000	0.040000	1.600000	0.054000	19.000000	106.000000	3.540000
0.99280	6.800000	0.360000	0.320000	1.800000	0.067000	4.000000	8.000000	3.360000
0.99286	8.000000	0.600000	0.080000	2.600000	0.056000	3.000000	7.000000	3.220000
0.99290	6.500000	0.510000	0.150000	3.000000	0.064000	12.000000	27.000000	3.330000
0.99292	6.000000	0.490000	0.000000	2.300000	0.068000	15.000000	33.000000	3.580000
0.99294	6.400000	0.475000	0.240000	2.550000	0.062000	14.500000	59.000000	3.345000
...	...	...	...	...	...	...	...	...
0.99970	10.350000	0.593750	0.421250	3.100000	0.089000	19.250000	49.500000	3.282500

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	pH
density								
0.99974	11.700000	0.450000	0.630000	2.200000	0.073000	7.000000	23.000000	3.210000
0.99975	9.300000	0.610000	0.260000	3.400000	0.090000	25.000000	87.000000	3.240000
0.99976	9.100000	0.660000	0.150000	3.200000	0.097000	9.000000	59.000000	3.280000
0.99980	9.820000	0.542000	0.420000	3.280000	0.091900	13.700000	57.400000	3.259000
0.99990	12.500000	0.560000	0.490000	2.400000	0.064000	5.000000	27.000000	3.080000
1.00000	11.580000	0.481500	0.525000	2.430000	0.086900	10.600000	28.900000	3.137000
1.00005	15.000000	0.210000	0.440000	2.200000	0.075000	10.000000	24.000000	3.070000
1.00010	10.700000	0.478750	0.450000	2.850000	0.083750	18.250000	63.750000	3.187500
1.00012	12.900000	0.500000	0.550000	2.800000	0.072000	7.000000	24.000000	3.090000
1.00015	9.900000	0.590000	0.070000	3.400000	0.102000	32.000000	71.000000	3.310000
1.00020	10.880000	0.495000	0.424000	3.090000	0.093500	19.000000	45.000000	3.276000
1.00024	11.600000	0.410000	0.580000	2.800000	0.096000	25.000000	101.000000	3.130000
1.00025	8.100000	0.870000	0.000000	3.300000	0.096000	26.000000	61.000000	3.600000
1.00030	11.500000	0.400000	0.495000	3.150000	0.074000	14.500000	42.000000	3.290000
1.00040	11.555556	0.450000	0.505556	2.800000	0.089111	11.555556	36.444444	3.181111
1.00060	12.150000	0.436667	0.498333	2.550000	0.087500	8.166667	26.833333	3.151667
1.00080	12.766667	0.460000	0.660000	2.333333	0.084000	14.666667	53.666667	3.090000
1.00100	10.850000	0.645000	0.440000	3.700000	0.104000	21.333333	90.833333	3.295000
1.00140	12.700000	0.442500	0.540000	4.200000	0.089333	15.500000	59.166667	3.116667
1.00150	10.000000	0.490000	0.200000	11.000000	0.071000	13.000000	50.000000	3.160000
1.00180	13.500000	0.530000	0.790000	4.800000	0.120000	23.000000	77.000000	3.180000
1.00210	13.000000	0.470000	0.490000	4.300000	0.085000	6.000000	47.000000	3.300000
1.00220	12.800000	0.615000	0.660000	5.800000	0.083000	7.000000	42.000000	3.070000
1.00242	9.900000	0.500000	0.500000	13.800000	0.205000	48.000000	82.000000	3.160000
1.00260	10.350000	0.465000	0.380000	10.300000	0.089500	8.500000	44.000000	3.220000
1.00289	10.700000	0.900000	0.340000	6.600000	0.112000	23.000000	99.000000	3.220000
1.00315	15.533333	0.645000	0.490000	4.200000	0.095000	10.000000	23.000000	2.920000
1.00320	15.600000	0.685000	0.760000	3.700000	0.100000	6.000000	43.000000	2.950000
1.00369	10.200000	0.540000	0.370000	15.400000	0.214000	55.000000	95.000000	3.180000

