

```
In [1]: 1 import numpy as n
2 a=n.array([54,78,32,46,89,76])
3 print("\nAn Array:",a)
4 print("\nSum:",n.sum(a))
5 print("Product",n.prod(a))
6 print("Mean:",n.mean(a))
7 print("Standard Deviation:",n.std(a))
8 print("Variance",n.var(a))
9 print("Minimum Value:",n.min(a))
10 print("Max:",n.max(a))
11 print("Min Index:",n.argmin(a))
12 print("Max Index:",n.argmax(a))
13 print("Median:",n.median(a))
```

An Array: [54 78 32 46 89 76]

Sum: 375

Product -1012440064

Mean: 62.5

Standard Deviation: 19.997916558148418

Variance 399.9166666666667

Minimum Value: 32

Max: 89

Min Index: 2

Max Index: 4

Median: 65.0

```
In [6]: 1 import numpy as np
2 import timeit
3 np.a=[4,5,1]
4 print(np.prod(np.a))
5 print("Time taken by vectorized product : ",end= "")
6 %timeit np.prod(np.a)
7 total = 1
8 for item in np.a:
9     total =total*item
10    t = total
11 print(t)
12 print("Time taken by iterative multiplication : ",end= " ")
13
```

20

Time taken by vectorized product : 8 μ s \pm 1.36 μ s per loop (mean \pm std. de v. of 7 runs, 100000 loops each)

20

Time taken by iterative multiplication :

```
In [7]: 1 import numpy as n
2 import timeit
3 print(n.sum(n.arange(4)))
4 print("Time taken to vectorized sum:")
5 %timeit n.sum(n.arange(4))
6 t=0
7 for i in range(0,4):
8     t+=i
9 a=t
10 print("\n"+str(a))
11 print("Time Taken by iterative sum:",end="")
12 %timeit a
```

6
Time taken to vectorized sum:
6.54 μ s \pm 237 ns per loop (mean \pm std. dev. of 7 runs, 100000 loops each)

6
Time Taken by iterative sum:34.9 ns \pm 5.6 ns per loop (mean \pm std. dev. of 7 runs, 1000000 loops each)

```
In [9]: 1 import numpy as n
2 da=[60,8,7,5,34,78]
3 d=n.array(da)
4 from functools import reduce as r
5 print(list(map(lambda num:num*2,d)))
6 print(list(filter(lambda num:num>2,d)))
7 print(r(lambda x,y:x+y,d))
```

[3600, 64, 49, 25, 1156, 6084]
[60, 8, 7, 5, 34, 78]
192

```
In [10]: 1 import pandas as p
2 d=p.DataFrame([[2,5,6],
3 [4,6,3],
4 [5,7,8]],
5 columns=["Maths","Java","Py"])
6 print(d)
7 c=d.agg(['sum','min','max','count','mean','median','std','size',])
8 print()
9 print(c)
```

	Maths	Java	Py
0	2	5	6
1	4	6	3
2	5	7	8

	Maths	Java	Py
sum	11.000000	18.0	17.000000
min	2.000000	5.0	3.000000
max	5.000000	7.0	8.000000
count	3.000000	3.0	3.000000
mean	3.666667	6.0	5.666667
median	4.000000	6.0	6.000000
std	1.527525	1.0	2.516611
size	3.000000	3.0	3.000000

```
In [11]: 1 import pandas as p
2 t={
3     'Course':['PY',"JV","DBMS","MMA","MMA"],
4     'Fee':[300,600,21,350,67],
5     'Complexity':[100,56,32,10,67]
6 }
7 d=p.DataFrame(t)
8 print(d)
9 c=d.groupby('Course').agg({'Fee':'min'})
10 print("\n",c)
```

	Course	Fee	Complexity
0	PY	300	100
1	JV	600	56
2	DBMS	21	32
3	MMA	350	10
4	MMA	67	67

	Course	Fee
	DBMS	21
	JV	600
	MMA	67
	PY	300

```
In [12]: 1 import pandas as pd
2 df=pd.DataFrame({'Name':['New york','paris','london'], "temp":[10,20,23]}
3 print(df)
4 df.pivot(columns='Name',values='temp')
5 df.melt()
```

	Name	temp
0	New york	10
1	paris	20
2	london	23

```
Out[12]:
```

	variable	value
0	Name	New york
1	Name	paris
2	Name	london
3	temp	10
4	temp	20
5	temp	23

```
In [13]: 1 import pandas as pd
2 from functools import reduce
3 data = {
4     'Numbers': [1, 2, 3, 4, 5],
5     'Letters': ['A', 'B', 'C', 'D', 'E']
6 }
7 df = pd.DataFrame(data)
8 sq=df['Numbers'].map(lambda x: x**2)
9 ev=list(filter(lambda x: x % 2 == 0, df['Numbers']))
10 po = reduce(lambda x, y: x * y, df['Numbers'])
11 print("Dataframe:\n",df)
12 print("\nMap for Squaring:\n",sq)
13 print("\n Filter:\n",ev)
14 print("\nReduce for product:\n", po)
```

Dataframe:

	Numbers	Letters
0	1	A
1	2	B
2	3	C
3	4	D
4	5	E

Map for Squaring:

0	1
1	4
2	9
3	16
4	25

Name: Numbers, dtype: int64

Filter:

[2, 4]

Reduce for product:

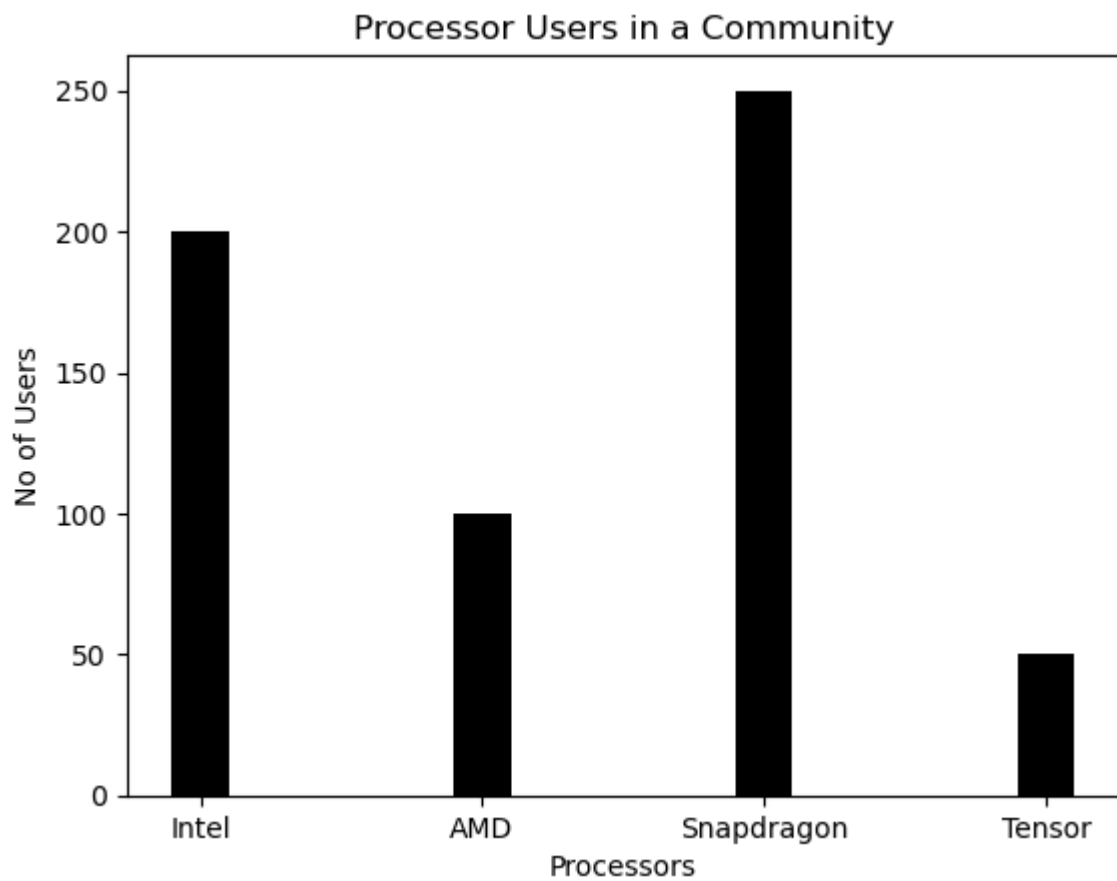
120

```
In [14]: 1 import numpy as np
2 import pandas as pd
3 df=pd.DataFrame({
4     'date':pd.date_range(start='2023-03-05',periods=20,freq='D'),"temp":np
5 df.head()
6 df['sh']=df['temp'].shift(1)
7 df.head()
8 df_week=df.resample("W",on='date').mean()
9 df_week.head()
```

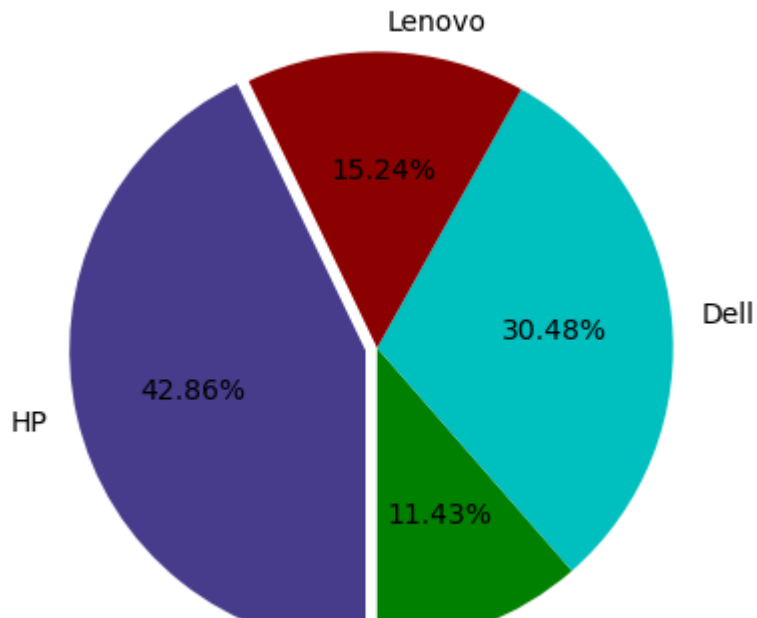
Out[14]:

	temp	sh
date		
2023-03-05	29.000000	NaN
2023-03-12	23.428571	24.428571
2023-03-19	25.857143	25.571429
2023-03-26	22.400000	23.000000

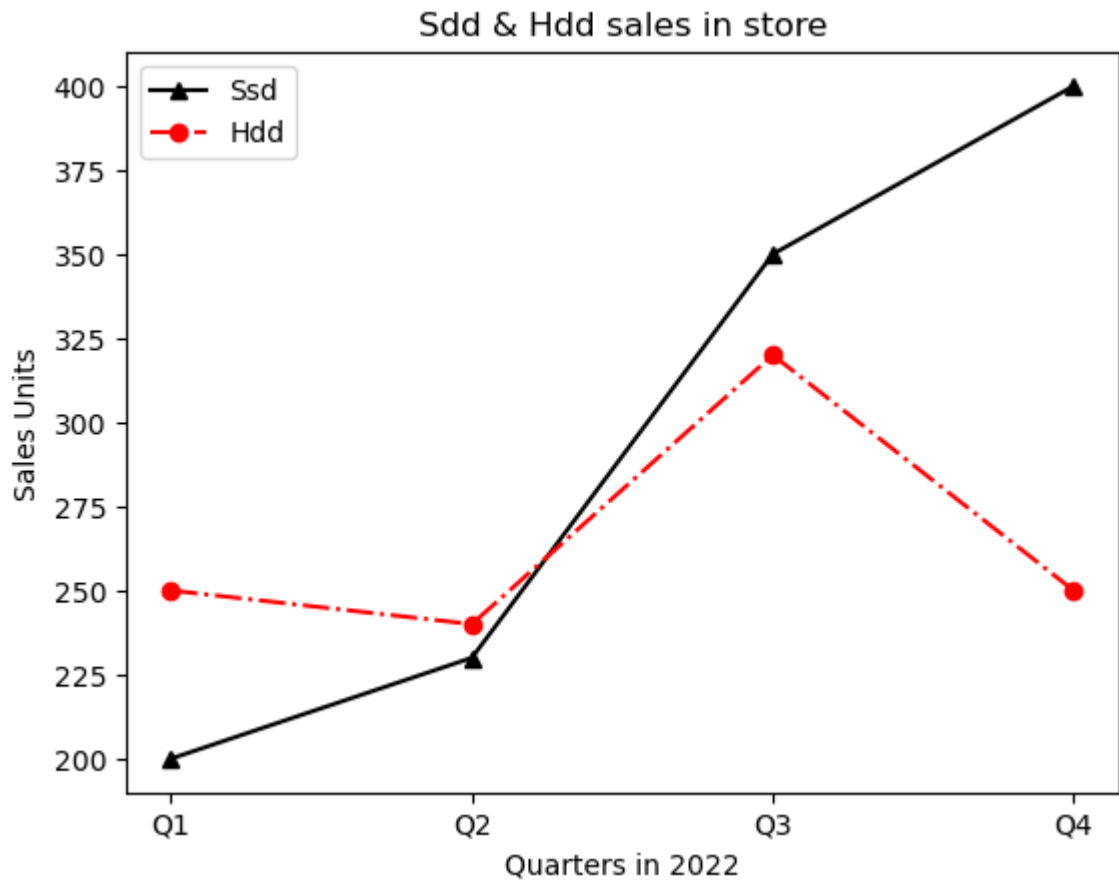
```
In [15]: 1 from matplotlib import pyplot as p
2 pro_na=["Intel","AMD","Snapdragon","Tensor"]
3 use=[200,100,250,50]
4 p.bar(pro_na,use,color='black',width=0.2)
5 p.xlabel("Processors"),p.ylabel("No of Users")
6 p.title("Processor Users in a Community")
7 p.show()
```



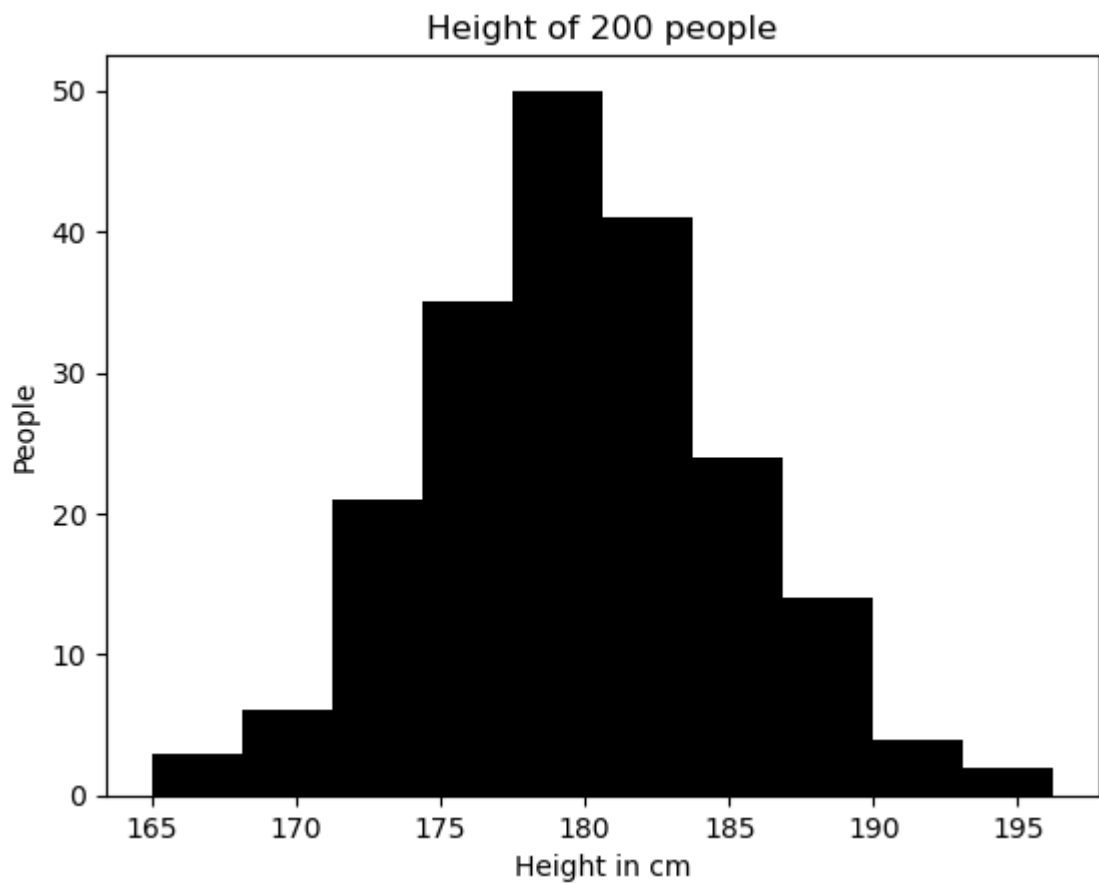
```
In [1]: 1 from matplotlib import pyplot as pi
2 us=[12,32,16,45]
3 la=["Asus","Dell","Lenovo","HP"]
4 e=[0,0,0,0.04]
5 c=["g","c","#8B0000","#473C8B"]
6 pi.pie(us,labels=la,startangle=270,explode=e,colors=c,autopct='%1.2f%%')
7 pi.show()
```



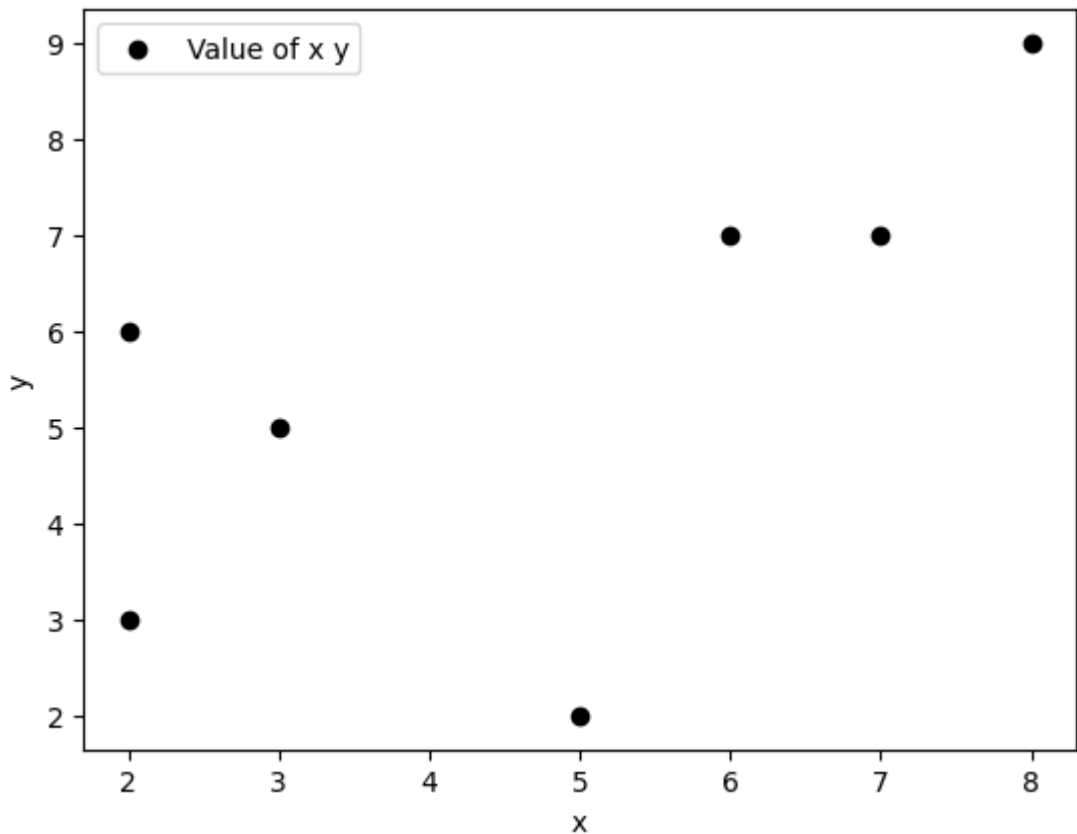
```
In [17]: 1 from matplotlib import pyplot as p
2 Q=["Q1","Q2","Q3","Q4"]
3 ssd=[200,230,350,400]
4 hdd=[250,240,320,250]
5 p.plot(Q,ssd,'^-',color='black')
6 p.plot(Q,hdd,'o-.r')
7 p.xlabel("Quarters in 2022"),p.ylabel("Sales Units")
8 p.title("Sdd & Hdd sales in store")
9 p.legend(['Ssd','Hdd'])
10 p.show()
```



```
In [18]: 1 from matplotlib import pyplot as p
2 import numpy as n
3 x=n.random.normal(180,5,200)
4 p.hist(x,color='k')
5 p.xlabel("Height in cm"),p.ylabel("People")
6 p.title("Height of 200 people")
7 p.show()
```




```
In [19]: 1 from matplotlib import pyplot as p
2 x=[2,6,8,7,3,2,5]
3 y=[6,7,9,7,5,3,2]
4 c=['k','b']
5 p.scatter(x,y,label='Value of x y',color='k')
6 p.xlabel('x')
7 p.ylabel('y')
8 p.legend()
9 p.show()
```



```
In [8]: 1 import pandas as pd
2 import matplotlib.pyplot as plt
3 %matplotlib inline
4 t = pd.read_csv("C:\Users\Anusha V\Downloads\annual-enterprise-survey-2
5 a = t.plot(color='k', linewidth=1)
6 plt.xticks(rotation=25)
7 a.set_ylabel('Temp')
8 plt.xlabel('Date')
9 plt.show()
```

File "C:\Users\Anusha V\AppData\Local\Temp\ipykernel_6172\3817053514.py", line 4

```
t = pd.read_csv("C:\Users\Anusha V\Downloads\annual-enterprise-survey-
2021-financial-year-provisional.csv", parse_dates=['day'], index_col
='day')
```

^

SyntaxError: (unicode error) 'unicodeescape' codec can't decode bytes in position 2-3: truncated \UXXXXXXXX escape

```
In [3]: 1 import pandas as pd
2 from matplotlib import pyplot as plt
3 t = pd.read_csv("C:\Users\Anusha V\Downloads\annual-enterprise-survey-2
4 species_colors = {
5     'Iris-setosa': 'k', 'Iris-versicolor': 'g', 'Iris-virginica': 'r'
6 }
7 for species, color in species_colors.items():
8     sl = t[t['species'] == species]['sepal_length']
9     sw = t[t['species'] == species]['sepal_width']
10    plt.scatter(sl, sw, color=color, label=species)
11    plt.legend()
12    plt.xlabel('Sepal Length')
13    plt.ylabel('Sepal Width')
14    plt.title('Sepal Width and Length for Iris Species')
15    plt.show()
```

File "C:\Users\Anusha V\AppData\Local\Temp\ipykernel_19616\1973929894.py", line 3

```
t = pd.read_csv("C:\Users\Anusha V\Downloads\annual-enterprise-survey-2021-financial-year-provisional-csv.csv")
```

^

SyntaxError: EOL while scanning string literal

```
In [7]: 1 import pandas as pd
2 from matplotlib import pyplot as plt
3 t = pd.read_csv("C:\Users\Anusha V\Downloads\annual-enterprise-survey-2
4 sv=t.groupby("species")["petal_length"].mean()
5 plt.pie(sv, labels=sv.index, startangle=90, autopct="%1.0f%%")
6 plt.title("Petal length of Species Average Visualization")
7 plt.show()
```

File "C:\Users\Anusha V\AppData\Local\Temp\ipykernel_6172\625104607.py", line 3

```
t = pd.read_csv("C:\Users\Anusha V\Downloads\annual-enterprise-survey-2021-financial-year-provisional-csv.csv")
```

^

SyntaxError: (unicode error) 'unicodeescape' codec can't decode bytes in position 2-3: truncated \UXXXXXXXX escape

```
In [6]: 1 import matplotlib.pyplot as plt
2 import pandas as pd
3 data = pd.read_csv("C:\Users\Anusha V\Downloads\annual-enterprise-surve
4 age_survived = data[data['Survived'] == 1]['Age']
5 age_not_survived = data[data['Survived'] == 0]['Age']
6 plt.hist(age_survived, color='g', alpha=0.9, label='Survived')
7 plt.hist(age_not_survived, color='k', alpha=0.5, label='Not Survived')
8 plt.xlabel('Age')
9 plt.ylabel('Frequency')
10 plt.title('Age Distribution of Survived and Not Survived Passengers')
11 plt.legend()
12 plt.show()
```

File "C:\Users\Anusha V\AppData\Local\Temp\ipykernel_6172\2648416779.py", line 3

data = pd.read_csv("C:\Users\Anusha V\Downloads\annual-enterprise-survey-2021-financial-year-provisional-csv.csv")

^

SyntaxError: (unicode error) 'unicodeescape' codec can't decode bytes in position 2-3: truncated \UXXXXXXXX escape

```
In [5]: 1 import pandas as p
2 import matplotlib.pyplot as m
3 d=p.read_csv("C:\Users\Anusha V\Downloads\Annual-enterprise-survey-2021
4 c=d["Pclass"].value_counts()
5 co=['g','r','k']
6 m.bar(c.index,c.values,color=co,width=0.5)
7 m.xticks([1,2,3],["1st class","2nd class","3rd class"])
8 m.xlabel("Classes");m.ylabel("No of Passengers");m.title("No of Passeng
9 m.show()
```

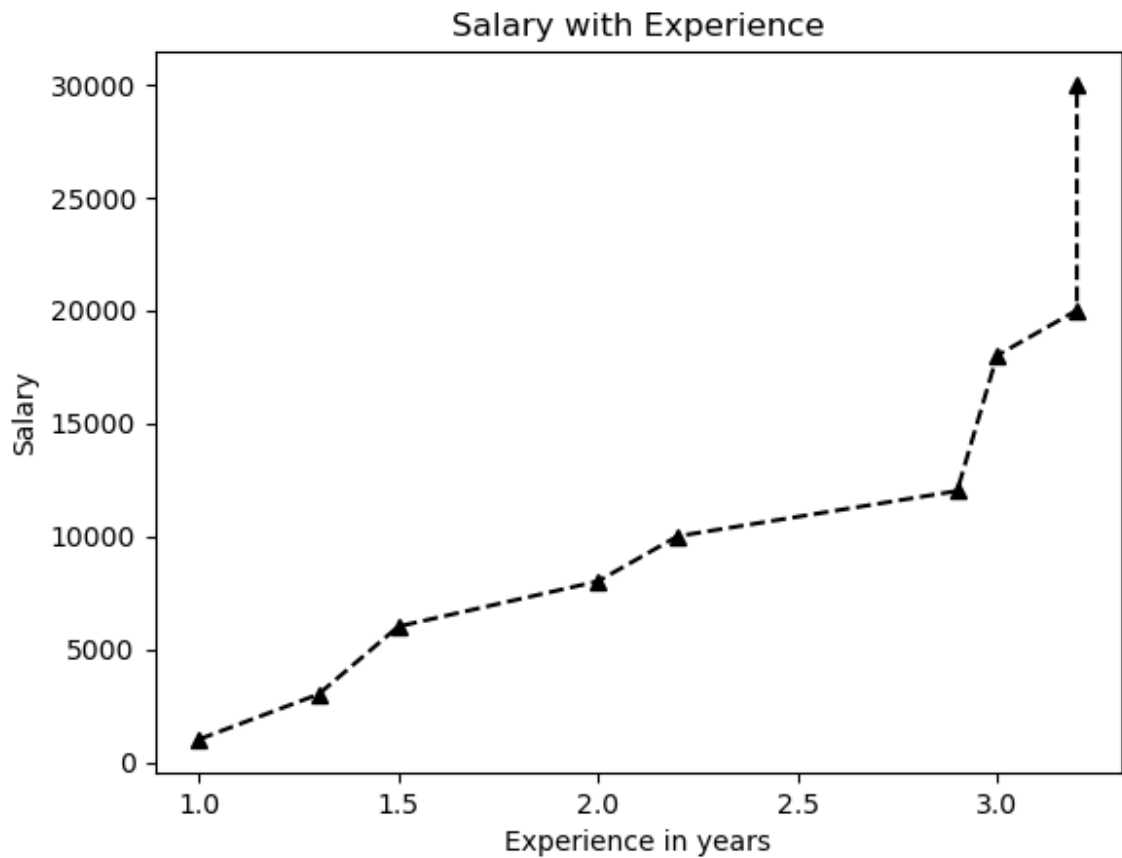
File "C:\Users\Anusha V\AppData\Local\Temp\ipykernel_6172\2465532605.py", line 3

d=p.read_csv("C:\Users\Anusha V\Downloads\Annual-enterprise-survey-2021-financial-year-provisional-csv.csv(1.csv)")

^

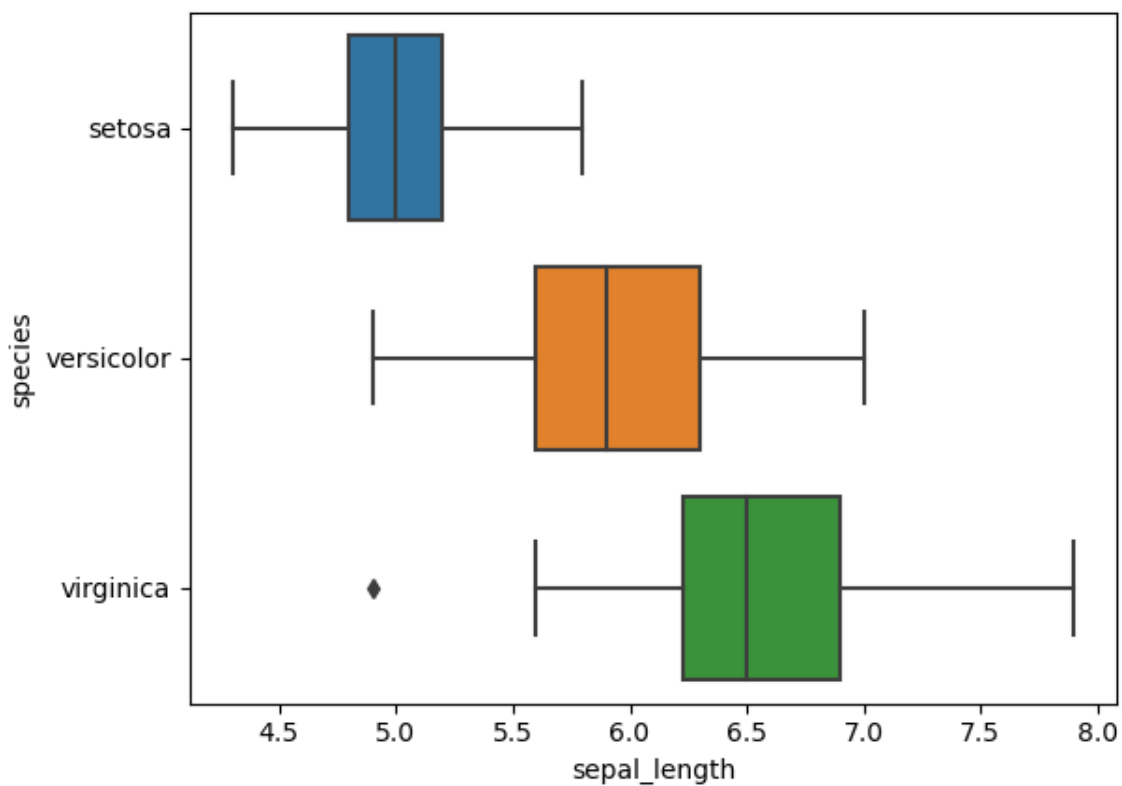
SyntaxError: (unicode error) 'unicodeescape' codec can't decode bytes in position 2-3: truncated \UXXXXXXXX escape

```
In [1]: 1 import pandas as p;import matplotlib.pyplot as m
2 d={"Ex":[1,1.3,1.5,2,2.2,2.9,3,3.2,3.2],
3   "Salary":[1000,3000,6000,8000,10000,12000,18000,20000,30000]}
4 df=p.DataFrame(d)
5 m.plot(df["Ex"],df["Salary"],'^--',color='k')
6 m.xlabel("Experience in years");m.ylabel("Salary");m.title("Salary with
7 m.show()
8 m.savefig("ammu.png")
```



<Figure size 640x480 with 0 Axes>

```
In [26]: 1 import seaborn as s
2 import matplotlib.pyplot as p
3 d=s.load_dataset('iris')
4 s.boxplot(x=d['sepal_length'],y=d['species'])
5 p.show()
6
```



```
In [27]: 1 import pandas as p
2 import matplotlib.pyplot as m
3 d={
4     "First_name":["Aryan","Rohan","Riya","Yash","Siddhant"],
5     "Last_name":["Singh","Agarwal","Shah","Bhatia","Khanna"],
6     "Type":["Full-Time","Itern","Full-Time","Part-Time","Full-Time"],
7     "Dept":["Administration","Technical","Administration","Technical","Man
8     'YoE':[2,3,5,7,6], "Salary":[20000,5000,10000,10000,20000]
9 }
10 df=p.DataFrame(d)
11 av=df.pivot_table(index=['Dept', 'Type'], values='Salary', aggfunc='mea
12 print("Average Salary from ecah dept:\n",av)
13 sm=df.pivot_table(index=['Type'], values='Salary', aggfunc=['sum', 'mea
14 sm.columns=['Total Salary', 'Mean Salary', 'Number of Employees']
15 print("\nSum and Mean of:\n",sm)
16 st=df.pivot_table(values='Salary', index='Type',aggfunc='std')
17 print("\nStandard Deviation:\n",st)
```

Average Salary from ecah dept:

		Salary
Dept	Type	
Administration	Full-Time	15000
Management	Full-Time	20000
Technical	Itern	5000
	Part-Time	10000

Sum and Mean of:

		Total Salary	Mean Salary	Number of Employees
Type				
Full-Time	50000	16666.666667		3
Itern	5000	5000.000000		1
Part-Time	10000	10000.000000		1

Standard Deviation:

		Salary
Type		
Full-Time	5773.502692	

```
In [28]: 1 import pandas as pd
2 a = pd.Series([10, 20, 30, 40, 50])
3 b = pd.Series([40, 50, 60, 70, 80])
4 print("Series A:")
5 print(a)
6 print("\nSeries B:")
7 print(b)
8 non_com = a[~a.isin(b)].tolist() + b[~b.isin(a)].tolist()
9 print("Items not common to both Series:")
10 print(non_com)
11 print("\nSmallest element in Series A:\n", a.min())
12 print("\nLargest element in Series A:\n",a.max())
13 print("\nSum of Series B:\n", b.sum())
14 print("\nAverage of Series A:\n",a.mean())
15 print("\nMedian of Series B:\n", a.median())
```

Series A:

```
0    10
1    20
2    30
3    40
4    50
```

dtype: int64

Series B:

```
0    40
1    50
2    60
3    70
4    80
```

dtype: int64

Items not common to both Series:

```
[10, 20, 30, 60, 70, 80]
```

Smallest element in Series A:

```
10
```

Largest element in Series A:

```
50
```

Sum of Series B:

```
300
```

Average of Series A:

```
30.0
```

Median of Series B:

```
30.0
```

```
In [29]: 1 da={
2         "mpg":[18,15,18,16,17],"cylinders":[8,8,6,4,8],"displacement":[307,350
3         "horsepower":[130,165,150,150,140],"weight":[3504,3693,3436,3433,3449]
4         "acceleration":[12.0,11.5,11.0,12.0,10.5],"model year":[70,71,70,80,70
5         "origin":[1,1,1,1,1],"car name":["cheverlot","buick","plymoth","amc","
6     }
7     df=pd.DataFrame(da)
8     sa=df.describe()
9     ei=df[df["cylinders"]==8]
10    ye = df.groupby('model year')['model year'].count()
11    print("Statistical:\n",sa)
12    print("\n8 cylinders:\n",ei)
13    print("\nBy year:\n",ye)
14
```

Statistical:

	mpg	cylinders	displacement	horsepower	weight \
count	5.000000	5.000000	5.000000	5.000000	5.000000
mean	16.800000	6.800000	316.200000	147.000000	3503.000000
std	1.30384	1.788854	19.879638	13.038405	110.006818
min	15.000000	4.000000	302.000000	130.000000	3433.000000
25%	16.000000	6.000000	304.000000	140.000000	3436.000000
50%	17.000000	8.000000	307.000000	150.000000	3449.000000
75%	18.000000	8.000000	318.000000	150.000000	3504.000000
max	18.000000	8.000000	350.000000	165.000000	3693.000000

	acceleration	model year	origin
count	5.000000	5.000000	5.0
mean	11.400000	72.200000	1.0
std	0.65192	4.38178	0.0
min	10.500000	70.000000	1.0
25%	11.000000	70.000000	1.0
50%	11.500000	70.000000	1.0
75%	12.000000	71.000000	1.0
max	12.000000	80.000000	1.0

8 cylinders:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year \
0	18	8	307	130	3504	12.0	70
1	15	8	350	165	3693	11.5	71
4	17	8	302	140	3449	10.5	70

	origin	car name
0	1	cheverlot
1	1	buick
4	1	ford

By year:

	model year
70	3
71	1
80	1

Name: model year, dtype: int64


```

In [2]: 1 import pandas as p
        2 import matplotlib.pyplot as m
        3 import seaborn as s
        4 # data as 32 Elements
        5 data=p.read_csv("C:\Users\Anusha V\Downloads\annual-enterprise-survey-2
        6 # HISTOGRAM
        7 mpg=data['mpg']
        8 m.hist(mpg,bins='auto',color='k',edgecolor='c')
        9 m.xlabel('Miles per gallon (mpg)');m.ylabel('Frequency')
       10 m.title('Frequency Distribution of mpg')
       11 m.show()
       12 # SCATTER
       13 wt=data['wt']
       14 iv=range(len(data))
       15 m.scatter(iv,mpg,color='k',label='mpg')
       16 m.scatter(iv,wt,color='g',label='wt')
       17 m.title("Relationship b/w Weigth and MPG")
       18 m.legend()
       19 m.show()
       20 # BAR PLOT
       21 c=data['am'].value_counts()
       22 co=['k','g']
       23 m.bar(c.index,c.values,color=co,width=0.3)
       24 m.xticks([0,1],['0-Automatic','1-Manual'])
       25 m.xlabel("Tranmisson Type");m.ylabel("No of Cars")
       26 m.title("Frequency distribution of transmission type of cars")
       27 m.show()
       28 # BOX PLOT
       29 s.boxplot(mpg,color='c')
       30 m.xlabel("MPG");m.ylabel("Values")
       31 m.title("BOX plot of MPG Vlues")

```

File "C:\Users\Anusha V\AppData\Local\Temp\ipykernel_6172\948308472.py", line 5

data=p.read_csv("C:\Users\Anusha V\Downloads\annual-enterprise-survey-2021-financial-year-provisional-csv.csv(1.csv)")

^

SyntaxError: (unicode error) 'unicodeescape' codec can't decode bytes in position 2-3: truncated \UXXXXXXXX escape

```
In [31]: 1 import pandas as p
2 import numpy as n
3 d={"Day":[1,2,3,4,5,6,7,8,9,10],
4   "Steps":[4335,9552,7332,4504,5335,7552,8332,6504,8965,7689]}
5 dp=p.DataFrame(d)
6 dp["+1000 Steps"]=dp["Steps"]+1000
7 fi=dp[dp["+1000 Steps"]>7000]["Day"]
8 print("DataFrame:\n",dp)
9 print("\nDays on which Steps were >7000:\n",fi
```

File "C:\Users\Anusha V\AppData\Local\Temp\ipykernel_17904\3745485220.py", line 9
 print("\nDays on which Steps were >7000:\n",fi

SyntaxError: unexpected EOF while parsing

```
In [32]: 1 import numpy as n
2 import pandas as p
3 import matplotlib.pyplot as m
4 da={
5     'n':[1,2,3,4,5], 'Pencil':[300,350,400,500,520], 'TextBooks':[250,350,400,500,520],
6     'Draw':[100,200,200,250,300], 'Total':[800,1000,1320,1510,2000], "Profits":
7     [500,750,950,1250,1800]}
8 }
9 df=p.DataFrame(da)
10 sta=df.describe()
11 print("Statistics:\n",sta)
12 su=df['Profits'].sum()
13 print("\nSum of Profits:\n",su)
14 mi=df.isna()
15 print("\nMissing values:\n",mi)
16 print("\nMaximum Value:\n",df['Draw'].max())
17 m.plot(df['n'],df['Profits'],'^-',color='k')
18 m.xlabel("Numbers");m.ylabel("Profits")
19 m.show()
20
```

Statistics:

	n	Pencil	TextBooks	Draw	Total	Pr ofits
count	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000
mean	3.000000	414.000000	384.000000	210.000000	1326.000000	11551.200000
std	1.581139	94.762862	92.357999	74.161985	466.669048	3882.152393
min	1.000000	300.000000	250.000000	100.000000	800.000000	8000.000000
25%	2.000000	350.000000	350.000000	200.000000	1000.000000	9500.000000
50%	3.000000	400.000000	400.000000	200.000000	1320.000000	10256.000000
75%	4.000000	500.000000	420.000000	250.000000	1510.000000	12000.000000
max	5.000000	520.000000	500.000000	300.000000	2000.000000	18000.000000

Sum of Profits:

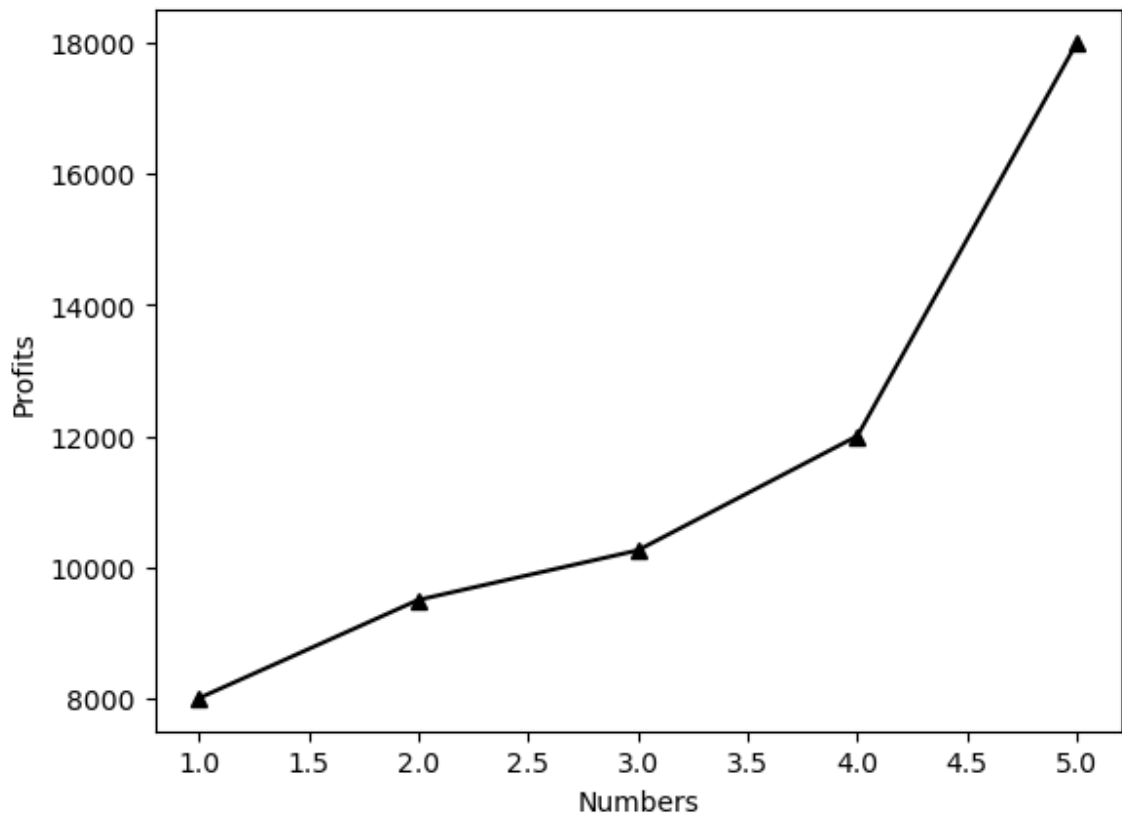
57756

Missing values:

	n	Pencil	TextBooks	Draw	Total	Profits
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	False	False	False	False	False	False
4	False	False	False	False	False	False

Maximum Value:

300



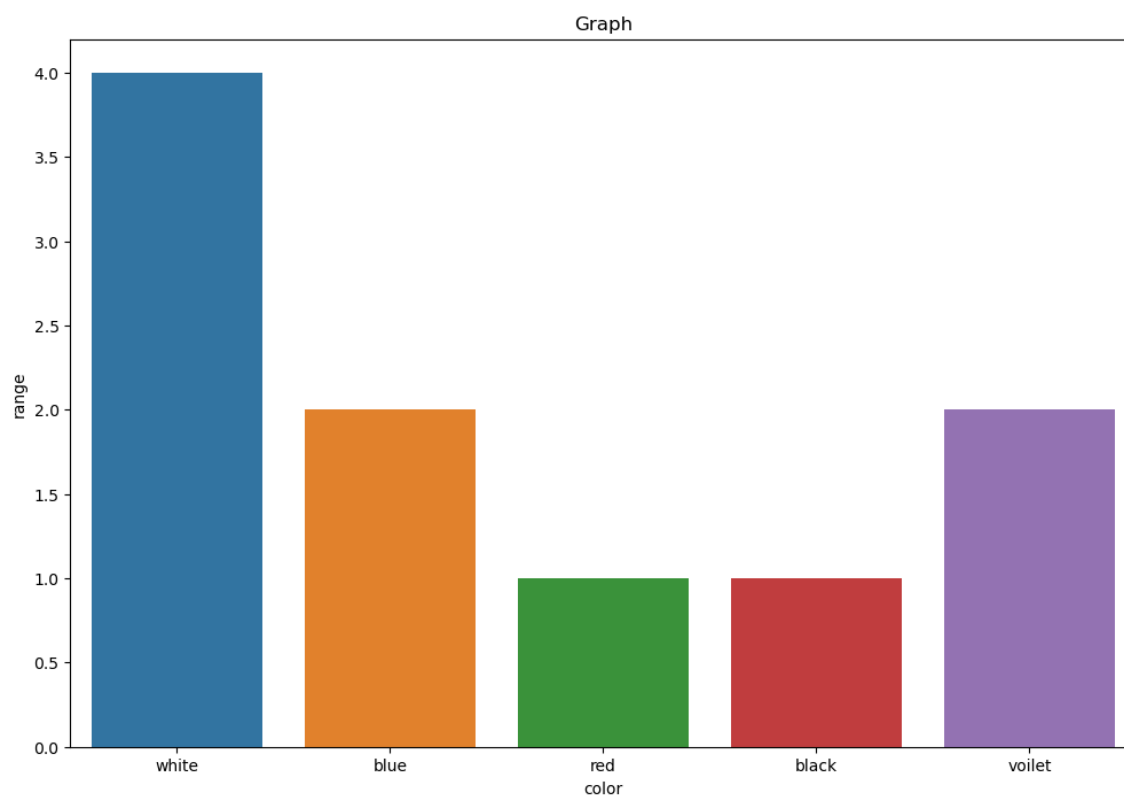
In [4]:

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5 import random
6 def pick(n):
7     result=[]
8     for i in range(1,n+1):
9         result.append(random.choice(['red','blue','voilet','black','whi
10     return result
11 result=pick(10)
12 print(result)
```

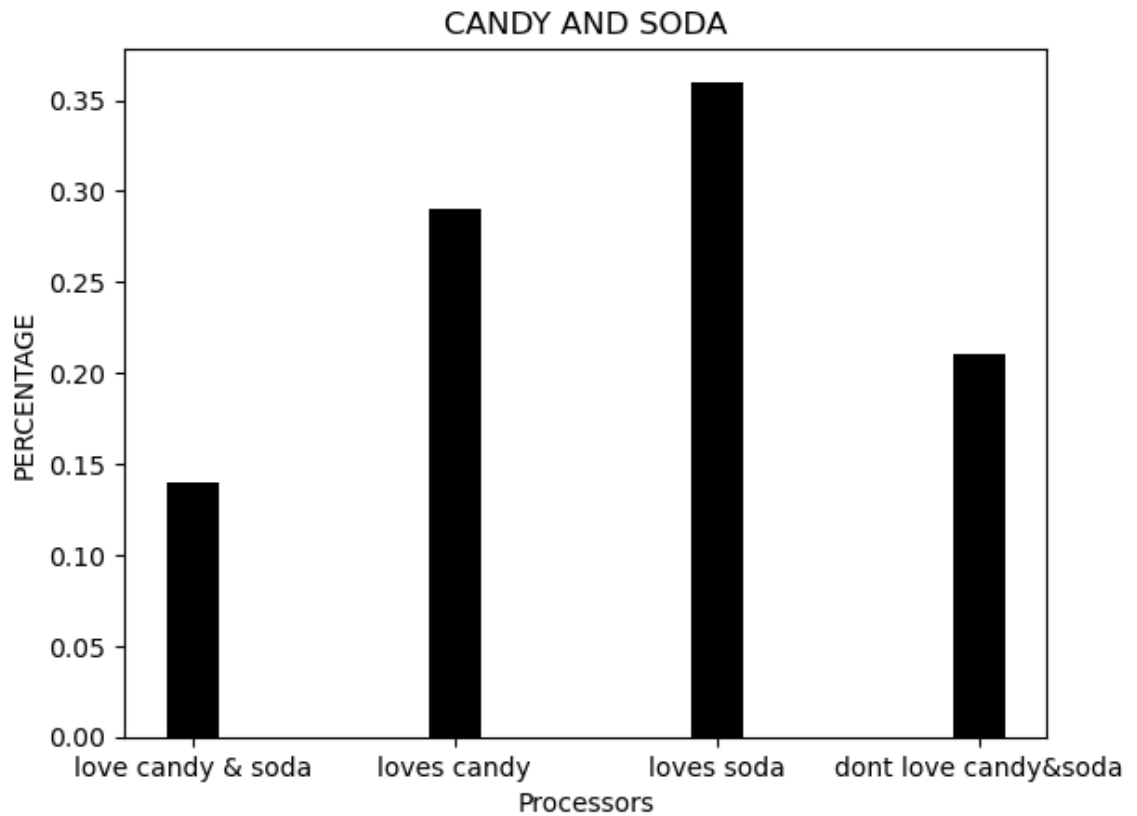
```
['white', 'white', 'blue', 'blue', 'white', 'white', 'red', 'black', 'voil
et', 'voilet']
```

In [5]:

```
1 plt.figure(figsize=(12,8))
2 sns.countplot(x=result)
3
4 plt.title('Graph')
5 plt.xlabel('color')
6 plt.ylabel('range')
7 plt.show()
8
```

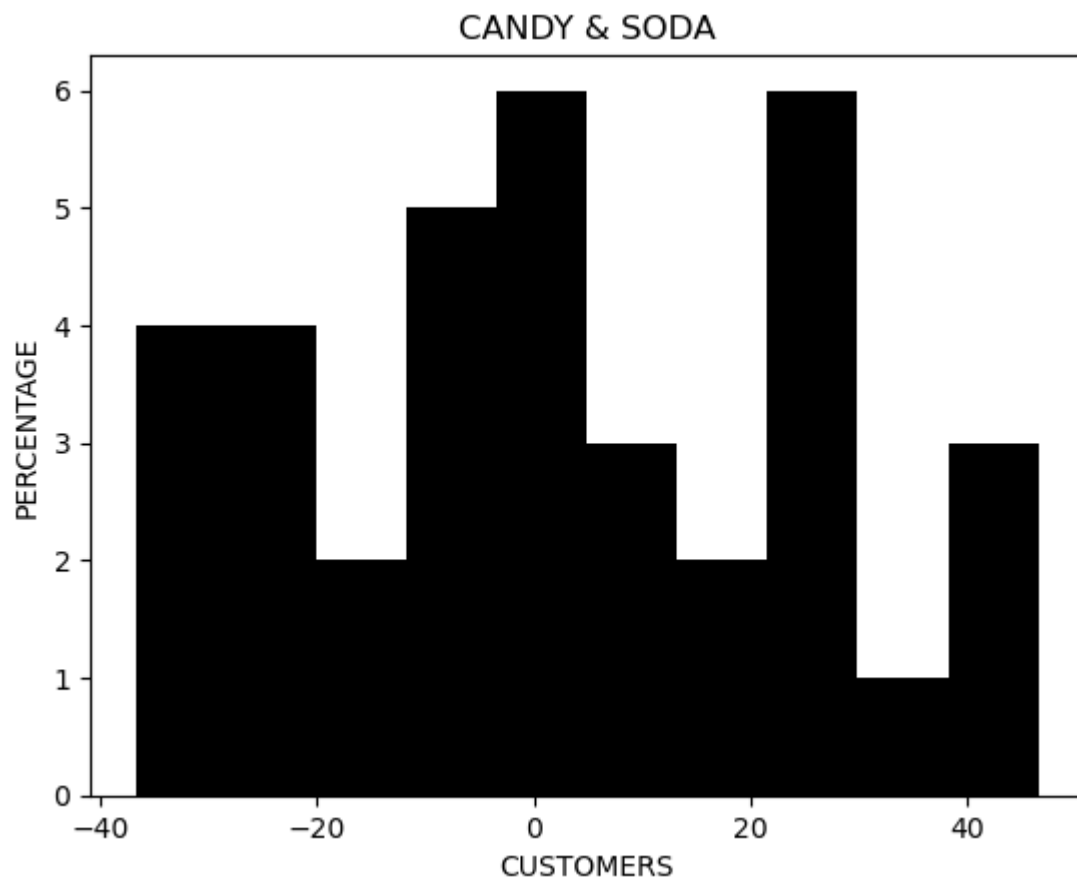


```
In [23]: 1 from matplotlib import pyplot as p
2 pro_na=["love candy & soda","loves candy","loves soda","dont love candy
3 use=[0.14,0.29,0.36,0.21]
4 p.bar(pro_na,use,color='black',width=0.2)
5 p.xlabel("Processors"),p.ylabel("PERCENTAGE")
6 p.title("CANDY AND SODA")
7 p.show()
```

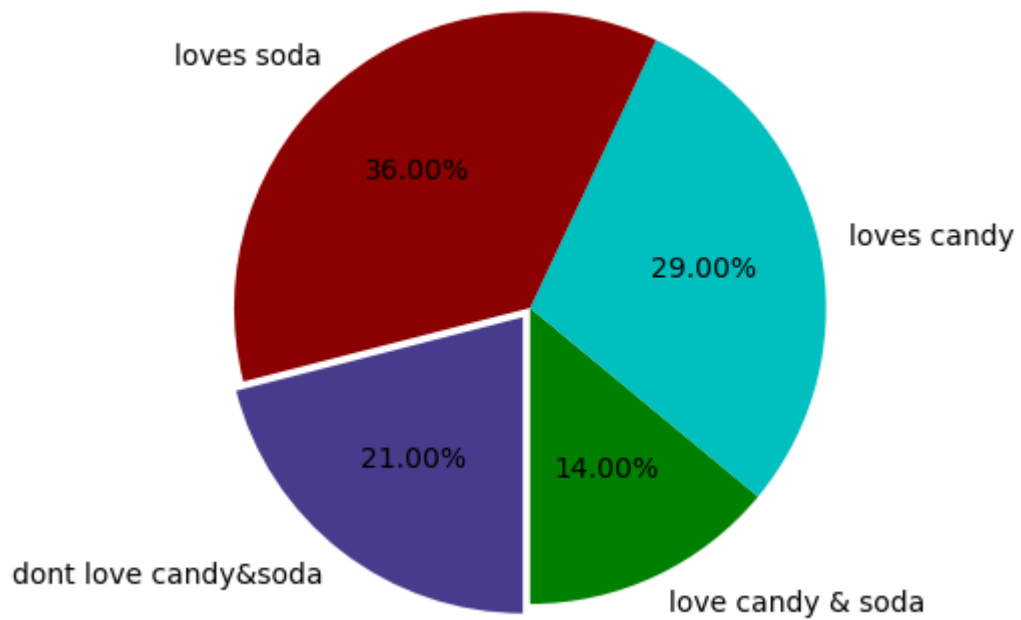


```
In [ ]: 1
```

```
In [31]: 1 from matplotlib import pyplot as p
2 import numpy as n
3 x=n.random.normal(14,29,36)
4 p.hist(x,color='k')
5 p.xlabel("CUSTOMERS"),p.ylabel("PERCENTAGE")
6 p.title("CANDY & SODA")
7 p.show()
```



```
In [32]: 1 from matplotlib import pyplot as pi
2 us=[0.14,0.29,0.36,0.21]
3 la=["love candy & soda","loves candy","loves soda","dont love candy&sod
4 e=[0,0,0,0.04]
5 c=["g","c","#8B0000","#473C8B"]
6 pi.pie(us,labels=la,startangle=270,explode=e,colors=c,autopct='%1.2f%%'
7 pi.show()
```



```
In [ ]: 1
```