PANDAS

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
movies = pd.read_csv("imdb.csv")
print (movies.head(5))



#discriptive statistics
print(movies.describe())



#check the column name
print("column names:%s" %movies.columns.values)



print(movies.dtypes)
print(movies.shape)



#check null data
print("null value:\n")
print(movies.isna())
print(movies.isna().sum())
print("null values %s " %movies.isnull().values.any())
print("null values %s " %movies.isna().sum().sum())
print(movies.describe())



#discretization or binning
print(movies['rating'][5:10])
print('minimum valus of all column:')
print(movies.min())



print(movies['rating'].idxmax())
print(movies.loc[movies['rating'].idxmax(),'name'])
print(movies['name'].loc[2764:2767])
print(movies[['name','year']].loc[2986:2990])
```

```
fig=plt.figure(figsize=(10,7))
sns.distplot(movies['rating'])
plt.xlabel('rating',fontsize=12)
sns.jointplot(x='rank',y='rating',data=movies);
plt.show()
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
movies = pd.read_csv("imdb.csv")
op_labels=['bad','moderate','good']
category=[0.,4.,7.,10.]
movies['year']=pd.cut(movies['rating'],labels=op_labels,bins=category,include_lowest=False)
print(movies[['name','rating','year']][35:86])
```

```
import seaborn as sns
sns.boxplot(x=movies['rating'],color='lime')
plt.xlabel('rating of the movies',fontsize=14)
plt.show()
```

```
print(movies['rating'].min())
print(movies['rating'].idxmax())
print(movies[['name','rating']].iloc[movies['rating'].idxmax()])
print(movies['rating'].describe())
```

```
print(movies.dtypes)
movies_num=movies.select_dtypes(exclude=['object'])
print(movies_num.shape)
```

```
from scipy.stats import zscore
numeric_cols=movies_num.select_dtypes(include=[np.number]).columns
movies_num[numeric_cols].apply(zscore)
```

```
from scipy.stats import zscore
movies_zscore=movies_num[(np.abs(movies_num[numeric_cols].apply(zscore))<=3).all(axis=1)]
print("shape after rejecting outliers: ")
print(movies_zscore.shape)
movies_zscore_usr_rev=movies_num[(np.abs(stats.zscore(
    movies_num[['rating']]))<=3).all(axis=1)]
print(type(movies_zscore_usr_rev))
```

MATRIX

```
import numpy as np
a=np.array([1,2,3])
print('type:%s'%type(a))
print('shape:%s'%a.shape)
print(a[0],a[1],a[2])
a[0]=5
print(a)
b=np.array([[1,2,3],(4,5,6)])
print('\n shape of b:',b.shape)
print(b[0,0],b[0,1],b[1,0])
a=np.zeros((2,2))
print('all zeros matrix:\n %s'%a)
b=np.ones((1,2))
print('\nall ones matrix:\n %s'%b)
c=np.random.random((2,2))
print('\n random matrix:\n%s'%c)
d=np.eye(2)
print('\n identify matrix:\n %s'%d)
print('vactorized sum example\n')
x=np.array([[1,2],[3,4]])
print('x:\n %s'%x)
print('sum:%s'%np.sum(x))
print('sum axis=0:%s'%np.sum(x,axis=0))
print('sum axis=1:%s'%np.sum(x,axis=1))


a=np.arange(10000)
b=np.arange(10000)
dp=np.dot(a,b)
print('dot product:%s\n' %dp)
op=np.outer(a,b)
print('\n outer product:%s\n' %op)
cp=np.multiply(a,b)
print('\n element wise product:%s\n' %cp)


import numpy as np
x=np.array([[1,2],[3,4]])
print('original x:\n %s' %x)
print('\n Transpose of x:\n %s' %x.T)
```

```
#singular value decomposition
from numpy import array
from scipy.linalg import svd
#define a matrix
A=array([[1,2],[3,4],[5,6]])
print(A)
#SVD
U,s,VT=svd(A)
print(U)
print(s)
print(VT)
```