436 rows × 11 columns

In [69]:

```
df.groupby('density')['chlorides'].mean()
```

Out[69]:

```
density
0.99007    0.049000
0.99020    0.046000
0.99064    0.012000
0.99080    0.044000
0.99084    0.048000
0.99120    0.066000
0.99150    0.063000
0.99154    0.048000
0.99157    0.044000
0.99160    0.050000
0.99162    0.055000
0.99170    0.060000
0.99182    0.048000
0.99191    0.063000
0.99210    0.034000
0.99220    0.062500
0.99235    0.076000
0.99236    0.050000
0.99240    0.044333
0.99242    0.059000
0.99252    0.065000
0.99256    0.066000
0.99258    0.051667
0.99264    0.044000
0.99270    0.054000
0.99280    0.067000
0.99286    0.056000
0.99290    0.064000
0.99292    0.068000
0.99294    0.062000
                ...
0.99970    0.089000
0.99974    0.073000
0.99975    0.090000
0.99976    0.097000
0.99980    0.091900
0.99990    0.064000
1.00000    0.086900
1.00005    0.075000
1.00010    0.083750
1.00012    0.072000
1.00015    0.102000
1.00020    0.093500
1.00024    0.096000
1.00025    0.096000
1.00030    0.074000
1.00040    0.089111
1.00060    0.087500
1.00080    0.084000
1.00100    0.104000
1.00140    0.089333
1.00150    0.071000
1.00180    0.120000
1.00210    0.085000
```

```
1.00220    0.083000
1.00242    0.205000
1.00260    0.089500
1.00289    0.112000
1.00315    0.095000
1.00320    0.100000
1.00369    0.214000
Name: chlorides, Length: 436, dtype: float64
```

In [70]:

```
df.groupby('density')[['chlorides']].mean().round(2)
```

Out[70]:

	chlorides
density	
0.99007	0.05
0.99020	0.05
0.99064	0.01
0.99080	0.04
0.99084	0.05
0.99120	0.07
0.99150	0.06
0.99154	0.05
0.99157	0.04
0.99160	0.05
0.99162	0.06
0.99170	0.06
0.99182	0.05
0.99191	0.06
0.99210	0.03
0.99220	0.06
0.99235	0.08
0.99236	0.05
0.99240	0.04
0.99242	0.06
0.99252	0.06
0.99256	0.07
0.99258	0.05
0.99264	0.04
0.99270	0.05
0.99280	0.07
0.99286	0.06
0.99290	0.06
0.99292	0.07
0.99294	0.06
...	...
0.99970	0.09
0.99974	0.07

	chlorides
density	
0.99975	0.09
0.99976	0.10
0.99980	0.09
0.99990	0.06
1.00000	0.09
1.00005	0.08
1.00010	0.08
1.00012	0.07
1.00015	0.10
1.00020	0.09
1.00024	0.10
1.00025	0.10
1.00030	0.07
1.00040	0.09
1.00060	0.09
1.00080	0.08
1.00100	0.10
1.00140	0.09
1.00150	0.07
1.00180	0.12
1.00210	0.08
1.00220	0.08
1.00242	0.20
1.00260	0.09
1.00289	0.11
1.00315	0.10
1.00320	0.10
1.00369	0.21

436 rows × 1 columns

In [74]:

```
df.groupby(['density','density'])[['pH']].mean()
#case sensitive
```

