

```
In [10]: import pandas as pd
import matplotlib as mpl
import seaborn as sb
```

```
In [25]: #read the csv files
passengerDF = pd.read_csv('den_passengers.csv')
cannabisDF = pd.read_csv('2016_sales.csv')
```

```
In [31]: passengerDF.head()
```

```
Out[31]:
```

	Month	American	Cargo	Delta	Frontier	Other	Southwest	United	United Express
0	1995-02-01	5833.0	0.0	4805.0	1434.0	8625.0	0.0	47230.0	2635.0
1	1995-03-01	228196.0	0.0	200964.0	36703.0	245171.0	0.0	1962445.0	169190.0
2	1995-04-01	184229.0	0.0	157853.0	40050.0	191393.0	0.0	1724836.0	124351.0
3	1995-05-01	187990.0	0.0	162622.0	41403.0	171425.0	0.0	1833305.0	135800.0
4	1995-06-01	229348.0	0.0	198499.0	43754.0	186030.0	0.0	1975754.0	170033.0

```
In [33]: cannabisDF.head()
```

```
Out[33]:
```

	Month	Year	County	Medical	Retail	Sales
0	2016-2016-2016-2016-2016-2016-02	2016	Adams	363668.0	2544485.0	2908153.0
1	2016-2016-2016-2016-2016-2016-02	2016	Alamosa	NaN	NaN	0.0
2	2016-2016-2016-2016-2016-2016-02	2016	Arapahoe	1547734.0	5949573.0	7497307.0
3	2016-2016-2016-2016-2016-2016-02	2016	Archuleta	NaN	NaN	0.0
4	2016-2016-2016-2016-2016-2016-02	2016	Boulder	2565967.0	4499880.0	7065847.0

```
In [52]: cannabis_sales2016 = pd.read_csv('2016_sales.csv')
# Filter for Boulder County
boulder_sales = cannabis_sales2016[cannabis_sales2016['County'] == 'Boulder']
boulder_sales['Month'] = boulder_sales['Year'].astype(str) + '-' + boulder_s
# Calculate total sales (Medical + Retail) for Boulder
boulder_sales['Total_Sales'] = boulder_sales[['Medical', 'Retail']].sum(axis
# Calculate average sale by month across Boulder County
average_sales_boulder16 = boulder_sales.groupby('Month')['Total_Sales'].mean
average_sales_boulder16
```

```
/tmp/ipykernel_169/3121378794.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
boulder_sales['Month'] = boulder_sales['Year'].astype(str) + '-' + boulder_sales['Month'].apply(lambda x: f"{x:02}")
```

```
/tmp/ipykernel_169/3121378794.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
boulder_sales['Total_Sales'] = boulder_sales[['Medical', 'Retail']].sum(axis=1)
```

Out [52]:

	Month	Total_Sales
0	2016-01	7065847.0
1	2016-02	6994921.0
2	2016-03	7856453.0
3	2016-04	7788867.0
4	2016-05	7751847.0
5	2016-06	7812331.0
6	2016-07	8653983.0
7	2016-08	8766671.0
8	2016-09	8644495.0
9	2016-10	8196584.0
10	2016-11	7543966.0
11	2016-12	8332161.0

```
In [49]: cannabis_sales2017 = pd.read_csv('2017_sales.csv')
# Filter for Boulder County
boulder_sales = cannabis_sales2017[cannabis_sales2017['County'] == 'Boulder']
boulder_sales['Month'] = boulder_sales['Year'].astype(str) + '-' + boulder_sales['Month'].apply(lambda x: f"{x:02}")
# Calculate total sales (Medical + Retail) for Boulder
boulder_sales['Total_Sales'] = boulder_sales[['Medical', 'Retail']].sum(axis=1)
# Calculate average sale by month across Boulder County
average_sales_boulder17 = boulder_sales.groupby('Month')['Total_Sales'].mean()
average_sales_boulder17
```

```
/tmp/ipykernel_169/2291285002.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
boulder_sales['Month'] = boulder_sales['Year'].astype(str) + '-' + boulder_sales['Month'].apply(lambda x: f"{x:02}")
```

```
/tmp/ipykernel_169/2291285002.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
boulder_sales['Total_Sales'] = boulder_sales[['Medical', 'Retail']].sum(axis=1)
```

Out [49]:

	Month	Total_Sales
0	2017-01	7295509.0
1	2017-02	6893030.0
2	2017-03	8319534.0
3	2017-04	7902073.0
4	2017-05	8275350.0
5	2017-06	8727646.0
6	2017-07	8922171.0
7	2017-08	9285128.0
8	2017-09	8957775.0
9	2017-10	8613529.0
10	2017-11	8100622.0
11	2017-12	8630959.0

```
In [50]: cannabis_sales2018 = pd.read_csv('2018_sales.csv')
# Filter for Boulder County
boulder_sales = cannabis_sales2018[cannabis_sales2018['County'] == 'Boulder']
boulder_sales['Month'] = boulder_sales['Year'].astype(str) + '-' + boulder_sales['Month'].apply(lambda x: f"{x:02}")
# Calculate total sales (Medical + Retail) for Boulder
boulder_sales['Total_Sales'] = boulder_sales[['Medical', 'Retail']].sum(axis=1)
# Calculate average sale by month across Boulder County
average_sales_boulder18 = boulder_sales.groupby('Month')['Total_Sales'].mean()
average_sales_boulder18
```

```
/tmp/ipykernel_169/664146788.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
boulder_sales['Month'] = boulder_sales['Year'].astype(str) + '-' + boulder_sales['Month'].apply(lambda x: f"{x:02}")
```

```
/tmp/ipykernel_169/664146788.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
boulder_sales['Total_Sales'] = boulder_sales[['Medical', 'Retail']].sum(axis=1)
```

Out [50]:

	Month	Total_Sales
0	2018-01	7859058.0
1	2018-02	7419096.0
2	2018-03	8936520.0
3	2018-04	8559127.0
4	2018-05	8845651.0
5	2018-06	9182717.0
6	2018-07	10231421.0
7	2018-08	10189578.0
8	2018-09	9399286.0
9	2018-10	9138113.0
10	2018-11	8681963.0
11	2018-12	9835324.0

```
In [59]: cannabis_sales2019 = pd.read_csv('2019_sales.csv')
# Filter for Boulder County
boulder_sales = cannabis_sales2019[cannabis_sales2019['County'] == 'Boulder']
boulder_sales['Month'] = boulder_sales['Year'].astype(str) + '-' + boulder_sales['Month'].apply(lambda x: f"{x:02}")
# Calculate total sales (Medical + Retail) for Boulder
boulder_sales['Total_Sales'] = boulder_sales[['Medical', 'Retail']].sum(axis=1)
# Calculate average sale by month across Boulder County
average_sales_boulder19 = boulder_sales.groupby('Month')['Total_Sales'].mean()
average_sales_boulder19
```

```
/tmp/ipykernel_169/919986246.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
boulder_sales['Month'] = boulder_sales['Year'].astype(str) + '-' + boulder_sales['Month'].apply(lambda x: f"{x:02}")
```

```
/tmp/ipykernel_169/919986246.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
boulder_sales['Total_Sales'] = boulder_sales[['Medical', 'Retail']].sum(axis=1)
```

Out [59]:

	Month	Total_Sales
0	2019-01	8403300.0
1	2019-02	7876992.0
2	2019-03	9188302.0
3	2019-04	9134288.0
4	2019-05	9868130.0
5	2019-06	9916957.0
6	2019-07	11174112.0
7	2019-08	11329871.0
8	2019-09	10449778.0
9	2019-10	9839278.0
10	2019-11	9840334.0
11	2019-12	9837780.0

```
In [60]: cannabis_sales2020 = pd.read_csv('2020_sales.csv')
# Filter for Boulder County
boulder_sales = cannabis_sales2020[cannabis_sales2020['County'] == 'Boulder']
boulder_sales['Month'] = boulder_sales['Year'].astype(str) + '-' + boulder_sales['Month'].apply(lambda x: f"{x:02}")
# Calculate total sales (Medical + Retail) for Boulder
boulder_sales['Total_Sales'] = boulder_sales[['Medical', 'Retail']].sum(axis=1)
# Calculate average sale by month across Boulder County
average_sales_boulder20 = boulder_sales.groupby('Month')['Total_Sales'].mean()
average_sales_boulder20
```

```
/tmp/ipykernel_169/3341567365.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
boulder_sales['Month'] = boulder_sales['Year'].astype(str) + '-' + boulder_sales['Month'].apply(lambda x: f"{x:02}")
```

```
/tmp/ipykernel_169/3341567365.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
boulder_sales['Total_Sales'] = boulder_sales[['Medical', 'Retail']].sum(axis=1)
```

Out[60]:

	Month	Total_Sales
0	2020-01	9178307.0
1	2020-02	8996402.0
2	2020-03	10350309.0
3	2020-04	9256964.0
4	2020-05	11276076.0
5	2020-06	12066049.0
6	2020-07	13903140.0
7	2020-08	13904619.0
8	2020-09	13109154.0
9	2020-10	13005156.0
10	2020-11	11563955.0
11	2020-12	11722886.0

In [58]: `import pandas as pd`

```
den_passengers_df = pd.read_csv('den_passengers.csv')
# Convert 'Month' to datetime to facilitate filtering and grouping
den_passengers_df['Month'] = pd.to_datetime(den_passengers_df['Month'])
# Filter the dataset for years 2016-2020
den_passengers = den_passengers_df[(den_passengers_df['Month'].dt.year >= 2016) && (den_passengers_df['Month'].dt.year <= 2020)]
# Sum all passenger numbers except for the 'Month' column to get total passengers
den_passengers['Total_Passengers'] = den_passengers.drop('Month', axis=1).sum(axis=1)
# Group by month (YYYY-MM) and sum total passengers for each month
monthly_totals = den_passengers.groupby(den_passengers['Month'].dt.to_period('M')).sum()
# Convert 'Month' back to string for display purposes
monthly_totals['Month'] = monthly_totals['Month'].astype(str)
# Displaying the results
monthly_totals[['Month', 'Total_Passengers']]
```

```
/tmp/ipykernel_169/1312505277.py:13: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
den_passengers['Total_Passengers'] = den_passengers.drop('Month', axis=  
1).sum(axis=1, numeric_only=True)
```

```
/tmp/ipykernel_169/1312505277.py:16: FutureWarning: The default value of nu  
meric_only in DataFrameGroupBy.sum is deprecated. In a future version, nume  
ric_only will default to False. Either specify numeric_only or select only  
columns which should be valid for the function.
```

```
monthly_totals = den_passengers.groupby(den_passengers['Month'].dt.to_per  
iod("M")).sum().reset_index()
```

Out [58]:

	Month	Total_Passengers
0	2016-01	4289432.0
1	2016-02	4069006.0
2	2016-03	4689761.0
3	2016-04	4377597.0
4	2016-05	4901890.0
5	2016-06	5208947.0
6	2016-07	5474503.0
7	2016-08	5395910.0
8	2016-09	5112271.0
9	2016-10	5226077.0
10	2016-11	4695897.0
11	2016-12	4825224.0
12	2017-01	4434862.0
13	2017-02	4204744.0
14	2017-03	5193855.0
15	2017-04	4812910.0
16	2017-05	5271223.0
17	2017-06	5650682.0
18	2017-07	5839661.0
19	2017-08	5727326.0
20	2017-09	5097144.0
21	2017-10	5282947.0
22	2017-11	4913238.0
23	2017-12	4950804.0
24	2018-01	4685586.0
25	2018-02	4345062.0
26	2018-03	5278712.0
27	2018-04	5096401.0
28	2018-05	5511995.0
29	2018-06	5905029.0
30	2018-07	6119150.0
31	2018-08	5973496.0
32	2018-09	5375947.0
33	2018-10	5704639.0



	Month	Total_Passengers
34	2018-11	5274999.0
35	2018-12	5223597.0
36	2019-01	4914602.0
37	2019-02	4588397.0
38	2019-03	5451215.0
39	2019-04	5321437.0
40	2019-05	6041706.0
41	2019-06	6385666.0
42	2019-07	6732686.0
43	2019-08	6455071.0
44	2019-09	5958431.0
45	2019-10	6073368.0
46	2019-11	5304165.0
47	2019-12	5788959.0
48	2020-01	5198334.0
49	2020-02	4908375.0
50	2020-03	2923357.0
51	2020-04	299098.0
52	2020-05	834871.0
53	2020-06	1750368.0
54	2020-07	2592277.0
55	2020-08	2985681.0
56	2020-09	2901918.0
57	2020-10	3314971.0
58	2020-11	2868692.0
59	2020-12	3163187.0

```
In [55]: merged_data = pd.merge(monthly_totals, average_sales_boulder16, on='Month',  
# Calculate correlation  
correlation = merged_data[['Total_Passengers', 'Total_Sales']].corr()  
correlation
```

Out [55]:

	Total_Passengers	Total_Sales
Total_Passengers	1.000000	0.854351
Total_Sales	0.854351	1.000000

```
In [56]: merged_data = pd.merge(monthly_totals, average_sales_boulder17, on='Month',
# Calculate correlation
correlation = merged_data[['Total_Passengers', 'Total_Sales']].corr()

correlation
```

Out [56]:

	Total_Passengers	Total_Sales
Total_Passengers	1.00000	0.89221
Total_Sales	0.89221	1.00000

```
In [57]: merged_data = pd.merge(monthly_totals, average_sales_boulder18, on='Month',
# Calculate correlation
correlation = merged_data[['Total_Passengers', 'Total_Sales']].corr()

correlation
```

Out [57]:

	Total_Passengers	Total_Sales
Total_Passengers	1.000000	0.862776
Total_Sales	0.862776	1.000000

```
In [61]: merged_data = pd.merge(monthly_totals, average_sales_boulder19, on='Month',
# Calculate correlation
correlation = merged_data[['Total_Passengers', 'Total_Sales']].corr()

correlation
```

Out [61]:

	Total_Passengers	Total_Sales
Total_Passengers	1.000000	0.911291
Total_Sales	0.911291	1.000000

```
In [62]: merged_data = pd.merge(monthly_totals, average_sales_boulder20, on='Month',
# Calculate correlation
correlation = merged_data[['Total_Passengers', 'Total_Sales']].corr()

correlation
```

Out [62]:

	Total_Passengers	Total_Sales
Total_Passengers	1.000000	-0.165205
Total_Sales	-0.165205	1.000000

In [63]:

```
cannabis_sales2020 = pd.read_csv('2020_sales.csv')
# Filter for Boulder County
denver_sales = cannabis_sales2020[cannabis_sales2020['County'] == 'Denver']
# Combine 'Year' and 'Month' into a unified 'Month' column with proper format
denver_sales['Month'] = denver_sales['Year'].astype(str) + '-' + denver_sales['Month']
# Calculate total sales (Medical + Retail) for Boulder
denver_sales['Total_Sales'] = denver_sales[['Medical', 'Retail']].sum(axis=1)
# Calculate average sale by month across Boulder County
average_sales_denver20 = denver_sales.groupby('Month')['Total_Sales'].mean()
average_sales_denver20
```

/tmp/ipykernel\_169/2784392800.py:5: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)  
denver\_sales['Month'] = denver\_sales['Year'].astype(str) + '-' + denver\_sales['Month'].apply(lambda x: f"{x:02}")

/tmp/ipykernel\_169/2784392800.py:7: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)  
denver\_sales['Total\_Sales'] = denver\_sales[['Medical', 'Retail']].sum(axis=1)

Out [63]:

	Month	Total_Sales
0	2020-01	44608998.0
1	2020-02	44256581.0
2	2020-03	50897219.0
3	2020-04	47112522.0
4	2020-05	63191565.0
5	2020-06	66098923.0
6	2020-07	75708966.0
7	2020-08	71562428.0
8	2020-09	65894751.0
9	2020-10	65205136.0
10	2020-11	56193940.0
11	2020-12	58746845.0

```
In [64]: merged_data = pd.merge(monthly_totals, average_sales_denver20, on='Month', h

# Calculate correlation
correlation = merged_data[['Total_Passengers', 'Total_Sales']].corr()

correlation
```

```
Out[64]:
```

	Total_Passengers	Total_Sales
Total_Passengers	1.000000	-0.305098
Total_Sales	-0.305098	1.000000

```
In [65]: cannabis_sales2016 = pd.read_csv('2016_sales.csv')
# Filter for Boulder County
denver_sales = cannabis_sales2016[cannabis_sales2016['County'] == 'Denver']
# Combine 'Year' and 'Month' into a unified 'Month' column with proper format
denver_sales['Month'] = denver_sales['Year'].astype(str) + '-' + denver_sales['Month']
# Calculate total sales (Medical + Retail) for Boulder
denver_sales['Total_Sales'] = denver_sales[['Medical', 'Retail']].sum(axis=1)
# Calculate average sale by month across Boulder County
average_sales_denver16 = denver_sales.groupby('Month')['Total_Sales'].mean()
average_sales_denver16
```

```
/tmp/ipykernel_169/718621435.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
denver_sales['Month'] = denver_sales['Year'].astype(str) + '-' + denver_sales['Month'].apply(lambda x: f"{x:02}")
```

```
/tmp/ipykernel_169/718621435.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
denver_sales['Total_Sales'] = denver_sales[['Medical', 'Retail']].sum(axis=1)
```

Out [65]:

	Month	Total_Sales
0	2016-01	33662180.0
1	2016-02	34431926.0
2	2016-03	38812830.0
3	2016-04	40297125.0
4	2016-05	39448519.0
5	2016-06	40474412.0
6	2016-07	45455882.0
7	2016-08	47846275.0
8	2016-09	48613660.0
9	2016-10	45774674.0
10	2016-11	43329183.0
11	2016-12	43464998.0

```
In [66]: merged_data = pd.merge(monthly_totals, average_sales_denver16, on='Month', h
# Calculate correlation
correlation = merged_data[['Total_Passengers', 'Total_Sales']].corr()

correlation
```

Out [66]:

	Total_Passengers	Total_Sales
Total_Passengers	1.000000	0.816161
Total_Sales	0.816161	1.000000

In [ ]: