Assignment on Store_sales dataset

Load the Dataset

In [1]:
```
import pandas as pd
ss=pd.read_csv("store_sales.csv")
type(ss)
```

Out[1]:
```
pandas.core.frame.DataFrame
```

In [3]:
```
ss.describe
```

Out[3]:
```
<bound method NDFrame.describe of    store_id        city  Jan  Feb  Mar  Apr  May  Jun  July  Aug  Sep  Oct  \
0       S_1       Texas    8   20   13   21   17   20    24   17   16    9
1       S_2  California   12   19   15   15   11   19     7   15   10   11
2       S_3  California   16   16   14   19   23    6    13   13   15   14
3       S_4       Texas    8   18   13   10   14   14     6    8    8   18
4       S_5       Texas   19    5   24    9    5   24    10    5   24   15
..      ...         ...  ...  ...  ...  ...  ...  ...   ...  ...  ...  ...
95     S_96       Texas    7   10   20   20   10   15    15   21   15    7
96     S_97  California   13    6    7   15   22   10    21   23   10    6
97     S_98       Texas   16    9    6   14   20   13    11   10    8   22
98     S_99     Arizona   18   16    9    5   12   22    11   13   21   17
99    S_100  California    5   23   17   24   15   21    19   10   12   20

    Nov  Dec
0     7    6
1    21   19
2    24    8
3     7   11
4     6   13
..  ...  ...
95   23   22
96   12    9
97   17   22
98   19   10
99    5    9

[100 rows x 14 columns]>
```

In [4]:
```
ss.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   store_id  100 non-null    object
 1   city      100 non-null    object
 2   Jan       100 non-null    int64 
 3   Feb       100 non-null    int64 
 4   Mar       100 non-null    int64 
 5   Apr       100 non-null    int64 
 6   May       100 non-null    int64 
 7   Jun       100 non-null    int64 
 8   July      100 non-null    int64 
 9   Aug       100 non-null    int64 
 10  Sep       100 non-null    int64 
 11  Oct       100 non-null    int64 
 12  Nov       100 non-null    int64 
 13  Dec       100 non-null    int64 
dtypes: int64(12), object(2)
memory usage: 11.1+ KB
```

Calculate the total sales for each store across all months.

In [5]:
```
ss['Total_Sales'] = ss.iloc[:, 2:].sum(axis=1)

print(ss[['store_id', 'Total_Sales']])
```

```
   store_id  Total_Sales
0       S_1          178
1       S_2          174
2       S_3          181
3       S_4          135
4       S_5          159
..      ...          ...
95     S_96          185
96     S_97          154
97     S_98          168
98     S_99          173
99    S_100          180

[100 rows x 2 columns]
```

Find the average sales for each month across all stores

In [6]:
```
monthly_avg_sales = ss.iloc[:, 2:].mean()
print(monthly_avg_sales)
```

```
Jan            14.46
Feb            15.09
Mar            14.56
Apr            14.57
May            13.56
Jun            13.80
July           14.38
Aug            15.81
Sep            14.91
Oct            14.10
Nov            15.31
Dec            13.57
Total_Sales   174.12
dtype: float64
```

Identify the store with the highest total sales.

In [7]:
```
max_sales_store = ss.loc[ss['Total_Sales'].idxmax()]
print(max_sales_store[['store_id', 'Total_Sales']])
```

```
store_id       S_62
Total_Sales     214
Name: 61, dtype: object
```

Calculate the total sales for each city.

In [8]:
```
city_sales = ss.groupby('city')['Total_Sales'].sum()
print(city_sales)
```

```
city
Arizona       3951
California    7522
Texas         5939
Name: Total_Sales, dtype: int64
```

List stores with total sales greater than 200

In [9]:
```
stores_above_200 = ss[ss['Total_Sales'] > 200]
print(stores_above_200[['store_id', 'Total_Sales']])
```

```
   store_id  Total_Sales
37     S_38          207
39     S_40          204
50     S_51          210
61     S_62          214
69     S_70          206
72     S_73          213
77     S_78          211
85     S_86          214
86     S_87          203
92     S_93          204
```

Which month had the highest average sales across all stores?

In [10]:
```
highest_avg_month = monthly_avg_sales.idxmax()
print(highest_avg_month)
```

```
Total_Sales
```

In [ ]:
```

```

Which city generated the most revenue?

In [11]:
```
max_revenue_city = city_sales.idxmax()
print(max_revenue_city)
```

```
California
```

In [ ]:
```

```