Out[74]:

		pH
density	**density**	
0.99007	**0.99007**	2.890000
0.99020	**0.99020**	3.490000
0.99064	**0.99064**	3.260000
0.99080	**0.99080**	3.500000
0.99084	**0.99084**	3.700000
0.99120	**0.99120**	3.250000
0.99150	**0.99150**	3.390000
0.99154	**0.99154**	3.710000
0.99157	**0.99157**	3.680000
0.99160	**0.99160**	3.680000
0.99162	**0.99162**	3.470000
0.99170	**0.99170**	3.720000
0.99182	**0.99182**	3.540000
0.99191	**0.99191**	3.450000
0.99210	**0.99210**	3.900000
0.99220	**0.99220**	3.285000
0.99235	**0.99235**	3.200000
0.99236	**0.99236**	3.300000
0.99240	**0.99240**	3.530000
0.99242	**0.99242**	3.390000
0.99252	**0.99252**	3.300000
0.99256	**0.99256**	3.520000
0.99258	**0.99258**	3.860000
0.99264	**0.99264**	3.560000
0.99270	**0.99270**	3.540000
0.99280	**0.99280**	3.360000
0.99286	**0.99286**	3.220000
0.99290	**0.99290**	3.330000
0.99292	**0.99292**	3.580000
0.99294	**0.99294**	3.345000
...	**...**	...
0.99970	**0.99970**	3.282500
0.99974	**0.99974**	3.210000

		pH
density	**density**	
0.99975	**0.99975**	3.240000
0.99976	**0.99976**	3.280000
0.99980	**0.99980**	3.259000
0.99990	**0.99990**	3.080000
1.00000	**1.00000**	3.137000
1.00005	**1.00005**	3.070000
1.00010	**1.00010**	3.187500
1.00012	**1.00012**	3.090000
1.00015	**1.00015**	3.310000
1.00020	**1.00020**	3.276000
1.00024	**1.00024**	3.130000
1.00025	**1.00025**	3.600000
1.00030	**1.00030**	3.290000
1.00040	**1.00040**	3.181111
1.00060	**1.00060**	3.151667
1.00080	**1.00080**	3.090000
1.00100	**1.00100**	3.295000
1.00140	**1.00140**	3.116667
1.00150	**1.00150**	3.160000
1.00180	**1.00180**	3.180000
1.00210	**1.00210**	3.300000
1.00220	**1.00220**	3.070000
1.00242	**1.00242**	3.160000
1.00260	**1.00260**	3.220000
1.00289	**1.00289**	3.220000
1.00315	**1.00315**	2.920000
1.00320	**1.00320**	2.950000
1.00369	**1.00369**	3.180000

436 rows × 1 columns

In [76]:

```
df.groupby(['density','chlorides'],sort=False)[['pH']].mean()
```

Out[76]:

		pH
density	**chlorides**	
0.99780	**0.076**	3.500000
0.99680	**0.098**	3.200000
0.99700	**0.092**	3.260000
0.99800	**0.075**	3.290000
0.99780	**0.075**	3.325000
0.99640	**0.069**	3.300000
0.99460	**0.065**	3.390000
0.99680	**0.073**	3.360000
0.99780	**0.071**	3.350000
0.99590	**0.097**	3.310000
0.99430	**0.089**	3.580000
0.99740	**0.114**	3.260000
0.99860	**0.176**	3.160000
	0.170	3.170000
0.99690	**0.092**	3.300000
0.99680	**0.368**	3.110000
0.99740	**0.086**	3.380000
0.99690	**0.341**	3.040000
0.99680	**0.077**	3.390000
0.99820	**0.082**	3.520000
0.99660	**0.106**	3.170000
0.99680	**0.084**	3.255000
	0.085	3.405000
0.99550	**0.080**	3.393333
0.99620	**0.080**	3.260000
0.99720	**0.080**	3.307500
0.99640	**0.082**	3.420000
0.99580	**0.089**	3.350000
0.99660	**0.105**	3.460000
	0.083	3.345000
...	**...**	...
0.99489	**0.061**	3.390000
0.99494	**0.059**	3.340000

		pH
density	**chlorides**	
0.99629	**0.077**	3.340000
0.99340	**0.230**	3.370000
0.99514	**0.038**	3.440000
0.99632	**0.069**	3.330000
0.99467	**0.075**	3.580000
0.99677	**0.074**	3.390000
0.99474	**0.060**	3.260000
0.99588	**0.081**	3.300000
0.99622	**0.076**	3.540000
0.99540	**0.118**	3.420000
0.99402	**0.053**	3.540000
0.99470	**0.068**	3.360000
0.99362	**0.053**	3.570000
0.99578	**0.074**	3.330000
0.99484	**0.061**	3.290000
0.99494	**0.066**	3.300000
0.99492	**0.065**	3.340000
0.99483	**0.066**	3.550000
0.99414	**0.068**	3.270000
0.99770	**0.073**	3.290000
0.99314	**0.077**	3.320000
0.99402	**0.089**	3.670000
0.99574	**0.076**	3.420000
0.99651	**0.068**	3.420000
0.99490	**0.090**	3.450000
0.99512	**0.062**	3.520000
0.99547	**0.075**	3.570000
0.99549	**0.067**	3.390000

1241 rows × 1 columns

In [77]:

```
df_sub = df[ df['pH'] > 3.2]
df_sub.head()
```

Out[77]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcoh
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9
5	7.4	0.66	0.00	1.8	0.075	13.0	40.0	0.9978	3.51	0.56	9
6	7.9	0.60	0.06	1.6	0.069	15.0	59.0	0.9964	3.30	0.46	9

In [78]:

```
df_sub.axes
```

Out[78]:

```
[Int64Index([   0,    2,    4,    5,    6,    7,    8,    9,   10,   11,
             ...
             1589, 1590, 1591, 1592, 1593, 1594, 1595, 1596, 1597, 1598],
            dtype='int64', length=1211),
 Index(['fixed acidity', 'volatile acidity', 'citric acid', 'residual suga
r',
        'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'densit
y',
        'pH', 'sulphates', 'alcohol', 'quality'],
       dtype='object')]
```

In [79]:

```
df_sub.shape
```

Out[79]:

```
(1211, 12)
```

In [80]:

```
df_w = df[ df['quality'] == 5]
df_w.head()
```

Out[80]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcoh
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9
5	7.4	0.66	0.00	1.8	0.075	13.0	40.0	0.9978	3.51	0.56	9

In [81]:

```
df_w.shape
```

Out[81]:

```
(681, 12)
```

In [83]:

```
df1 = df['pH']
```

In [84]:

```
type(df1)
```

Out[84]:

```
pandas.core.series.Series
```

In [85]:

```
type(df)
```

Out[85]:

```
pandas.core.frame.DataFrame
```

In [86]:

```
#Look at the first few elements of the output
df1.head()
```

Out[86]:

```
0    3.51
1    3.20
2    3.26
3    3.16
4    3.51
Name: pH, dtype: float64
```

In [87]:

```
df2 = df[['pH']]
```

In [88]:

```
type(df2)
```

Out[88]:

```
pandas.core.frame.DataFrame
```

In [89]:

```
df[0:10]
```

Out[89]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcoh
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9
5	7.4	0.66	0.00	1.8	0.075	13.0	40.0	0.9978	3.51	0.56	9
6	7.9	0.60	0.06	1.6	0.069	15.0	59.0	0.9964	3.30	0.46	9
7	7.3	0.65	0.00	1.2	0.065	15.0	21.0	0.9946	3.39	0.47	10
8	7.8	0.58	0.02	2.0	0.073	9.0	18.0	0.9968	3.36	0.57	9
9	7.5	0.50	0.36	6.1	0.071	17.0	102.0	0.9978	3.35	0.80	10

In [90]:

```
df.loc[10:20,['quality', 'density','pH']]
```

Out[90]:

	quality	density	pH
10	5	0.9959	3.28
11	5	0.9978	3.35
12	5	0.9943	3.58
13	5	0.9974	3.26
14	5	0.9986	3.16
15	5	0.9986	3.17
16	7	0.9969	3.30
17	5	0.9968	3.11
18	4	0.9974	3.38
19	6	0.9969	3.04
20	6	0.9968	3.39

In [91]:

```
df_sub.head(10)
```

Out[91]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alco
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	
5	7.4	0.66	0.00	1.8	0.075	13.0	40.0	0.9978	3.51	0.56	
6	7.9	0.60	0.06	1.6	0.069	15.0	59.0	0.9964	3.30	0.46	
7	7.3	0.65	0.00	1.2	0.065	15.0	21.0	0.9946	3.39	0.47	1
8	7.8	0.58	0.02	2.0	0.073	9.0	18.0	0.9968	3.36	0.57	
9	7.5	0.50	0.36	6.1	0.071	17.0	102.0	0.9978	3.35	0.80	1
10	6.7	0.58	0.08	1.8	0.097	15.0	65.0	0.9959	3.28	0.54	
11	7.5	0.50	0.36	6.1	0.071	17.0	102.0	0.9978	3.35	0.80	1

In [92]:

```
df_sub.loc[0:10,['quality','density','alcohol','pH']]
```

Out[92]:

	quality	density	alcohol	pH
0	5	0.9978	9.4	3.51
2	5	0.9970	9.8	3.26
4	5	0.9978	9.4	3.51
5	5	0.9978	9.4	3.51
6	5	0.9964	9.4	3.30
7	7	0.9946	10.0	3.39
8	7	0.9968	9.5	3.36
9	5	0.9978	10.5	3.35
10	5	0.9959	9.2	3.28

In [93]:

```
df_sub.iloc[0:10, [0,3,4,5]]
```

Out[93]:

	fixed acidity	residual sugar	chlorides	free sulfur dioxide
0	7.4	1.9	0.076	11.0
2	7.8	2.3	0.092	15.0
4	7.4	1.9	0.076	11.0
5	7.4	1.8	0.075	13.0
6	7.9	1.6	0.069	15.0
7	7.3	1.2	0.065	15.0
8	7.8	2.0	0.073	9.0
9	7.5	6.1	0.071	17.0
10	6.7	1.8	0.097	15.0
11	7.5	6.1	0.071	17.0

In [94]:

```
df_sorted = df.sort_values(by = 'quality')
df_sorted.head()
```

Out[94]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	al
1478	7.1	0.875	0.05	5.7	0.082	3.0	14.0	0.99808	3.40	0.52	
832	10.4	0.440	0.42	1.5	0.145	34.0	48.0	0.99832	3.38	0.86	
899	8.3	1.020	0.02	3.4	0.084	6.0	11.0	0.99892	3.48	0.49	
1374	6.8	0.815	0.00	1.2	0.267	16.0	29.0	0.99471	3.32	0.51	
459	11.6	0.580	0.66	2.2	0.074	10.0	47.0	1.00080	3.25	0.57	

In [95]:

```
df.sort_values(by = 'quality', ascending = False, inplace = True)
df.head()
```

Out[95]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	al
495	10.7	0.35	0.53	2.6	0.070	5.0	16.0	0.99720	3.15	0.65	
1403	7.2	0.33	0.33	1.7	0.061	3.0	13.0	0.99600	3.23	1.10	
390	5.6	0.85	0.05	1.4	0.045	12.0	88.0	0.99240	3.56	0.82	
1061	9.1	0.40	0.50	1.8	0.071	7.0	16.0	0.99462	3.21	0.69	
1202	8.6	0.42	0.39	1.8	0.068	6.0	12.0	0.99516	3.35	0.69	

In [96]:

```
df.head()
```

Out[96]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	al
495	10.7	0.35	0.53	2.6	0.070	5.0	16.0	0.99720	3.15	0.65	
1403	7.2	0.33	0.33	1.7	0.061	3.0	13.0	0.99600	3.23	1.10	
390	5.6	0.85	0.05	1.4	0.045	12.0	88.0	0.99240	3.56	0.82	
1061	9.1	0.40	0.50	1.8	0.071	7.0	16.0	0.99462	3.21	0.69	
1202	8.6	0.42	0.39	1.8	0.068	6.0	12.0	0.99516	3.35	0.69	

