250840125020_Python_8

October 10, 2025

[114]:
```
PRNs=[250840125020,250840125052]
print(PRNs)
```

```
[250840125020, 250840125052]
```

1 Assignment 8

Note: use appropriate labels, title and legend for each chart

[26]:
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
```

1.1 Q1.

[27]:
```
df=pd.read_csv('Population.csv')
df.head()
```

[27]:

	Year	Population	Yearly % Change	Yearly Change	Median Age	Fertility Rate	Density	Urban Pop %	Urban Population	Country's Share	World Population
0	1955	387700887	2.29%	8284413	19.7	5.91	130	18.60%	71958495	14.15%	2740213792
1	1960	435990338	2.38%	9657890	19.2	5.92	147	18.50%	80565723	14.46%	3015470894
2	1965	490140146	2.37%	10829962	18.5	5.94	165	19.10%	93493844	14.70%	3334533703
3	1970	545864268	2.18%	11144824	18.1	5.62	184	20.00%	109388950	14.77%	3694683794
4	1975	611309535	2.29%	13089053	18.4	5.20	206	21.70%	132533810	15.02%	4070735277

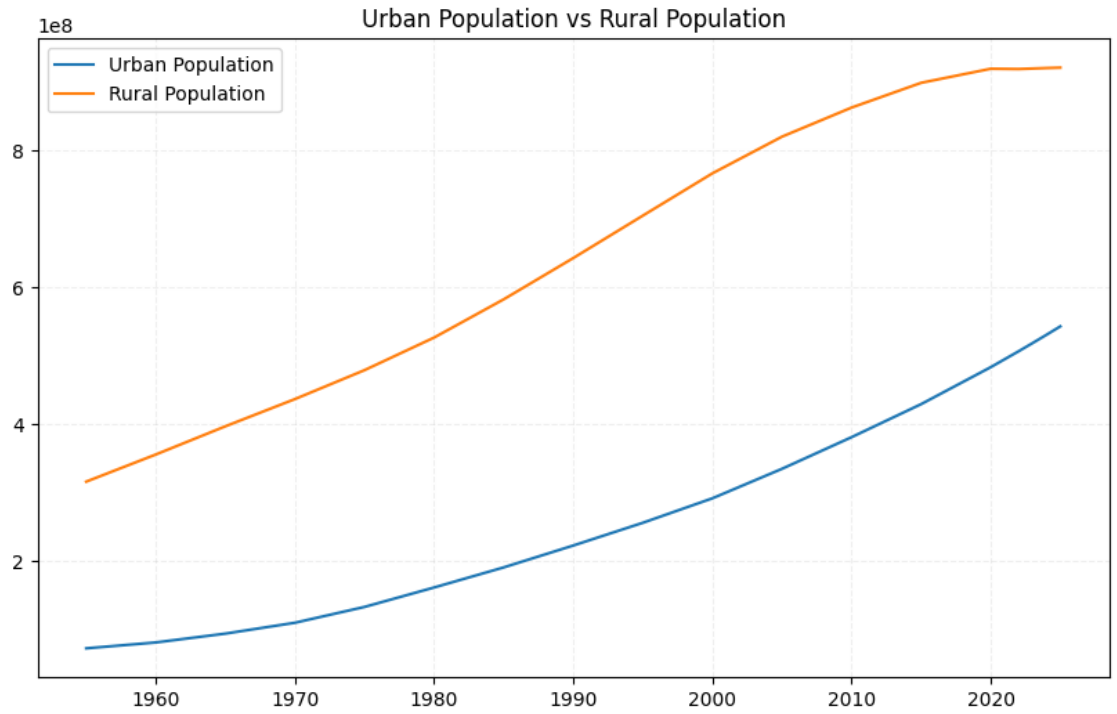
Draw line & bar charts to show urban and rural population of India over the years using population.csv [rural population is the difference between total population and urban population][For bar chart consider 8 yrs data]

[28]:
```
df['Rural Population']=df['Population']-df['Urban Population']
df.head()
```

[28]:

	Year	Population	Yearly % Change	Yearly Change	Median Age	Fertility Rate	Density	Urban Pop %	Urban Population	Country's Share	World Population	Rural Population
0	1955	387700887	2.29%	8284413	19.7	5.91	130	18.60%	71958495	14.15%	2740213792	315742392
1	1960	435990338	2.38%	9657890	19.2	5.92	147	18.50%	80565723	14.46%	3015470894	355424615
2	1965	490140146	2.37%	10829962	18.5	5.94	165	19.10%	93493844	14.70%	3334533703	396646302
3	1970	545864268	2.18%	11144824	18.1	5.62	184	20.00%	109388950	14.77%	3694683794	436475318
4	1975	611309535	2.29%	13089053	18.4	5.20	206	21.70%	132533810	15.02%	4070735277	478775725

[29]:
```
plt.figure(figsize=(10,6))
plt.plot(df['Year'],df['Urban Population'],label='Urban Population')
plt.plot(df['Year'],df['Rural Population'],label='Rural Population')
plt.legend()
plt.title('Urban Population vs Rural Population')
plt.grid(True, linestyle='--', alpha=0.2)
plt.show()
```



[30]:
```
bar_df=df.tail(8)
bar_df
```

[30]:

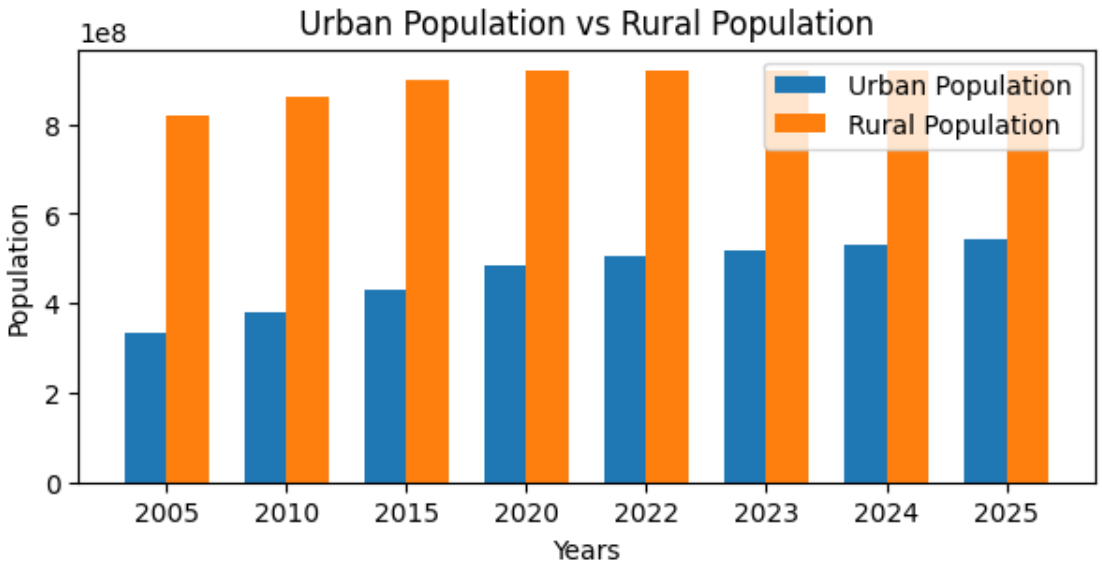
	Year	Population	Yearly % Change	Yearly Change	Median Age	\
10	2005	1154676322	1.77%	19350718	22.2	
11	2010	1243481564	1.49%	17761048	23.6	
12	2015	1328024498	1.32%	16908587	25.3	
13	2020	1402617695	1.10%	14918639	27.0	
14	2022	1425423212	0.81%	11402759	27.7	
15	2023	1438069596	0.89%	12646384	28.1	
16	2024	1450935791	0.89%	12866195	28.4	
17	2025	1463865525	0.89%	12929734	28.8	

	Fertility Rate	Density	Urban Pop %	Urban Population	Country's Share	\
10	2.96	388	29.00%	334479406	17.53%	
11	2.60	418	30.60%	380744554	17.71%	
12	2.29	447	32.30%	429069459	17.78%	
13	2.05	472	34.40%	483098640	17.78%	
14	1.99	479	35.50%	506304869	17.77%	
15	1.98	484	36.00%	518239122	17.77%	
16	1.96	488	36.60%	530387142	17.78%	
17	1.94	492	37.10%	542742539	17.78%	

	World Population	Rural Population

10	6586970132	820196916
11	7021732148	862737010
12	7470491872	898955039
13	7887001292	919519055
14	8021407192	919118343
15	8091734930	919830474
16	8161972572	920548649
17	8231613070	921122986

[31]:
```
bar_width=0.35
x = np.arange(len(bar_df['Year']))
x_urban = x - bar_width/2
x_rural = x + bar_width/2
plt.figure(figsize=(7,3))
plt.title('Urban Population vs Rural Population')
plt.bar(x_urban, bar_df['Urban Population'], bar_width, label='Urban␣
 ↪Population')
plt.bar(x_rural, bar_df['Rural Population'], bar_width, label='Rural␣
 ↪Population')
plt.ylabel('Population')
plt.xlabel('Years')
plt.xticks(x, bar_df['Year'])
plt.legend()
plt.show()
```



1.2 Q2.

Using gapminder.csv show life expectancy of top 10 highly populated countries from Asia

[42]:
```
df=pd.read_csv('gapminder2007.csv')
df.sort_values(by='pop', ascending=False).head(10)
```

[42]:

	country	pop	continent	lifeExp	gdpPercap
24	China	1.318683e+09	Asia	72.961	4959.114854
58	India	1.110396e+09	Asia	64.698	2452.210407
134	United States	3.011399e+08	Americas	78.242	42951.653090
59	Indonesia	2.235470e+08	Asia	70.650	3540.651564
14	Brazil	1.900106e+08	Americas	72.390	9065.800825
97	Pakistan	1.692706e+08	Asia	65.483	2605.947580
8	Bangladesh	1.504483e+08	Asia	64.062	1391.253792
94	Nigeria	1.350312e+08	Africa	46.859	2013.977305
66	Japan	1.274680e+08	Asia	82.603	31656.068060
82	Mexico	1.087009e+08	Americas	76.195	11977.574960

1.3 Q3.

Using a pie chart show distribution of top six selling cars [Create data as required]

[45]:
```
cardata = {
'Model': ['Ford F-150', 'Chevrolet Silverado', 'Toyota Camry', 'Honda CR-V',␣
 ↪'Tesla Model Y',
          'Nissan Rogue', 'GMC Sierra', 'Ram 1500/2500/3500', 'Toyota RAV4',␣
 ↪'Jeep Grand Cherokee'],
'Sales': [787000, 523000, 375000, 361000, 385000, 333000, 289000, 545000,␣
 ↪434000, 259000]}

df = pd.DataFrame(cardata)
df.head()
```

[45]:

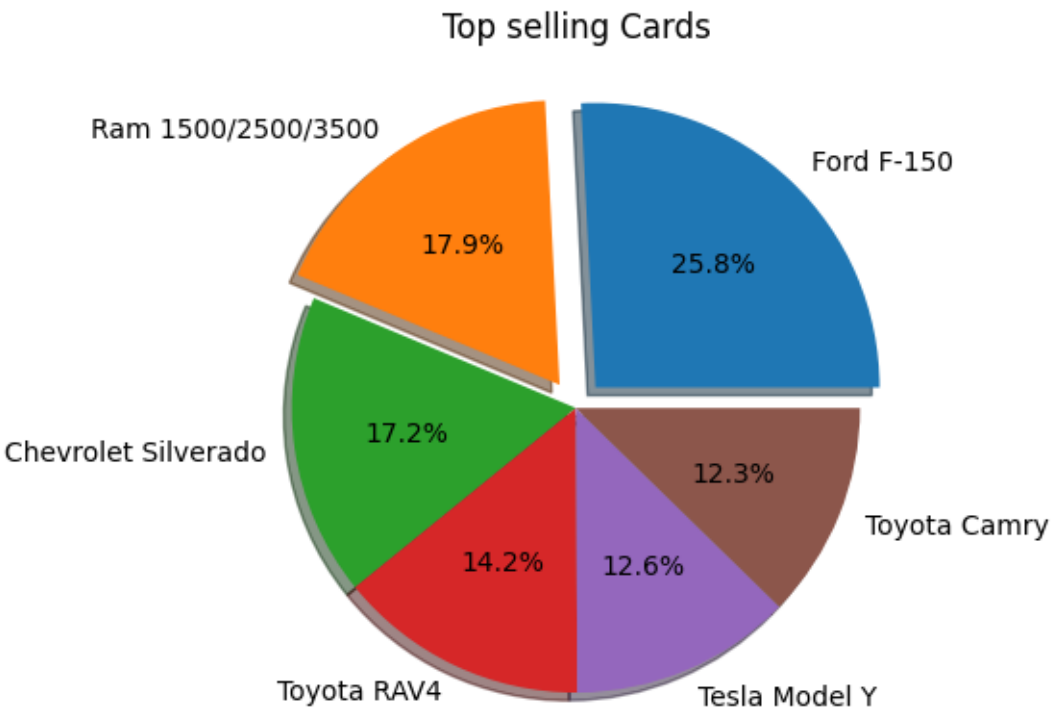
	Model	Sales
0	Ford F-150	787000
1	Chevrolet Silverado	523000
2	Toyota Camry	375000
3	Honda CR-V	361000
4	Tesla Model Y	385000

[47]:
```
top6 = df.sort_values(by='Sales',ascending=False).head(6)
top6
```

[47]:

	Model	Sales
0	Ford F-150	787000
7	Ram 1500/2500/3500	545000
1	Chevrolet Silverado	523000
8	Toyota RAV4	434000
4	Tesla Model Y	385000
2	Toyota Camry	375000

[63]:
```
explode = (0.1, 0.1, 0, 0, 0, 0)
plt.title('Top selling Cards')
plt.pie(top6['Sales'],
        labels=top6['Model'],
        explode=explode,
        autopct='%1.1f%%',
        shadow=True)
plt.show()
```



1.4 Q4.

Using Histogram show distribution of developers by their experience [use survey_data_sample.csv]

[72]:
```
df=pd.read_csv('survey_results_sample.csv')
df.head()
```

[72]:

	ResponseId	Age	YearsCode	CompTotal	ConvertedCompYearly
0	1	18-24 years old	NaN	NaN	NaN
1	2	25-34 years old	18	285000.0	285000.0
2	3	45-54 years old	27	250000.0	250000.0
3	4	25-34 years old	12	156000.0	156000.0
4	5	25-34 years old	6	1320000.0	23456.0

[73]:
```
df['YearsCode'].unique()
```

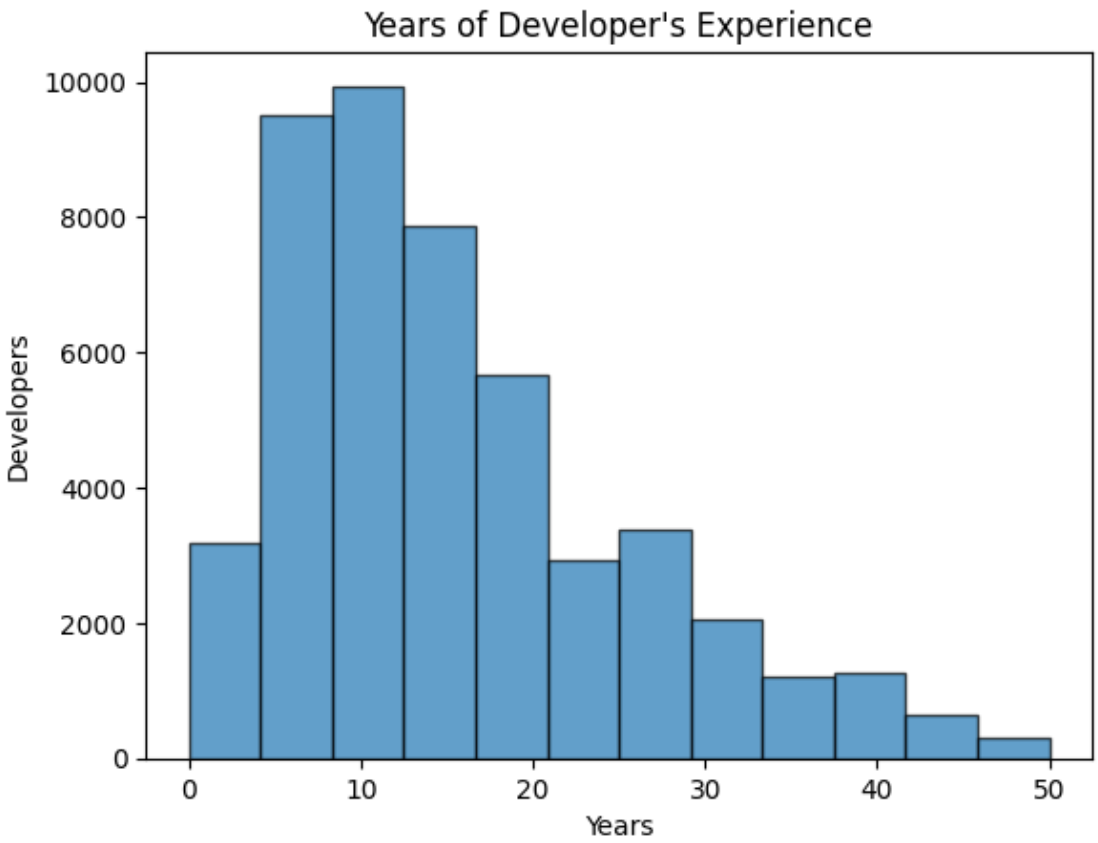
[73]:
```
array([nan, '18', '27', '12', '6', '21', '4', '5', '20', '14', '10', '15',
       '11', '3', '24', '8', '13', 'Less than 1 year', '16', '33', '22',
       '30', '32', '7', '35', '28', '40', '17', '29', '19',
       'More than 50 years', '9', '38', '26', '34', '25', '2', '45', '23',
       '31', '43', '1', '48', '41', '50', '39', '42', '37', '36', '44',
       '46', '49', '47'], dtype=object)
```

[93]:
```
df['YearsCode']=df['YearsCode'].replace('Less than 1 year','0')
df['YearsCode']=df['YearsCode'].replace('More than 50 years','50')
df['YearsCode']=pd.to_numeric(df['YearsCode'], downcast='signed')
df=df.dropna()
df.head()
```

[93]:

	ResponseId	Age	YearsCode	CompTotal	ConvertedCompYearly
1	2	25-34 years old	18	285000.0	285000.0
2	3	45-54 years old	27	250000.0	250000.0
3	4	25-34 years old	12	156000.0	156000.0
4	5	25-34 years old	6	1320000.0	23456.0
5	6	35-44 years old	21	78000.0	96828.0

[111]:
```
plt.hist(df['YearsCode'],
         bins=12,
         edgecolor='black',
         alpha=0.7)
plt.title("Years of Developer's Experience")
plt.xlabel("Years")
plt.ylabel("Developers")
plt.show()
```


Years of Developer's Experience
Developers
Years
0
2000
4000
6000
8000
10000
0
10
20
30
40
50

[ ]: