In [3]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn  as sns
```

In [5]:

```
df = pd.read_csv(r'E:\Data Sets -Data Frames\dataset_1_202503101237.csv')
```

In [7]:

```
df.head()
```

Out[7]:

	destination	passanger	weather	temperature	time	coupon	expiration	gende
0	No Urgent Place	Alone	Sunny	55	2PM	Restaurant(<20)	1d	Femal
1	No Urgent Place	Friend(s)	Sunny	80	10AM	Coffee House	2h	Femal
2	No Urgent Place	Friend(s)	Sunny	80	10AM	Carry out & Take away	2h	Femal
3	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	2h	Femal
4	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	1d	Femal

5 rows × 27 columns

In [36]:

```
# dislay only two rows weather and temperature
df[['weather','temperature']].head()
```

Out[36]:

	weather	temperature
0	Sunny	55
1	Sunny	80
2	Sunny	80
3	Sunny	80
4	Sunny	80

In [38]:

```
df.head(10) #display 10 rows
```

Out[38]:

	destination	passanger	weather	temperature	time	coupon	expiration	gende
0	No Urgent Place	Alone	Sunny	55	2PM	Restaurant(<20)	1d	Femal
1	No Urgent Place	Friend(s)	Sunny	80	10AM	Coffee House	2h	Femal
2	No Urgent Place	Friend(s)	Sunny	80	10AM	Carry out & Take away	2h	Femal
3	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	2h	Femal
4	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	1d	Femal
5	No Urgent Place	Friend(s)	Sunny	80	6PM	Restaurant(<20)	2h	Femal
6	No Urgent Place	Friend(s)	Sunny	55	2PM	Carry out & Take away	1d	Femal
7	No Urgent Place	Kid(s)	Sunny	80	10AM	Restaurant(<20)	2h	Femal
8	No Urgent Place	Kid(s)	Sunny	80	10AM	Carry out & Take away	2h	Femal
9	No Urgent Place	Kid(s)	Sunny	80	10AM	Bar	1d	Femal

10 rows × 27 columns

In [ ]:

In [52]:

```
df['passanger'].unique() # it will diplay all category of passanger i the passanger
```

Out[52]:

```
array(['Alone', 'Friend(s)', 'Kid(s)', 'Partner'], dtype=object)
```

In [58]:

```
df['passanger'].nunique()# it will display the total count of passanger in the pass
```

Out[58]:

```
4
```

In [67]:

```
df[df['destination']=='Home'].head()     # condtion destination ==  home
```

Out[67]:

	destination	passanger	weather	temperature	time	coupon	expiration	gender
13	Home	Alone	Sunny	55	6PM	Bar	1d	Female
14	Home	Alone	Sunny	55	6PM	Restaurant(20-50)	1d	Female
15	Home	Alone	Sunny	80	6PM	Coffee House	2h	Female
35	Home	Alone	Sunny	55	6PM	Bar	1d	Male
36	Home	Alone	Sunny	55	6PM	Restaurant(20-50)	1d	Male

5 rows × 27 columns

In [71]:

```python
df.sort_values('coupon')
```

Out[71]:

	destination	passanger	weather	temperature	time	coupon	expiration	g
11702	Home	Partner	Sunny	30	10PM	Bar	2h	F
9930	No Urgent Place	Alone	Snowy	30	2PM	Bar	1d	F
10632	Home	Alone	Rainy	55	6PM	Bar	1d	
7997	No Urgent Place	Friend(s)	Rainy	55	10PM	Bar	2h	
11166	Work	Alone	Snowy	30	7AM	Bar	1d	F
...	...	...	...	...	...	...	...	
10476	Home	Alone	Sunny	80	6PM	Restaurant(<20)	1d	F
5447	Home	Alone	Sunny	80	10PM	Restaurant(<20)	2h	F
10478	Home	Alone	Snowy	30	10PM	Restaurant(<20)	2h	F
5440	No Urgent Place	Alone	Sunny	80	2PM	Restaurant(<20)	2h	F
0	No Urgent Place	Alone	Sunny	55	2PM	Restaurant(<20)	1d	F

12684 rows × 27 columns

In [73]:
```
df.sort_values('temperature')
```

Out[73]:

	destination	passanger	weather	temperature	time	coupon	expiration	g
7842	No Urgent Place	Friend(s)	Sunny	30	10PM	Restaurant(<20)	2h	F
11460	No Urgent Place	Friend(s)	Snowy	30	2PM	Restaurant(<20)	1d	F
11459	No Urgent Place	Friend(s)	Sunny	30	10AM	Carry out & Take away	2h	F
8750	Home	Alone	Sunny	30	6PM	Carry out & Take away	2h	F
8751	Home	Alone	Snowy	30	6PM	Coffee House	1d	F
...	...	...	...	...	...	...	...	
5117	Home	Alone	Sunny	80	10PM	Restaurant(<20)	2h	F
5118	Home	Alone	Sunny	80	10PM	Carry out & Take away	1d	F
5119	Home	Alone	Sunny	80	10PM	Bar	1d	F
5121	Work	Alone	Sunny	80	7AM	Coffee House	2h	F
12683	Work	Alone	Sunny	80	7AM	Restaurant(20-50)	2h	

12684 rows × 27 columns

In [87]:
```
df.rename(columns={'destination':'Destinations'},inplace =True)
```

In [109…
```
df.groupby('occupation').size().to_frame('Count').reset_index()
```

Out[109…

	occupation	Count
0	Architecture & Engineering	175
1	Arts Design Entertainment Sports & Media	629
2	Building & Grounds Cleaning & Maintenance	44
3	Business & Financial	544
4	Community & Social Services	241
5	Computer & Mathematical	1408
6	Construction & Extraction	154
7	Education&Training&Library	943
8	Farming Fishing & Forestry	43
9	Food Preparation & Serving Related	298
10	Healthcare Practitioners & Technical	244
11	Healthcare Support	242
12	Installation Maintenance & Repair	133
13	Legal	219
14	Life Physical Social Science	170
15	Management	838
16	Office & Administrative Support	639
17	Personal Care & Service	175
18	Production Occupations	110
19	Protective Service	175
20	Retired	495
21	Sales & Related	1093
22	Student	1584
23	Transportation & Material Moving	218
24	Unemployed	1870

In [111…

```
df.groupby('occupation').size().to_frame('Count').reset_index()
```

Out[111…

	occupation	Count
0	Architecture & Engineering	175
1	Arts Design Entertainment Sports & Media	629
2	Building & Grounds Cleaning & Maintenance	44
3	Business & Financial	544
4	Community & Social Services	241
5	Computer & Mathematical	1408
6	Construction & Extraction	154
7	Education&Training&Library	943
8	Farming Fishing & Forestry	43
9	Food Preparation & Serving Related	298
10	Healthcare Practitioners & Technical	244
11	Healthcare Support	242
12	Installation Maintenance & Repair	133
13	Legal	219
14	Life Physical Social Science	170
15	Management	838
16	Office & Administrative Support	639
17	Personal Care & Service	175
18	Production Occupations	110
19	Protective Service	175
20	Retired	495
21	Sales & Related	1093
22	Student	1584
23	Transportation & Material Moving	218
24	Unemployed	1870

In [116…

```
# Find the average of weather based on temperature :
df.groupby('weather')['temperature'].mean().to_frame('Count').reset_index()
```

Out[116…

	weather	Count
0	Rainy	55.000000
1	Snowy	30.000000
2	Sunny	68.946271

In [120…

```
df.groupby('weather')['temperature'].size().to_frame('Count_temp').reset_index()
```

Out[120…

	weather	Count_temp
0	Rainy	1210
1	Snowy	1405
2	Sunny	10069

In [126…

```
df.groupby('weather')['temperature'].size().to_frame('Count_temp').reset_index
```

Out[126…

```
<bound method DataFrame.reset_index of          Count_temp
weather
Rainy          1210
Snowy          1405
Sunny         10069>
```

In [130…

```
df.groupby('weather')['temperature'].sum().to_frame('Sum_temp').reset_index()
```

Out[130…

	weather	Sum_temp
0	Rainy	66550
1	Snowy	42150
2	Sunny	694220

In [132…

```
df.groupby('weather')['temperature'].min().to_frame('Min_temp').reset_index()
```

Out[132…

	weather	Min_temp
0	Rainy	55
1	Snowy	30
2	Sunny	30

In [154…

```
df.groupby('weather')['temperature'].max().to_frame().reset_index()
```

Out[154…

	weather	temperature
0	Rainy	55
1	Snowy	30
2	Sunny	80

In [156…

```
df.groupby('occupation').filter(lambda x: x['occupation'].iloc[0] == 'Student').gro
```

Out[156…

```
occupation
Student    1584
dtype: int64
```

In [162…

```
df[df['occupation'] == 'Student'].shape[0]
```

Out[162…

```
1584
```

In [168…

```
df.iloc[1]  iloc is used to for indexing in particular  row
```

Out[168…

```
Destinations                       No Urgent Place
passanger                                Friend(s)
weather                                      Sunny
temperature                                     80
time                                          10AM
coupon                                Coffee House
expiration                                      2h
gender                                      Female
age                                             21
maritalStatus                    Unmarried partner
has_children                                     1
education                  Some college - no degree
occupation                              Unemployed
income                             $37500 - $49999
car                                            NaN
Bar                                          never
CoffeeHouse                                  never
CarryAway                                      NaN
RestaurantLessThan20                           4~8
Restaurant20To50                               1~3
toCoupon_GEQ5min                                 1
toCoupon_GEQ15min                                0
toCoupon_GEQ25min                                0
direction_same                                   0
direction_opp                                    1
Y                                                0
row_count                                        2
Name: 1, dtype: object
```

In [184…

```
pd.concat([df, df1])['destination'].drop_duplicates()
```

Out[184…

```
0                 NaN
0                Home
1     No Urgent Place
2               UNION
3                Work
Name: destination, dtype: object
```

In [176…

```
df1 = pd.read_csv(r'E:\Data Sets -Data Frames\table_to_union.csv')
```

In [178…

```
df1
```

Out[178…

	destination
0	Home
1	No Urgent Place
2	UNION
3	Work

In [ ]:

```
```

In [196…

```
df2 = pd.read_csv(r'E:\Data Sets -Data Frames\table_to_join.csv')
```

In [198…

```
df2
```

Out[198…

	time	part_of_day
0	2PM	Afternoon
1	10AM	Morning
2	6PM	Evening
3	7AM	Morning
4	10PM	Night

In [206…

```
pd.merge(df, df2[['time', 'part_of_day']], on='time', how='inner')[['Destinations',
```

Out[206…

	Destinations	time	part_of_day
0	No Urgent Place	2PM	Afternoon
1	No Urgent Place	10AM	Morning
2	No Urgent Place	10AM	Morning
3	No Urgent Place	2PM	Afternoon
4	No Urgent Place	2PM	Afternoon
...	...	...	...
12679	Home	6PM	Evening
12680	Work	7AM	Morning
12681	Work	7AM	Morning
12682	Work	7AM	Morning
12683	Work	7AM	Morning

12684 rows × 3 columns

In [204…

```
df.columns
```

Out[204…

```
Index(['Destinations', 'passanger', 'weather', 'temperature', 'time', 'coupon',
       'expiration', 'gender', 'age', 'maritalStatus', 'has_children',
       'education', 'occupation', 'income', 'car', 'Bar', 'CoffeeHouse',
       'CarryAway', 'RestaurantLessThan20', 'Restaurant20To50',
       'toCoupon_GEQ5min', 'toCoupon_GEQ15min', 'toCoupon_GEQ25min',
       'direction_same', 'direction_opp', 'Y', 'row_count'],
      dtype='object')
```

In [213…

```
df[df['passanger'] == 'Alone'][['Destinations', 'passanger']]
```

Out[213…

	Destinations	passanger
0	No Urgent Place	Alone
13	Home	Alone
14	Home	Alone
15	Home	Alone
16	Work	Alone
...	...	...
12676	Home	Alone
12680	Work	Alone
12681	Work	Alone
12682	Work	Alone
12683	Work	Alone

7305 rows × 2 columns

In [217…

```
df[df['weather'].str.startswith('Sun')]
```

Out[217…

	Destinations	passanger	weather	temperature	time	coupon	expiration
0	No Urgent Place	Alone	Sunny	55	2PM	Restaurant(<20)	1d
1	No Urgent Place	Friend(s)	Sunny	80	10AM	Coffee House	2h
2	No Urgent Place	Friend(s)	Sunny	80	10AM	Carry out & Take away	2h
3	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	2h
4	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	1d
...	...	...	...	...	...	...	...
12673	Home	Alone	Sunny	30	6PM	Carry out & Take away	1d
12676	Home	Alone	Sunny	80	6PM	Restaurant(20-50)	1d
12677	Home	Partner	Sunny	30	6PM	Restaurant(<20)	1d
12678	Home	Partner	Sunny	30	10PM	Restaurant(<20)	2h
12683	Work	Alone	Sunny	80	7AM	Restaurant(20-50)	2h

10069 rows × 27 columns

In [225…

```
df[(df['temperature'] >= 29) & (df['temperature'] <= 75)]['temperature'].nunique()
```

Out[225…

```
2
```

In [227…

```
df[df['occupation'].isin(['Sales & Related', 'Management'])][['occupation']]
```

```
Out[227…
             occupation
  193    Sales & Related
  194    Sales & Related
  195    Sales & Related
  196    Sales & Related
  197    Sales & Related
  ...                ...
12679    Sales & Related
12680    Sales & Related
12681    Sales & Related
12682    Sales & Related
12683    Sales & Related

1931 rows × 1 columns
```

```
In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:
```

```
In [ ]:

In [ ]:

In [ ]:

In [ ]:
```