Good Morning Everyone

Pandas (http://pandas.pydata.org/)

- Data Importing/Exporting
- Data Analysis
- Data Cleaning
- Data Visualization

Pandas is an open-source python library providing efficient easy-to-use data structures and analysis tools

- Series
- DataFrame

In [1]:

```
import pandas as pd
```

--> pip install pandas

In [4]:

```
s1 = pd.Series([1, 2, 3])
print(s1)
```

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

In [6]:

```
print(type(s1))
```

```
<class 'pandas.core.series.Series'>
```

In [7]:

```
import numpy as np

print(np.array([1,2,3]))
```

```
[1 2 3]
```

In [8]:

```
s1 = pd.Series([1, 2, 3], index=['one', 'two', 'three'])
print(s1)
```

```
one      1
two      2
three    3
dtype: int64
```

In [9]:

```
s1[0]
```

Out[9]:

```
1
```

In [10]:

```
s1['one']
```

Out[10]:

```
1
```

In [11]:

```
s1['two']
```

Out[11]:

```
2
```

DataFrame

In [ ]:

```
[1,2]


[[1,2], [3,4]]
```

In [92]:

```
df = pd.DataFrame([[1,2], [3,4]])
print(df)
```

```
   0
3  1
4  2
```

In [15]:

```
df = pd.DataFrame([[1,2], [3,4]], index=['r1', 'r2'], columns=['c1', 'c2'])
print(df)
```

```
    c1  c2
r1   1   2
r2   3   4
```

In [16]:

```
print(type(df))
```

```
<class 'pandas.core.frame.DataFrame'>
```

In [17]:

```
df['c1']
```

Out[17]:

```
r1    1
r2    3
Name: c1, dtype: int64
```

In [18]:

```
df['c1'].dtype
```

Out[18]:

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

In [19]:

```
type(df['c1'])
```

Out[19]:

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

In [20]:

```
print(df.columns)
```

```
Index(['c1', 'c2'], dtype='object')
```

In [21]:

```
print(df.index)
```

```
Index(['r1', 'r2'], dtype='object')
```

In [22]:

```
df.columns = ['col1', 'col2']
df.index = ['row1','row2']

print(df)
```

```
      col1  col2
row1     1     2
row2     3     4
```

In [23]:

```
df['col2']
```

Out[23]:

```
row1    2
row2    4
Name: col2, dtype: int64
```

.iloc --> index location

In [25]:

```
df.iloc[0]
```

Out[25]:

```
col1    1
col2    2
Name: row1, dtype: int64
```

In [37]:

```
import numpy as np

num = {'num': np.arange(1,100), 'sqnum': np.arange(1,100)**2, 'cnum':np.arange(1,100) ** 3}
```

In [28]:

```
numdf = pd.DataFrame(num)

print(numdf)
```

```
    num  sqnum    cnum
0     1      1       1
1     2      4       8
2     3      9      27
3     4     16      64
4     5     25     125
5     6     36     216
6     7     49     343
7     8     64     512
8     9     81     729
9    10    100    1000
10   11    121    1331
11   12    144    1728
12   13    169    2197
13   14    196    2744
14   15    225    3375
15   16    256    4096
16   17    289    4913
17   18    324    5832
18   19    361    6859
19   20    400    8000
20   21    441    9261
21   22    484   10648
22   23    529   12167
23   24    576   13824
24   25    625   15625
25   26    676   17576
26   27    729   19683
27   28    784   21952
28   29    841   24389
29   30    900   27000
..  ...    ...     ...
69   70   4900  343000
70   71   5041  357911
71   72   5184  373248
72   73   5329  389017
73   74   5476  405224
74   75   5625  421875
75   76   5776  438976
76   77   5929  456533
77   78   6084  474552
78   79   6241  493039
79   80   6400  512000
80   81   6561  531441
81   82   6724  551368
82   83   6889  571787
83   84   7056  592704
84   85   7225  614125
85   86   7396  636056
86   87   7569  658503
87   88   7744  681472
88   89   7921  704969
89   90   8100  729000
90   91   8281  753571
```

```
91   92   8464  778688
92   93   8649  804357
93   94   8836  830584
94   95   9025  857375
95   96   9216  884736
96   97   9409  912673
97   98   9604  941192
98   99   9801  970299

[99 rows x 3 columns]
```

indexing

In [33]:

```
numdf.iloc[0:5]['sqnum']
```

Out[33]:

```
0     1
1     4
2     9
3    16
4    25
Name: sqnum, dtype: int32
```

.loc

In [34]:

```
print(df)
```

```
      col1  col2
row1     1     2
row2     3     4
```

In [35]:

```
df.loc['row1']
```

Out[35]:

```
col1    1
col2    2
Name: row1, dtype: int64
```

In [36]:

```
df.loc['row1']['col1']
```

Out[36]:

```
1
```

Types of Data

Structured Data -- DB, Excel, CSV.....

Semi-structured Data -- json, xml..

UnStructured Data -- doc, pdfs, images, videos

In [59]:

```
titanic = pd.read_csv('https://raw.githubusercontent.com/AP-State-Skill-Development-Corpora
titanic
```

Out[59]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	PP 9549
11	12	1	1	Bonnell, Miss. Elizabeth	female	58.0	0	0	113783
12	13	0	3	Saundercock, Mr. William Henry	male	20.0	0	0	A/5. 2151
13	14	0	3	Andersson, Mr. Anders Johan	male	39.0	1	5	347082

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
14	15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14.0	0	0	350406
15	16	1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55.0	0	0	248706
16	17	0	3	Rice, Master. Eugene	male	2.0	4	1	382652
17	18	1	2	Williams, Mr. Charles Eugene	male	NaN	0	0	244373
18	19	0	3	Vander Planke, Mrs. Julius (Emelia Maria Vande...	female	31.0	1	0	345763
19	20	1	3	Masselmani, Mrs. Fatima	female	NaN	0	0	2649
20	21	0	2	Fynney, Mr. Joseph J	male	35.0	0	0	239865
21	22	1	2	Beesley, Mr. Lawrence	male	34.0	0	0	248698
22	23	1	3	McGowan, Miss. Anna "Annie"	female	15.0	0	0	330923
23	24	1	1	Sloper, Mr. William Thompson	male	28.0	0	0	113788
24	25	0	3	Palsson, Miss. Torborg Danira	female	8.0	3	1	349909
25	26	1	3	Asplund, Mrs. Carl Oscar (Selma Augusta Emilia...	female	38.0	1	5	347077
26	27	0	3	Emir, Mr. Farred Chehab	male	NaN	0	0	2631
27	28	0	1	Fortune, Mr. Charles Alexander	male	19.0	3	2	19950
28	29	1	3	O'Dwyer, Miss. Ellen "Nellie"	female	NaN	0	0	330959
29	30	0	3	Todoroff, Mr. Lalio	male	NaN	0	0	349216
...	...	...	...	...	...	...	...	...	...
861	862	0	2	Giles, Mr. Frederick Edward	male	21.0	1	0	28134

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
862	863	1	1	Swift, Mrs. Frederick Joel (Margaret Welles Ba...	female	48.0	0	0	17466
863	864	0	3	Sage, Miss. Dorothy Edith "Dolly"	female	NaN	8	2	CA. 2343
864	865	0	2	Gill, Mr. John William	male	24.0	0	0	233866
865	866	1	2	Bystrom, Mrs. (Karolina)	female	42.0	0	0	236852
866	867	1	2	Duran y More, Miss. Asuncion	female	27.0	1	0	SC/PARIS 2149
867	868	0	1	Roebling, Mr. Washington Augustus II	male	31.0	0	0	PC 17590
868	869	0	3	van Melkebeke, Mr. Philemon	male	NaN	0	0	345777
869	870	1	3	Johnson, Master. Harold Theodor	male	4.0	1	1	347742
870	871	0	3	Balkic, Mr. Cerin	male	26.0	0	0	349248
871	872	1	1	Beckwith, Mrs. Richard Leonard (Sallie Monypeny)	female	47.0	1	1	11751
872	873	0	1	Carlsson, Mr. Frans Olof	male	33.0	0	0	695
873	874	0	3	Vander Cruyssen, Mr. Victor	male	47.0	0	0	345765
874	875	1	2	Abelson, Mrs. Samuel (Hannah Wizosky)	female	28.0	1	0	P/PP 3381
875	876	1	3	Najib, Miss. Adele Kiamie "Jane"	female	15.0	0	0	2667
876	877	0	3	Gustafsson, Mr. Alfred Ossian	male	20.0	0	0	7534
877	878	0	3	Petroff, Mr. Nedelio	male	19.0	0	0	349212
878	879	0	3	Laleff, Mr. Kristo	male	NaN	0	0	349217
879	880	1	1	Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)	female	56.0	0	1	11767

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
880	881	1	2	Shelley, Mrs. William (Imanita Parrish Hall)	female	25.0	0	1	230433
881	882	0	3	Markun, Mr. Johann	male	33.0	0	0	349257
882	883	0	3	Dahlberg, Miss. Gerda Ulrika	female	22.0	0	0	7552
883	884	0	2	Banfield, Mr. Frederick James	male	28.0	0	0	C.A./SOTON 34068
884	885	0	3	Sutehall, Mr. Henry Jr	male	25.0	0	0	SOTON/OQ 392076
885	886	0	3	Rice, Mrs. William (Margaret Norton)	female	39.0	0	5	382652
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376

891 rows × 12 columns

In [44]:

```
type(titanic)
```

Out[44]:

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

In [45]:

```
titanic.head()
```

. . .

In [46]:

```
titanic.head(10)
```

. . .

In [47]:

```
titanic.tail()
```

. . .

In [48]:

```
titanic.tail(10)
```

Out[48]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
881	882	0	3	Markun, Mr. Johann	male	33.0	0	0	349257	7.895
882	883	0	3	Dahlberg, Miss. Gerda Ulrika	female	22.0	0	0	7552	10.516
883	884	0	2	Banfield, Mr. Frederick James	male	28.0	0	0	C.A./SOTON 34068	10.500
884	885	0	3	Sutehall, Mr. Henry Jr	male	25.0	0	0	SOTON/OQ 392076	7.050
885	886	0	3	Rice, Mrs. William (Margaret Norton)	female	39.0	0	5	382652	29.125
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.000
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.000
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.450
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.000
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.750

In [50]:

```
titanic['Sex'].value_counts()
```

Out[50]:

```
male      577
female    314
Name: Sex, dtype: int64
```

In [51]:

```
titanic['Survived'].value_counts()
```

Out[51]:

```
0    549
1    342
Name: Survived, dtype: int64
```

In [54]:

```
titanic.isnull()
```

...

In [55]:

```
titanic.isnull().sum()
```

Out[55]:

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

Dealing With missing Values

- mean()
- median()
- previous value
- after value
- 0,string,...

In [68]:

```
titanic = titanic.fillna('missing_values')

titanic.tail()
```

Out[68]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
886	887	0	2	Montvila, Rev. Juozas	male	27	0	0	211536	1
887	888	1	1	Graham, Miss. Margaret Edith	female	19	0	0	112053	3
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	missing_values	1	2	W./C. 6607	2
889	890	1	1	Behr, Mr. Karl Howell	male	26	0	0	111369	3
890	891	0	3	Dooley, Mr. Patrick	male	32	0	0	370376	

In [69]:

```
titanic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
PassengerId    891 non-null int64
Survived       891 non-null int64
Pclass         891 non-null int64
Name           891 non-null object
Sex            891 non-null object
Age            891 non-null object
SibSp          891 non-null int64
Parch          891 non-null int64
Ticket         891 non-null object
Fare           891 non-null float64
Cabin          891 non-null object
Embarked       891 non-null object
dtypes: float64(1), int64(5), object(6)
memory usage: 83.6+ KB
```

In [70]:

```
titanic = pd.read_csv('https://raw.githubusercontent.com/AP-State-Skill-Development-Corpora
```

In [71]:

```
titanic.info()
```

. . .

In [60]:

```
print(titanic['Age'].mean())
```

```
29.69911764705882
```

In [73]:

```
titanic = titanic.fillna(titanic['Age'].mean())
titanic.tail()
```

Out[73]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
886	887	0	2	Montvila, Rev. Juozas	male	27.000000	0	0	211536	13.00
887	888	1	1	Graham, Miss. Margaret Edith	female	19.000000	0	0	112053	30.00
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	29.699118	1	2	W./C. 6607	23.45
889	890	1	1	Behr, Mr. Karl Howell	male	26.000000	0	0	111369	30.00
890	891	0	3	Dooley, Mr. Patrick	male	32.000000	0	0	370376	7.75

In [74]:

```
titanic.describe()
```

Out[74]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	13.002015	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	22.000000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	29.699118	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	35.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

Grouping of Data

In [78]:

```
titanic.groupby(['Survived', 'Sex']).count()
```

Out[78]:

		PassengerId	Pclass	Name	Age	SibSp	Parch	Ticket	Fare	Cabin	Embar
Survived	**Sex**										
0	**female**	81	81	81	81	81	81	81	81	81	
	male	468	468	468	468	468	468	468	468	468	
1	**female**	233	233	233	233	233	233	233	233	233	
	male	109	109	109	109	109	109	109	109	109	

In [79]:

```
titanic.sort_values(by = 'Age')
```

. . .

In [80]:

```
titanic.sort_values(by = 'Age', ascending=False)
```

. . .

In [81]:

```
titanic.index
```

Out[81]:

```
RangeIndex(start=0, stop=891, step=1)
```

In [82]:

```
titanic.set_index('PassengerId', inplace=True)

titanic.head()
```

...

In [83]:

```
titanic.sort_index(ascending=False)
```

...

In [86]:

```
titanic[['Pclass','Age']].plot(kind = 'box')
```

...

In [87]:

```
help(titanic[['Pclass','Age']].plot(kind = 'box'))
```

...

In [88]:

```
titanic.to_csv('modified_titanic.csv')
```

In [89]:

```
titanic.to_json('modified_titanic.json')
```

In [90]:

```
len(dir(pd))
```

Out[90]:

```
139
```

In [91]:

```
len(dir(df))
```

Out[91]:

```
452
```