In [1]:

```
import numpy as np
```

In [3]:

```
a = np.array([2.1, 4.9, 3.7, 4.1, 5.2])
```

In [6]:

```
exp_a = np.exp(a)
exp_a
```

Out[6]:

```
array([  8.16616991, 134.28977968,  40.44730436,  60.3402876 ,
       181.27224188])
```

In [7]:

```
exp_a.sum()
```

Out[7]:

```
424.51578343008373
```

In [9]:

```
probab_dist = exp_a/exp_a.sum()
probab_dist
```

Out[9]:

```
array([0.01923643, 0.31633636, 0.09527868, 0.14213909, 0.42700943])
```

In [ ]:

```

```

Row wise Unique

In [12]:

```
day1 = np.array([1, 1, 2, 5, 7, 6, 7, 7, 6, 3])
day1
```

Out[12]:

```
array([1, 1, 2, 5, 7, 6, 7, 7, 6, 3])
```

In [16]:

```
uniq, count = np.unique(day1, return_counts=True)
print(uniq)
print(count)
```

```
[1 2 3 5 6 7]
[2 1 1 1 2 3]
```

In [20]:

```
result = np.zeros(10)
result
```

Out[20]:

```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

In [21]:

```
result[uniq-1] = count
result
```

Out[21]:

```
array([2., 1., 1., 0., 1., 2., 3., 0., 0., 0.])
```

In [ ]:

```

```

Pandas

In [23]:

```
import pandas as pd
```

In [26]:

```
a = [1,2,3,4,5]
b = [10,20,30,40,50]
```

In [63]:

```
df = pd.DataFrame(zip(a,b), columns=['a', 'b'])
df
```

Out[63]:

	a	b
0	1	10
1	2	20
2	3	30
3	4	40
4	5	50

In [30]:

```
string = "How much wood would a woodchuck chuck if a woodchuck could chuck wood"
```

In [50]:

```
pd.Series(string.split()).value_counts().sort_index()
```

Out[50]:

```
How           1
a             2
chuck         2
could         1
if            1
much          1
wood          2
woodchuck     2
would         1
dtype: int64
```

Stats

In [52]:

```
elements = np.array([10, 10, 6, 9, 4, 5, 7, 1, 8, 929, 100])
elements
```

Out[52]:

```
array([ 10,  10,   6,   9,   4,   5,   7,   1,   8, 929, 100])
```

In [58]:

```
iqr = np.quantile(elements, 0.75) - np.quantile(elements, 0.25)
iqr
```

Out[58]:

```
4.5
```

In [60]:

```
np.quantile(elements, 0.25) - iqr
```

Out[60]:

```
1.0
```

In [61]:

```
np.quantile(elements, 0.75) + iqr
```

Out[61]:

```
14.5
```

LOF

In [ ]:

```
df.where(cond= df['test_score']<50 )
```

In [ ]:

```

```

In [ ]:

```

```

Case Study

In [64]:

```
df = pd.read_csv("walmart.csv")
df.shape
```

Out[64]:

```
(550068, 10)
```

In [65]:

```
df.head()
```

Out[65]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	M
0	1000001	P00069042	F	0-17	10	A	2	
1	1000001	P00248942	F	0-17	10	A	2	
2	1000001	P00087842	F	0-17	10	A	2	
3	1000001	P00085442	F	0-17	10	A	2	
4	1000002	P00285442	M	55+	16	C	4+	

In [73]:

```
df.Gender.value_counts()
```

Out[73]:

```
M    414259
F    135809
Name: Gender, dtype: int64
```

In [71]:

```
df.groupby(by='Gender').mean()['Purchase']
```

Out[71]:

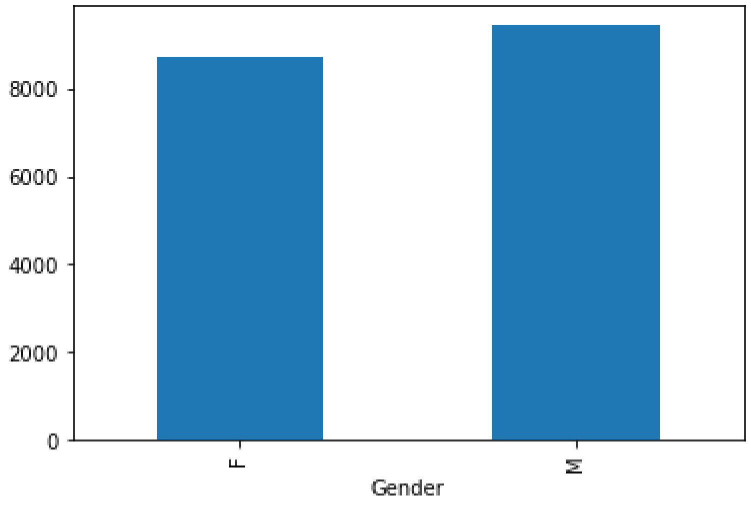
```
Gender
F    8734.565765
M    9437.526040
Name: Purchase, dtype: float64
```

In [70]:

```
df.groupby(by='Gender').mean()['Purchase'].plot.bar()
```

Out[70]:

```
<AxesSubplot:xlabel='Gender'>
```



In [74]:

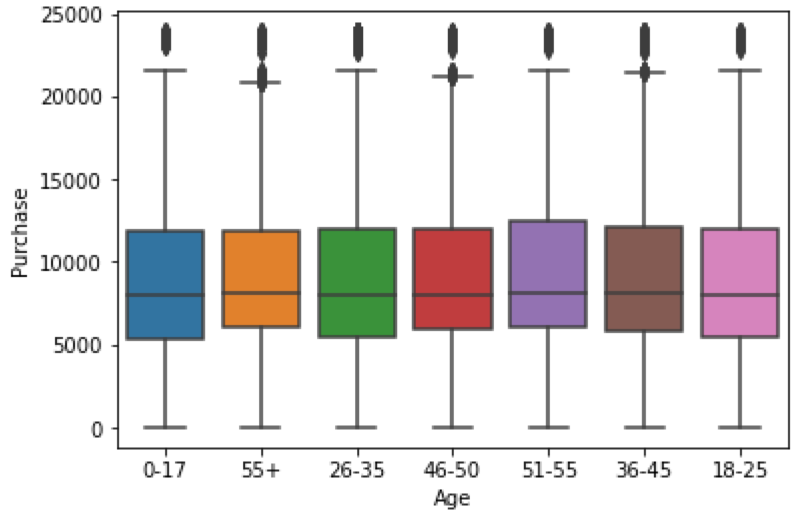
```
import seaborn as sns
```

In [75]:

```
sns.boxplot(x = 'Age', y = 'Purchase', data=df)
```

Out[75]:

```
<AxesSubplot:xlabel='Age', ylabel='Purchase'>
```



In [ ]:

```

```

In [ ]:

```

```