In [97]:

```
df.sort_index(axis=0, ascending = True)
df.head()
```

Out[97]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	al
495	10.7	0.35	0.53	2.6	0.070	5.0	16.0	0.99720	3.15	0.65	
1403	7.2	0.33	0.33	1.7	0.061	3.0	13.0	0.99600	3.23	1.10	
390	5.6	0.85	0.05	1.4	0.045	12.0	88.0	0.99240	3.56	0.82	
1061	9.1	0.40	0.50	1.8	0.071	7.0	16.0	0.99462	3.21	0.69	
1202	8.6	0.42	0.39	1.8	0.068	6.0	12.0	0.99516	3.35	0.69	

In [98]:

```
df.sort_index(axis=0, ascending = True, inplace= True)
df.head()
```

Out[98]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcoh
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9

In [99]:

```
df_sorted = df.sort_values(by = ['quality', 'pH'], ascending = [True,False])
df_sorted.head(10)
```

Out[99]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	al
690	7.4	1.185	0.00	4.25	0.097	5.0	14.0	0.99660	3.63	0.54	
1505	6.7	0.760	0.02	1.80	0.078	6.0	12.0	0.99600	3.55	0.63	
1299	7.6	1.580	0.00	2.10	0.137	5.0	9.0	0.99476	3.50	0.40	
899	8.3	1.020	0.02	3.40	0.084	6.0	11.0	0.99892	3.48	0.49	
1478	7.1	0.875	0.05	5.70	0.082	3.0	14.0	0.99808	3.40	0.52	
832	10.4	0.440	0.42	1.50	0.145	34.0	48.0	0.99832	3.38	0.86	
1374	6.8	0.815	0.00	1.20	0.267	16.0	29.0	0.99471	3.32	0.51	
1469	7.3	0.980	0.05	2.10	0.061	20.0	49.0	0.99705	3.31	0.55	
459	11.6	0.580	0.66	2.20	0.074	10.0	47.0	1.00080	3.25	0.57	
517	10.4	0.610	0.49	2.10	0.200	5.0	16.0	0.99940	3.16	0.63	

In [100]:

```
data=[[1,2,3],[3,4,np.nan],[5,6,np.nan],[np.nan,np.nan,np.nan]]
df1=pd.DataFrame(data, columns=['one', 'two', 'three'])
df1
```

Out[100]:

	one	two	three
0	1.0	2.0	3.0
1	3.0	4.0	NaN
2	5.0	6.0	NaN
3	NaN	NaN	NaN

In [101]:

```
df1.isnull().sum()
```

Out[101]:

```
one      1
two      1
three    3
dtype: int64
```

In [102]:

```
df1.count()
```

Out[102]:

```
one      3
two      3
three    1
dtype: int64
```

In [103]:

```
df1.dropna(axis=1) # Dropping columns that have at least one missing value
```

Out[103]:

0
1
2
3

In [104]:

```
pd.isna(df1['three'])
```

Out[104]:

```
0    False
1     True
2     True
3     True
Name: three, dtype: bool
```

In [105]:

```
df1['three'].notna()
```

Out[105]:

```
0     True
1    False
2    False
3    False
Name: three, dtype: bool
```

In [106]:

```
df1.fillna("Missing")
```

Out[106]:

	one	two	three
0	1	2	3
1	3	4	Missing
2	5	6	Missing
3	Missing	Missing	Missing

In [107]:

```
df1['three'].values
```

Out[107]:

```
array([ 3., nan, nan, nan])
```

In [112]:

```
flights = pd.read_csv("http://rcs.bu.edu/examples/python/data_analysis/flights.csv")
flights.head() #I am trying out the functions on same dataset to check if i get same result
```

Out[112]:

	year	month	day	dep_time	dep_delay	arr_time	arr_delay	carrier	tailnum	flight	origin
0	2013	1	1	517.0	2.0	830.0	11.0	UA	N14228	1545	EWR
1	2013	1	1	533.0	4.0	850.0	20.0	UA	N24211	1714	LGA
2	2013	1	1	542.0	2.0	923.0	33.0	AA	N619AA	1141	JFK
3	2013	1	1	554.0	-6.0	812.0	-25.0	DL	N668DN	461	LGA
4	2013	1	1	554.0	-4.0	740.0	12.0	UA	N39463	1696	EWR

In [110]:

```
flights.shape
```

Out[110]:

```
(160754, 16)
```

In [111]:

```
flights[flights.isnull().any(axis=1)].head()
```

Out[111]:

	year	month	day	dep_time	dep_delay	arr_time	arr_delay	carrier	tailnum	flight	orig
330	2013	1	1	1807.0	29.0	2251.0	NaN	UA	N31412	1228	EW
403	2013	1	1	NaN	NaN	NaN	NaN	AA	N3EHAA	791	LG
404	2013	1	1	NaN	NaN	NaN	NaN	AA	N3EVAA	1925	LG
855	2013	1	2	2145.0	16.0	NaN	NaN	UA	N12221	1299	EW
858	2013	1	2	NaN	NaN	NaN	NaN	AA	NaN	133	JF

In [118]:

```
flights[flights.isnull().any(axis=1)].shape
```

Out[118]:

```
(2827, 16)
```

In [114]:

```
flights1 = flights[ flights['arr_delay'].notnull( )]
flights1.head()
```

Out[114]:

	year	month	day	dep_time	dep_delay	arr_time	arr_delay	carrier	tailnum	flight	origin
0	2013	1	1	517.0	2.0	830.0	11.0	UA	N14228	1545	EWR
1	2013	1	1	533.0	4.0	850.0	20.0	UA	N24211	1714	LGA
2	2013	1	1	542.0	2.0	923.0	33.0	AA	N619AA	1141	JFK
3	2013	1	1	554.0	-6.0	812.0	-25.0	DL	N668DN	461	LGA
4	2013	1	1	554.0	-4.0	740.0	12.0	UA	N39463	1696	EWR

In [115]:

```
flights2 = flights.dropna()
flights2.shape
```

Out[115]:

```
(157927, 16)
```

In [116]:

```
nomiss =flights['dep_delay'].fillna(0)
nomiss.isnull().any()
```

Out[116]:

```
False
```

In [119]:

```
flights.count()
```

Out[119]:

```
year         160754
month        160754
day          160754
dep_time     158418
dep_delay    158418
arr_time     158275
arr_delay    157927
carrier      160754
tailnum      159321
flight       160754
origin       160754
dest         160754
air_time     157927
distance     160754
hour         158418
minute       158418
dtype: int64
```

In [127]:

```
flights.min()
```

Out[127]:

```
year         2013
month           1
day             1
dep_time        1
dep_delay     -33
arr_time        1
arr_delay     -75
carrier        AA
flight          1
origin        EWR
dest          ANC
air_time       21
distance       17
hour            0
minute          0
dtype: object
```

In [128]:

```
flights.groupby('carrier')['dep_delay'].mean()
```

Out[128]:

```
carrier
AA     8.586016
AS     5.804775
DL     9.264505
UA    12.106073
US     3.782418
Name: dep_delay, dtype: float64
```

In [129]:

```
# We can use agg() methods for aggregation:
flights[['dep_delay','arr_delay']].agg(['min','mean','max'])
```

Out[129]:

	dep_delay	arr_delay
min	-33.000000	-75.000000
mean	9.463773	2.094537
max	1014.000000	1007.000000

In [130]:

```
# An example of computing different statistics for different columns
flights.agg({'dep_delay':['max','mean','min'], 'carrier':['nunique']})
```

Out[130]:

	dep_delay	carrier
max	1014.000000	NaN
mean	9.463773	NaN
min	-33.000000	NaN
nunique	NaN	5.0

In [131]:

```
flights.dep_delay.describe()
```

Out[131]:

```
count    158418.000000
mean          9.463773
std          36.545109
min         -33.000000
25%          -5.000000
50%          -2.000000
75%           7.000000
max        1014.000000
Name: dep_delay, dtype: float64
```

In [132]:

```
flights['dep_delay'].idxmin()
```

Out[132]:

```
54111
```

In [133]:

```
flights['carrier'].value_counts()
```

Out[133]:

```
UA    58665
DL    48110
AA    32729
US    20536
AS      714
Name: carrier, dtype: int64
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: