python-numpy-pandas-matplotlib-seaborn

Use the "Run" button to execute the code.

```
!pip install jovian --upgrade --quiet
```

```
import jovian
```

```
# Execute this to save new versions of the notebook
jovian.commit(project="python-numpy-pandas-matplotlib-seaborn")
```

```
#built in function
"""
type()
str()
format()
int()
len()
print()

"""
```

```
#mrthod:
"""
var.lower()
var.upper()
var.capitalize()
var.replace('','')
var.split()
var.strip()

"""
```

```
"\nvar.lower()\nvar.upper()\nvar.capitalize()\nvar.replace('','')\nvar.split()\nvar.stri
```

```
#list:
"""
list=[1,2,3,4]
list_=list(range(1,9))

"""


#list Method:
"""
list.lower()
list.append('value')
list.insert(index,'value')
list.copy()
```

```
list.pop(index)
list.remove('value')

"""
```

```
"\nlist.lower()\nlist.append('value')\nlist.insert(index,'value')\nlist.copy()\nlist.pop
```

```
#dictionary:
"""
dict={
   'name':'Asad',
   'age':22
}

"""

#dictionaey Method:
"""
dict.keys()
dict.values()
dict.items()
dict['old_key']='value'
dict['new_key']='value'
"""

#Tuple:
"""
tup=(1,2,3,4,5)
tup=tuple(range(1,10))

"""
```

```
'\ntup=(1,2,3,4,5)\ntup=tuple(range(1,10))\n\n'
```

```
#if in var:

"""
var=3
if(var==3):
    print('')
else:
    print('')

"""

#if in list:
"""
list_=['asad','rahat','shanto']
friend='mehedi'

if friend in list:
     print('')
```

```
else:
    print('')

"""
```

```
"\nlist_=['asad','rahat','shanto']\nfriend='mehedi'\n\nif friend in list:\n
print('')\nelse:\n    print('')\n        \n"
```

```
#for in var
"""
name='asadul islam hamzah'
for x in var:
    print(x)

"""


#for in range:
"""
for i in range(10):  #0-9
    print(i)

#for in list:

for x in list_:
    print(x)

for i in range(len(list_)):
    print(i)

for i in zip(list1,list2):
    print(i)

for i,y in zip(list1,list2):
    print(i,y)

"""


#for in dictionary:
"""
for key in dict:
    print('key=',key,'value=',dict[key])

"""
```

```
"\nfor key in dict:\n    print('key=',key,'value=',dict[key])\n\n"
```

```
#Numpy:
"""
* numpy.array() , numpy.arange() , numpy.random.rand(,) #dim

* numpy.dot() or x@y , numpy.matmul(,)
```

```
* numpy.mean() , numpy.median() , numpy.mode() , numpy.std() , numpy.var() , numpy.max(

* np_array.shape , np_array.reshape() , np_array.dtype

"""
```

'\n* numpy.array() , numpy.arange() , numpy.random.rand(,) #dim\n\n* numpy.dot() or x@y , numpy.matmul(,)\n\n* numpy.mean() , numpy.median() , numpy.mode() , numpy.std() , numpy.var() , numpy.max() , numpy.sum()\n\n* np_array.shape , np_array.reshape() , np_array.dtype\n\n'

```
#download file:
"""
import urllib.request as url

url.retrieve(url,'name.ext')

"""

#take in var:
"""
in_var=numpy.genfromtxt('name.ext',delimiter='',skip_header=1)

"""

#Concatenate column:
"""
in_var_new=numpy.concatenate( (in_var,new_col.reshape(x,1)), axis=1 )

"""

#save from var:
"""
numpy.savetxt(
        'name.ext',
        var_data,
        fmt='%.2f',
        delimiter='',
        header='x,y,z'
)

"""
```

"\nnumpy.savetxt(\n 'name.ext',\n var_data,\n fmt='%.2f',\n delimiter='',\n header='x,y,z'\n)\n\n"

```
#pandas:
"""
import pandas
```

```
* data=pandas.read_csv('name.csv')

  pandas.read_csv('name.csv',index_col='Column').loc['column']  #for 1 col read

  schema=pandas.read_csv('schema.csv',index_col='Column').QuestionText_col   #for schem

* data.to_csv('name.csv',index=None)


* pandas.DatetimeIndex(time_col).year/month


* data,info() , data.describe() , data.columns , data.isnull().any()


* data.loc[:] , data.head() , data.tail() , data.tail() , dara.sample()

* data['col'] or data.col , data[['col1','col2']] , data['col'][10]


* data[data.col >10] , data [(data.col1 == True) & (data.col2 == True )]

  data.drop(column=['col'],inplace=True) , data.drop(data[data.col>10].index,inplace=Tr

  data['new_col']=new_col_data


* data.sort_values('col') , data.sort_index()


* data.col.mean() / sum() / comsum()  ...


* data['col'].value_counts() , data.col.counts()


* data.grupby('col_month')[['col']].sum()

* data .merge(data2,on='base_col')

* data.col.plot(title='',kind='')


"""
```

"\nimport pandas\n\n* data=pandas.read_csv('name.csv') \n\n pandas.read_csv('name.csv',index_col='Column').loc['column'] #for 1 col read\n \n schema=pandas.read_csv('schema.csv',index_col='Column').QuestionText_col   #for schema\n\n* data.to_csv('name.csv',index=None)\n\n\n* pandas.DatetimeIndex(time_col).year/month\n\n\n* data,info() , data.describe() , data.columns , data.isnull().any() \n\n\n* data.loc[:] , data.head() , data.tail() , data.tail() , dara.sample()\n\n* data['col'] or data.col , data[['col1','col2']] ,

```
data['col'][10]\n\n\n* data[data.col >10] , data [(data.col1 == True) & (data.col2 == True )]\n\n  data.drop(column=['col'],inplace=True) , data.drop(data[data.col>10].index,inplace=True)\n  \n  data['new_col']=new_col_data\n\n\n* data.sort_values('col') , data.sort_index()\n\n\n* data.col.mean() / sum() / comsum()  ...\n\n\n* data['col'].value_counts() , data.col.counts()\n\n\n* data.grupby('col_month')[['col']].sum()\n\n* data .merge(data2,on='base_col')\n\n* data.col.plot(title='',kind='')\n\n\n"
```

```
# matplotlib & Seaborn:
"""
import matplotlib.pyplot as plt
import seaborn as sns

"""
#plot:
"""
plt.figure(figsize=(12,6))
plt.title('')
plt.plot(x,y1,marker='',c='red')
plt.plot(x,y2,marker='',ls='--')
plt.xlabel('')
plt.ylabel('')
plt.legend(['Y1','Y2'])

"""

#Overlap Bar:  for non-num col

"""

plt.bar(x,y1)
plt.bar(x,y2,bottom=y1)

"""

#Histogram:  for num col

"""

plt.title('')
plt.hist( ['col1',col2] , bins=np.arange(,,,) , stacked=True )
plt.legend(['col1','col2']);

"""


#Pie-chart:  for non-num col
"""
plt.title()
plt.pie( col , labels=col.index , autopct='%1.1f%%' , startangle=180 );
```

```
"""


#plot in percentage:    for non-num col

"""

( df.col.value_counts(normalize=True , ascending=True)*100 ).plot( kind='barh',color='g

"""


#sns:
"""
sns.load_dataset('name')
sns_data.col.unique()

"""


#Scatterplot: for non-num col

"""
sns.scatterplot(
                x=x_col,
                y=y_col,
                hue=cat_col,  #optional
                s=100,
                data=data_df
               );

"""

#barplot:   for nom_num col

"""

sns.barplot(
            x=x_col,
            y=y_col,
            hue=cat_col,
            data=df
            );
"""


#heatmap:

"""

sns.heatmap( df , fmt='d' , annot=True , camp='Blues');
```

```
"""

#Countplot: for non-num col . it count itsef (value_counts()) of same type value

"""
sns.countplot(y=df.col)  # vertical bar

"""
```

```
'\nsns.countplot(y=df.col)  # vertical bar\n\n'
```

```

```