PMDS508L - Python Programming
Jupyter Notebook Demonstrating Python Pandas Basics

Pandas Basics

[1]:
```
import pandas as pd
```

[2]:
```
d = {'one' : pd.Series([1, 2, 3], index=['a', 'b', 'c']),
   'two' : pd.Series([1, 2, 3, 4], index=['a', 'b', 'c', 'd'])}

df = pd.DataFrame(d)
print(df['one'])
```

```
a    1.0
b    2.0
c    3.0
d    NaN
Name: one, dtype: float64
```

[3]:
```
df['three'] = pd.Series([20,21,22],index=['a','b','c'])
```

[4]:
```
print(df)
```

```
   one  two  three
a  1.0    1   20.0
b  2.0    2   21.0
c  3.0    3   22.0
d  NaN    4    NaN
```

[5]:
```
df['four'] = df['two'] + df['three']
```

[6]:
```
print(df)
```

```
   one  two  three  four
a  1.0    1   20.0  21.0
b  2.0    2   21.0  23.0
c  3.0    3   22.0  25.0
d  NaN    4    NaN   NaN
```

[7]:
```
print(df[1:3])
```

```
   one  two  three  four
b  2.0    2   21.0  23.0
c  3.0    3   22.0  25.0
```

[8]:
```
df.loc['a']
```

[8]:
```
one       1.0
two       1.0
three    20.0
four     21.0
Name: a, dtype: float64
```

[9]:
```
df[0:2]
```

[9]:
```
   one  two  three  four
a  1.0    1   20.0  21.0
b  2.0    2   21.0  23.0
```

[10]:
```
df1 = pd.DataFrame([[1,2],[3,4]],index=['a','b'])
```

[11]:
```
df2 = pd.DataFrame([[5,6],[7,8]],index=['a','b'])
```

[12]:
```
df3 = pd.DataFrame([[9,10],[11,12]],columns = [2,3])
```

[13]:
```
print(df1)
print(df2)
print(df3)
```

```
   0  1
a  1  2
b  3  4
   0  1
a  5  6
b  7  8
    2   3
0   9  10
1  11  12
```

[14]:
```
df1 = df1.append(df3)
```

```
C:\Users\BSRVPrasad\AppData\Local\Temp\ipykernel_11592\1176390029.py:1:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  df1 = df1.append(df3)
```

[15]:
```
print(df1)
```

```
     0    1     2     3
a  1.0  2.0   NaN   NaN
b  3.0  4.0   NaN   NaN
```

```
0  NaN  NaN   9.0  10.0
1  NaN  NaN  11.0  12.0
```

```
[16]: df1.size
```

```
[16]: 16
```

```
[17]: df1.dtypes
```

```
[17]: 0    float64
      1    float64
      2    float64
      3    float64
      dtype: object
```

```
[18]: df2 =  pd.DataFrame([[14,15],[10,11]], columns=['one','two'])
```

```
[19]: df2.dtypes
```

```
[19]: one    int64
      two    int64
      dtype: object
```

```
[20]: df1.shape
```

```
[20]: (4, 4)
```

```
[21]: df1
```

```
[21]:      0    1     2     3
      a  1.0  2.0   NaN   NaN
      b  3.0  4.0   NaN   NaN
      0  NaN  NaN   9.0  10.0
      1  NaN  NaN  11.0  12.0
```

```
[22]: df1.cumsum()
```

```
[22]:      0    1     2     3
      a  1.0  2.0   NaN   NaN
      b  4.0  6.0   NaN   NaN
      0  NaN  NaN   9.0  10.0
      1  NaN  NaN  20.0  22.0
```

```
[23]: df = df.append(df2)
```

```
C:\Users\BSRVPrasad\AppData\Local\Temp\ipykernel_11592\948459739.py:1:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  df = df.append(df2)
```

[24]:
```
df
```

[24]:
```
    one  two  three  four
a   1.0    1   20.0  21.0
b   2.0    2   21.0  23.0
c   3.0    3   22.0  25.0
d   NaN    4    NaN   NaN
0  14.0   15    NaN   NaN
1  10.0   11    NaN   NaN
```

[25]:
```
df3 =  pd.DataFrame([[24,25],[20,21]], columns=['one','two'], index = ['a',0])
```

[26]:
```
df3
```

[26]:
```
   one  two
a   24   25
0   20   21
```

[27]:
```
df = df.append(df3)
```

```
C:\Users\BSRVPrasad\AppData\Local\Temp\ipykernel_11592\1066977462.py:1:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  df = df.append(df3)
```

[28]:
```
df
```

[28]:
```
    one  two  three  four
a   1.0    1   20.0  21.0
b   2.0    2   21.0  23.0
c   3.0    3   22.0  25.0
d   NaN    4    NaN   NaN
0  14.0   15    NaN   NaN
1  10.0   11    NaN   NaN
a  24.0   25    NaN   NaN
0  20.0   21    NaN   NaN
```

[29]:
```
df.T
```

[29]:
```
          a     b     c    d     0     1     a     0
one     1.0   2.0   3.0  NaN  14.0  10.0  24.0  20.0
two     1.0   2.0   3.0  4.0  15.0  11.0  25.0  21.0
three  20.0  21.0  22.0  NaN   NaN   NaN   NaN   NaN
four   21.0  23.0  25.0  NaN   NaN   NaN   NaN   NaN
```

[30]:
```
df.info() #Checks whether there are any null values in each column of the␣
 ↪dataframe
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 8 entries, a to 0
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   one     7 non-null      float64
 1   two     8 non-null      int64
 2   three   3 non-null      float64
 3   four    3 non-null      float64
dtypes: float64(3), int64(1)
memory usage: 620.0+ bytes
```

[31]:
```
df.axes
```

[31]:
```
[Index(['a', 'b', 'c', 'd', 0, 1, 'a', 0], dtype='object'),
 Index(['one', 'two', 'three', 'four'], dtype='object')]
```

[32]:
```
df.dtypes
```

[32]:
```
one      float64
two        int64
three    float64
four     float64
dtype: object
```

[33]:
```
df.empty
```

[33]:
```
False
```

[34]:
```
df.ndim
```

[34]:
```
2
```

[35]:
```
df.shape
```

[35]:
```
(8, 4)
```

[36]:
```
df.size
```

[36]:
```
32
```

[37]:
```
df.describe()
```

[37]:
```
             one        two  three  four
count   7.000000   8.000000    3.0   3.0
mean   10.571429  10.250000   21.0  23.0
std     9.162553   9.269767    1.0   2.0
min     1.000000   1.000000   20.0  21.0
25%     2.500000   2.750000   20.5  22.0
```

```
50%    10.000000   7.500000   21.0  23.0
75%    17.000000  16.500000   21.5  24.0
max    24.000000  25.000000   22.0  25.0
```

[38]:
```
df.sum()
```

[38]:
```
one      74.0
two      82.0
three    63.0
four     69.0
dtype: float64
```

[39]:
```
df.sum(axis=1) #df.sum(1)
```

[39]:
```
a    43.0
b    48.0
c    53.0
d     4.0
0    29.0
1    21.0
a    49.0
0    41.0
dtype: float64
```

[40]:
```
df.mean()
```

[40]:
```
one      10.571429
two      10.250000
three    21.000000
four     23.000000
dtype: float64
```

[41]:
```
df.head()
```

[41]:
```
    one  two  three  four
a   1.0    1   20.0  21.0
b   2.0    2   21.0  23.0
c   3.0    3   22.0  25.0
d   NaN    4    NaN   NaN
0  14.0   15    NaN   NaN
```

[42]:
```
df.tail()
```

[42]:
```
    one  two  three  four
d   NaN    4    NaN   NaN
0  14.0   15    NaN   NaN
1  10.0   11    NaN   NaN
a  24.0   25    NaN   NaN
```

```
0  20.0   21    NaN    NaN
```

[43]:

```
df1
```

[43]:

```
     0    1     2     3
a  1.0  2.0   NaN   NaN
b  3.0  4.0   NaN   NaN
0  NaN  NaN   9.0  10.0
1  NaN  NaN  11.0  12.0
```

[44]:

```
df1.columns
```

[44]:

```
Int64Index([0, 1, 2, 3], dtype='int64')
```

[45]:

```
df1
```

[45]:

```
     0    1     2     3
a  1.0  2.0   NaN   NaN
b  3.0  4.0   NaN   NaN
0  NaN  NaN   9.0  10.0
1  NaN  NaN  11.0  12.0
```

[46]:

```
df1.drop(1,axis=1)
```

[46]:

```
     0     2     3
a  1.0   NaN   NaN
b  3.0   NaN   NaN
0  NaN   9.0  10.0
1  NaN  11.0  12.0
```

[47]:

```
df1
```

[47]:

```
     0    1     2     3
a  1.0  2.0   NaN   NaN
b  3.0  4.0   NaN   NaN
0  NaN  NaN   9.0  10.0
1  NaN  NaN  11.0  12.0
```

[48]:

```
df11 = pd.DataFrame([[1, 2], [3, 4]], columns = ['a','b'])
df21 = pd.DataFrame([[5, 6], [7, 8]], columns = ['a','b'])
```

[49]:

```
df11 = pd.concat([df11,df21],ignore_index=True)
```

[50]:

```
df11
```

[50]:

```
   a  b
0  1  2
1  3  4
```

```
2  5  6
3  7  8
```

```
[51]: df11
```

```
[51]:    a  b
      0  1  2
      1  3  4
      2  5  6
      3  7  8
```

```
[52]: df21
```

```
[52]:    a  b
      0  5  6
      1  7  8
```

```
[53]: del df11
```

```
[54]: df11 = pd.DataFrame([[1, 2], [3, 4]], columns = ['a','b'])
```

```
[55]: df11
```

```
[55]:    a  b
      0  1  2
      1  3  4
```

```
[56]: df11 = df11.append(df21)
```

```
C:\Users\BSRVPrasad\AppData\Local\Temp\ipykernel_11592\899904042.py:1:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  df11 = df11.append(df21)
```

```
[57]: df11.drop('b',axis=1)
```

```
[57]:    a
      0  1
      1  3
      0  5
      1  7
```

```
[58]: df11
```

```
[58]:    a  b
      0  1  2
      1  3  4
      0  5  6
```

```
1  7  8
```

Reading the data from Text files and Excel Files

```
[59]: data1 = pd.read_csv('data1.txt')
```

```
[60]: data1
```

```
[60]:    1   2   3   4
      0  5   6   7   8
      1  9  10  11  12
```

```
[61]: data2 = pd.read_csv('data2.txt')
```

```
[62]: data2
```

```
[62]:      1     2    3   4
      0  5.0   NaN  7.0   8
      1  NaN  10.0  NaN  12
```

```
[63]: WQdata = pd.read_excel('WQ.xlsx')
```

```
[64]: WQdata.shape
```

```
[64]: (36, 16)
```

```
[65]: WQdata.describe()
```

```
[65]:           St. No      Depth        S.D       W.T   Salinity  Turbidity  \
      count  36.000000  35.000000  34.000000  35.00000  35.000000  35.000000
      mean   18.500000   1.554286   0.533676  30.76000  16.045714  16.842286
      std    10.535654   0.786501   0.389326   1.49119   8.986210  12.950463
      min     1.000000   0.250000   0.010000  28.50000   0.000000   1.800000
      25%     9.750000   0.800000   0.250000  29.70000  10.800000   7.070000
      50%    18.500000   1.250000   0.500000  30.30000  14.200000  13.800000
      75%    27.250000   2.250000   0.750000  31.85000  19.550000  25.820000
      max    36.000000   3.250000   1.500000  34.20000  35.300000  55.600000

                    DO         pH         TSM      NH4-N      NO2-N      NO3-N  \
      count  35.000000  35.000000   35.000000  35.000000  35.000000  35.000000
      mean    7.236571   8.390571   33.102829   3.422857   0.499429   3.169714
      std     1.341053   0.375264   32.129909   3.344098   0.446153   3.949222
      min     4.870000   7.530000    1.279000   0.200000   0.100000   0.400000
      25%     6.500000   8.240000   18.000000   0.800000   0.205000   0.985000
      50%     6.980000   8.310000   23.200000   2.300000   0.390000   1.270000
      75%     7.835000   8.500000   34.240000   4.750000   0.505000   4.250000
      max    10.600000   9.610000  170.200000  12.600000   1.930000  16.530000
```

```
           PO4-P    SiO2-Si         TN         TP
count  35.000000  35.000000  31.000000  21.000000
mean    2.674000  21.646000  13.525806   5.428095
std     2.192376  16.826631   9.120013   4.529567
min     0.290000   1.500000   1.000000   1.100000
25%     0.780000   9.900000   5.400000   2.200000
50%     2.570000  19.800000  11.600000   3.800000
75%     3.370000  27.100000  20.400000   6.800000
max     9.300000  80.000000  32.000000  17.400000
```

[66]:
```python
WQdata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36 entries, 0 to 35
Data columns (total 16 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   St. No     36 non-null     int64
 1   Depth      35 non-null     float64
 2   S.D        34 non-null     float64
 3   W.T        35 non-null     float64
 4   Salinity   35 non-null     float64
 5   Turbidity  35 non-null     float64
 6   DO         35 non-null     float64
 7   pH         35 non-null     float64
 8   TSM        35 non-null     float64
 9   NH4-N      35 non-null     float64
 10  NO2-N      35 non-null     float64
 11  NO3-N      35 non-null     float64
 12  PO4-P      35 non-null     float64
 13  SiO2-Si    35 non-null     float64
 14  TN         31 non-null     float64
 15  TP         21 non-null     float64
dtypes: float64(15), int64(1)
memory usage: 4.6 KB
```

[67]:
```python
xl = pd.ExcelFile('WQ.xlsx')
```

[68]:
```python
sheets = xl.sheet_names
print(sheets)
```

```
['May 04', 'June 04', 'July 04', 'Aug 04', 'Sep 04', 'Oct 04', 'Nov 04', 'Dec
04', 'Jan 05', 'Feb 05', 'Mar 05', 'Apr 05', 'May 05', 'June 05', 'July 05',
'Aug 05', 'Sep 05', 'Oct 05', 'Nov 05', 'Dec 05', 'Jan 06', 'Feb 06', 'Mar 06',
'Apr 06', 'May 06', 'June 06', 'July 06', 'Aug 06', 'Sep 06']
```

[69]:
```python
WQ = pd.DataFrame()
for name in sheets:
    temp = pd.read_excel('WQ.xlsx',sheet_name=name)
    Month = pd.DataFrame([name]*temp.shape[0],columns=['Month'])
    temp = pd.concat([Month,temp],axis=1)
    #print(temp)
    if WQ.empty == True:
        WQ = temp
    else:
        WQ = pd.concat([WQ, temp])
```

```
[70]: WQ.head()
```

```
[70]:     Month  St. No  Depth  S.D   W.T  Salinity  Turbidity    DO    pH   TSM  \
      0  May 04       1   1.00  1.0  29.5      10.0       5.21  8.37  9.05  18.0
      1  May 04       2   2.25  0.5  30.0      10.4      16.00  6.90  8.27  23.2
      2  May 04       3   2.50  0.5  29.6      11.8      13.70  6.79  8.27  26.8
      3  May 04       4   1.85  NaN  29.6      10.7       3.72  5.17  8.64  22.6
      4  May 04       5   1.50  1.5  29.3      11.7       4.22  6.46  8.70  18.0

         NH4-N  NO2-N  NO3-N  PO4-P  SiO2-Si    TN    TP
      0    1.5   0.45   4.20    6.3      1.5   NaN   NaN
      1    9.9   0.39   1.04    4.5      8.0  19.1  11.6
      2   11.2   0.30   0.79    6.0     12.0  18.6  11.6
      3    2.9   0.21   1.07    9.3      9.0  21.1   NaN
      4    2.3   0.41   1.09    6.6      8.0  29.1  10.8
```

```
[71]: WQ.tail()
```

```
[71]:      Month  St. No  Depth  S.D   W.T  Salinity  Turbidity    DO    pH    TSM  \
      31  Sep 06      32   1.50  0.0  29.5      0.16       45.4  6.97  7.27   53.0
      32  Sep 06      33   2.50  0.0  29.0      0.13      150.0  6.82  7.61  212.0
      33  Sep 06      34   1.75  0.0  31.5      0.20      332.0  6.20  8.01  243.0
      34  Sep 06      35   1.50  0.0  31.0      1.54      170.0  6.35  8.07   98.0
      35  Sep 06      36   3.50  0.5  31.5      1.66      252.0  6.28  8.12  323.0

            NH4-N    NO2-N     NO3-N    PO4-P   SiO2-Si        TN       TP
      31  0.41024  0.12741   0.29809  0.55437  103.2355   9.58318  1.33308
      32  0.46152  0.42333  14.55427  0.73916  178.9055  92.91518  1.67164
      33  0.46152  0.25800   7.82650  0.46741  170.2575  49.99920  1.05800
      34  0.92304  0.19350   8.18885  0.57611  142.1515  44.99928  1.20612
      35  1.74352  0.27950   7.16675  0.23914  143.2325  40.83268  0.59248
```

```
[72]: WQ.describe()
```

```
[72]:             St. No        Depth          S.D          W.T     Salinity  \
      count  1044.000000  1034.000000  1033.000000  1035.000000  1035.000000
      mean     18.500000     1.742621     0.576462    29.256531    12.163740
      std      10.393274     0.792122     0.532560     2.734294     9.174874
```

```
min       1.000000     0.100000     0.000000    20.100000     0.000000
25%       9.750000     1.250000     0.150000    27.850000     4.405000
50%      18.500000     1.750000     0.500000    29.500000    11.130000
75%      27.250000     2.250000     1.000000    31.300000    19.100000
max      36.000000     4.500000     3.750000    37.800000    35.300000

         Turbidity           DO           pH          TSM        NH4-N  \
count  1030.000000  1035.000000  1036.000000  1026.000000  1026.000000
mean     54.999029     7.159511     8.304210    75.191285     2.960870
std     113.721950     1.700331     0.417211   148.795049     2.705394
min       0.040000     1.110000     6.760000     0.800000     0.000000
25%       5.017500     6.350000     8.060000    19.200000     1.025600
50%      14.000000     7.000000     8.255000    32.550000     2.202520
75%      49.925000     7.970000     8.500000    69.382500     4.035200
max     996.000000    15.950000    10.050000  2656.000000    22.117140

             NO2-N        NO3-N        PO4-P      SiO2-Si           TN  \
count  1036.000000  1031.000000  1035.000000  1033.000000  1027.000000
mean      0.273333     2.292057     0.327828    41.252782    29.351678
std       0.392468     3.595086     0.671334    30.656762    17.398524
min       0.000000     0.000000     0.000000     0.370320     1.000000
25%       0.072720     0.399960     0.079200    17.364600    17.941580
50%       0.152705     0.854230     0.145620    35.262100    25.113000
75%       0.315750     2.216085     0.302990    55.777600    37.609045
max       5.748160    22.791220     9.300000   194.039500   138.779640

                TP
count  1012.000000
mean      0.857775
std       1.342963
min       0.070560
25%       0.307655
50%       0.481710
75%       0.844900
max      17.400000
```

[73]:

```
WQ.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1044 entries, 0 to 35
Data columns (total 17 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Month      1044 non-null   object
 1   St. No     1044 non-null   int64
 2   Depth      1034 non-null   float64
 3   S.D        1033 non-null   float64
 4   W.T        1035 non-null   float64
 5   Salinity   1035 non-null   float64
 6   Turbidity  1030 non-null   float64
 7   DO         1035 non-null   float64
 8   pH         1036 non-null   float64
 9   TSM        1026 non-null   float64
 10  NH4-N      1026 non-null   float64
 11  NO2-N      1036 non-null   float64
 12  NO3-N      1031 non-null   float64
 13  PO4-P      1035 non-null   float64
 14  SiO2-Si    1033 non-null   float64
 15  TN         1027 non-null   float64
 16  TP         1012 non-null   float64
dtypes: float64(15), int64(1), object(1)
memory usage: 146.8+ KB
```

```
[74]: WQG = WQ.groupby('Month').describe()
```

```
[75]: WQG
```

```
[75]:         St. No                                                  Depth          \
           count  mean        std  min   25%   50%    75%   max count       mean
 Month
 Apr 05     36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  36.0  1.380556
 Apr 06     36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  35.0  1.478571
 Aug 04     36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  36.0  1.847222
 Aug 05     36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  35.0  2.231429
 Aug 06     36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  36.0  2.354167
 Dec 04     36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  36.0  1.568889
 Dec 05     36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  36.0  1.823611
 Feb 05     36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  36.0  1.388889
 Feb 06     36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  36.0  1.479167
 Jan 05     36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  36.0  1.548611
 Jan 06     36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  36.0  1.537500
 July 04    36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  34.0  1.583824
 July 05    36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  36.0  1.916667
 July 06    36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  36.0  1.936111
 June 04    36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  34.0  1.764706
 June 05    36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  36.0  1.795833
 June 06    36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  36.0  1.787500
 Mar 05     36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  36.0  1.323611
 Mar 06     36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  36.0  1.319444
 May 04     36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  35.0  1.554286
 May 05     36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  34.0  1.639706
 May 06     36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  35.0  1.610000
 Nov 04     36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  36.0  1.615278
 Nov 05     36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  36.0  1.918056
 Oct 04     36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  36.0  1.854167
 Oct 05    36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  36.0  2.119444
 Sep 04    36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  36.0  1.569167
 Sep 05    36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  36.0  2.340278
 Sep 06    36.0  18.5  10.535654  1.0  9.75  18.5  27.25  36.0  36.0  2.233333

          ...         TN               TP                                  \
          ...        75%         max count      mean       std      min
 Month    ...
 Apr 05   ...  34.230300   55.90612  36.0  0.956509  0.432982  0.35552
 Apr 06   ...  23.888865   56.66661  35.0  0.474532  0.174322  0.20230
 Aug 04   ...  46.857600  111.08755  36.0  1.646746  1.148325  0.21400
 Aug 05   ...  21.756850   40.38320  35.0  2.103575  1.000277  0.63630
 Aug 06   ...  61.785220  125.71328  36.0  0.433544  0.250895  0.14966
 Dec 04   ...  28.746000   49.26000  36.0  0.530835  0.470406  0.15568
 Dec 05   ...  22.586365   50.00035  36.0  0.367771  0.245646  0.08936
 Feb 05   ...  32.601600   59.10336  36.0  0.298594  0.204916  0.09930
 Feb 06   ...  28.707097   51.03484  36.0  0.502114  0.297327  0.14280
 Jan 05   ...  46.647000  138.77964  36.0  0.537713  0.486317  0.14520
 Jan 06   ...  18.518500   32.96293  35.0  0.397333  0.225449  0.13404
 July 04  ...  47.110750   71.62200  26.0  3.250831  3.390553  0.34800
 July 05  ...  47.334187   71.83820  36.0  0.667881  0.522992  0.08775
 July 06  ...  25.989910   41.37960  36.0  0.406346  0.205541  0.21525
 June 04  ...  47.004000   82.08000  34.0  1.301741  0.771296  0.15980
 June 05  ...  43.176775   54.05625  36.0  0.258319  0.165775  0.07800
 June 06  ...  22.666440   33.99966  36.0  0.479708  0.126147  0.28725
 Mar 05   ...  29.453200   43.71080  36.0  0.530152  0.360777  0.11916
 Mar 06   ...  40.086487   56.92228  36.0  0.571013  0.307171  0.17640
 May 04   ...  20.400000   32.00000  21.0  5.428095  4.529567  1.10000
 May 05   ...  50.207850   83.15336  35.0  1.259644  1.706591  0.09729
 May 06   ...  23.019770   29.25923  35.0  0.433488  0.103450  0.27209
 Nov 04   ...  42.089987   64.41090  36.0  0.701683  0.422216  0.23805
 Nov 05   ...  30.191885   50.76852  36.0  0.388728  0.204024  0.07056
 Oct 04   ...  41.265925   56.76750  36.0  1.345954  1.112400  0.19320
 Oct 05   ...  30.192185   56.53767  36.0  0.290043  0.218859  0.07994
 Sep 04   ...  36.851450   52.36400  36.0  0.758186  0.420298  0.12100
 Sep 05   ...  48.332850  103.99896  36.0  0.434504  0.182820  0.16184
 Sep 06   ...  59.061555  103.74834  36.0  0.724143  0.462847  0.13188


               25%       50%       75%        max
 Month
 Apr 05   0.689150  0.888800  1.156502   2.255400
 Apr 06   0.321300  0.464100  0.561820   0.916300
 Aug 04   0.611995  1.493840  2.577195   5.198200
 Aug 05   1.120930  2.317950  2.999430   4.189680
 Aug 06   0.221817  0.374150  0.548935   1.090380
 Dec 04   0.312878  0.405080  0.490343   2.383850
 Dec 05   0.173135  0.329515  0.463555   1.251040
 Feb 05   0.128700  0.203565  0.409725   0.795200
 Feb 06   0.327250  0.416500  0.538475   1.392300
 Jan 05   0.308300  0.437670  0.595525   2.919600
 Jan 06   0.212230  0.335100  0.536160   0.927110
 July 04  0.943900  2.434200  3.908800  13.440000
 July 05  0.246938  0.620945  0.965812   2.590650
 July 06  0.297250  0.345700  0.447720   1.383030
 June 04  0.810300  1.076100  1.633250   3.408000
 June 05  0.164988  0.206300  0.298100   0.737200
 June 06  0.367680  0.442365  0.577373   0.781320
 Mar 05   0.297900  0.417060  0.595800   1.787400
 Mar 06   0.335160  0.593880  0.682080   1.599360
 May 04   2.200000  3.800000  6.800000  17.400000
 May 05   0.193200  0.441600  1.210320   5.567680
 May 06   0.354900  0.425880  0.514640   0.714000
 Nov 04   0.498125  0.598440  0.736088   2.650450
 Nov 05   0.232785  0.382570  0.556725   0.730880
 Oct 04   0.596075  1.074440  1.768800   5.569200
 Oct 05   0.196690  0.251240  0.324065   1.404660
 Sep 04   0.478595  0.644220  0.887250   1.870000
 Sep 05   0.289560  0.418140  0.559035   0.969696
 Sep 06   0.280245  0.615440  1.073870   1.671640

 [29 rows x 128 columns]
```