1011_heatmap

February 7, 2023

1 Heat Map

- 1011
- github: https://github.com/Maruf2309
- Linkedin: https://www.linkedin.com/in/maruf-hossain-682213150/
- Kaggle : https://www.kaggle.com/marufnthewindows

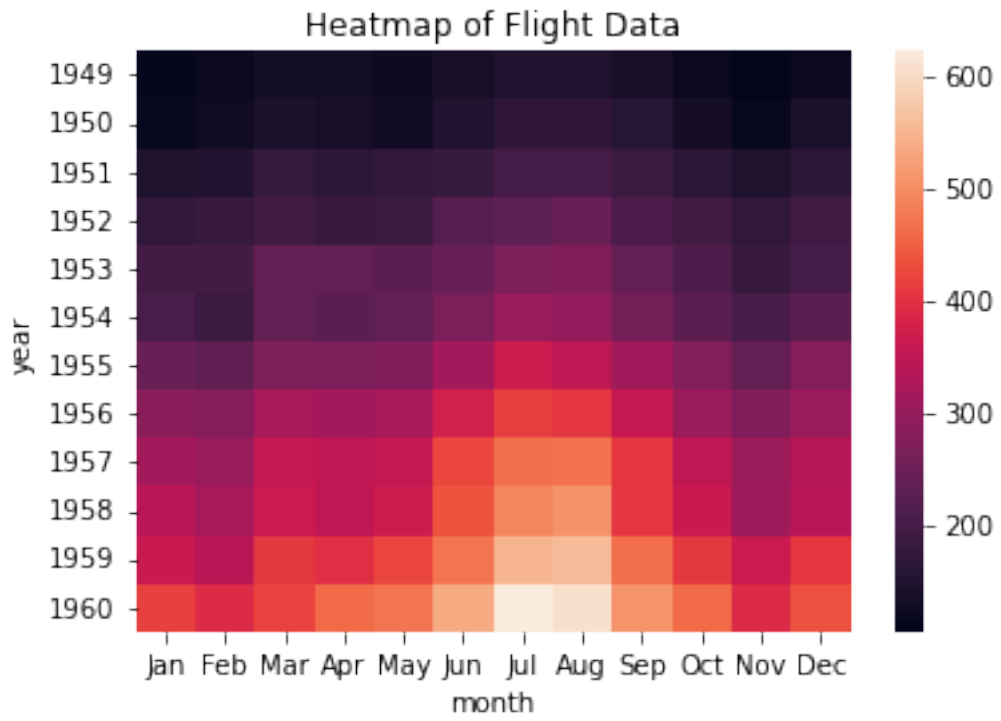
[15]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import sklearn
```

[2]:
```
flight_data = sns.load_dataset('flights') # Built in Dataset loaded
flight_data.head()
```

[2]:

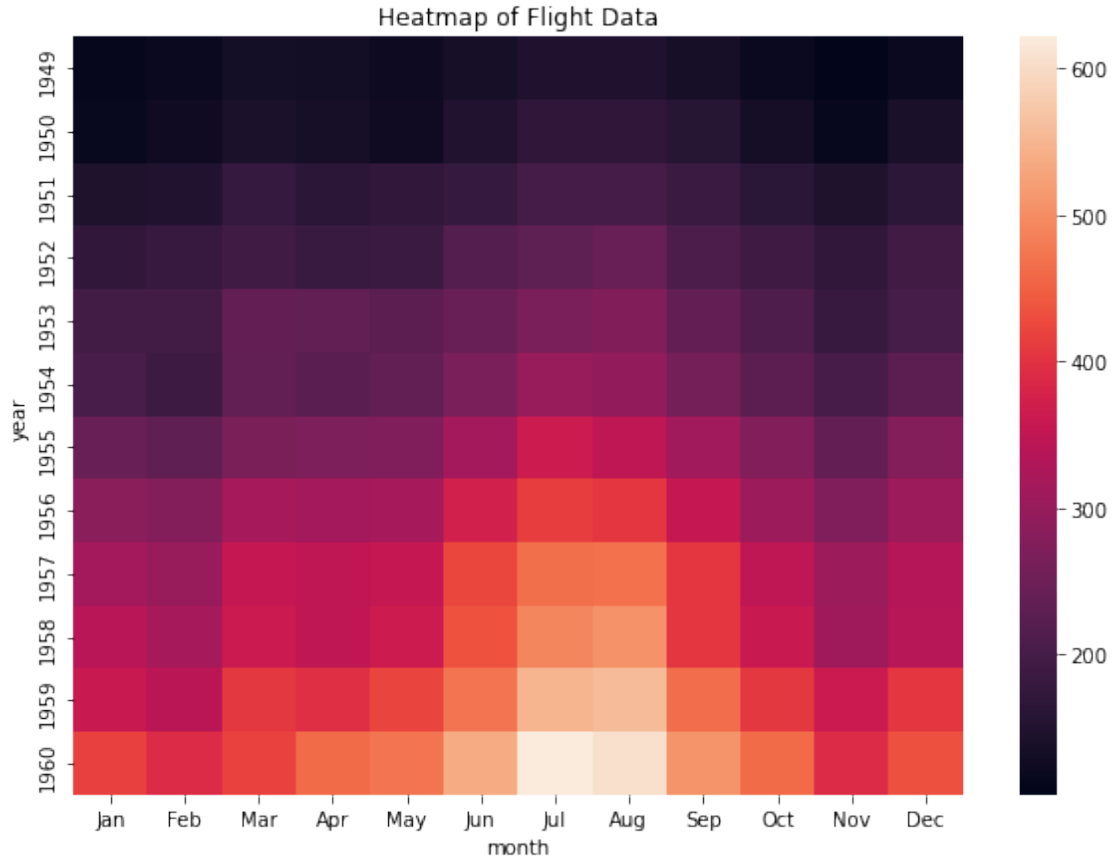
	year	month	passengers
0	1949	Jan	112
1	1949	Feb	118
2	1949	Mar	132
3	1949	Apr	129
4	1949	May	121

[3]:
```
flights = flight_data.pivot("year", "month", "passengers")  # Left: Year, x =␣
 ↪Month, right: Passengers
ax = sns.heatmap(flights)
plt.title("Heatmap of Flight Data")
plt.show()
```

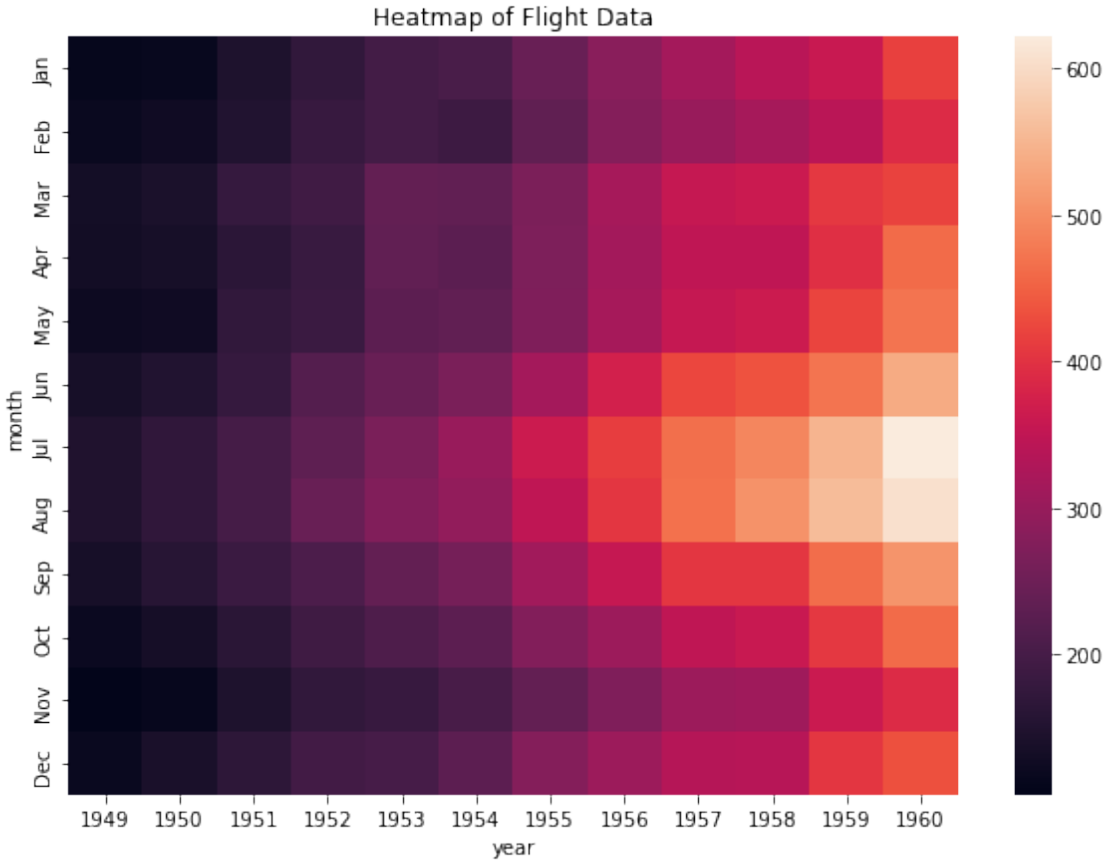


- Lighter shade indicate higher value & lower shade indicate lower value

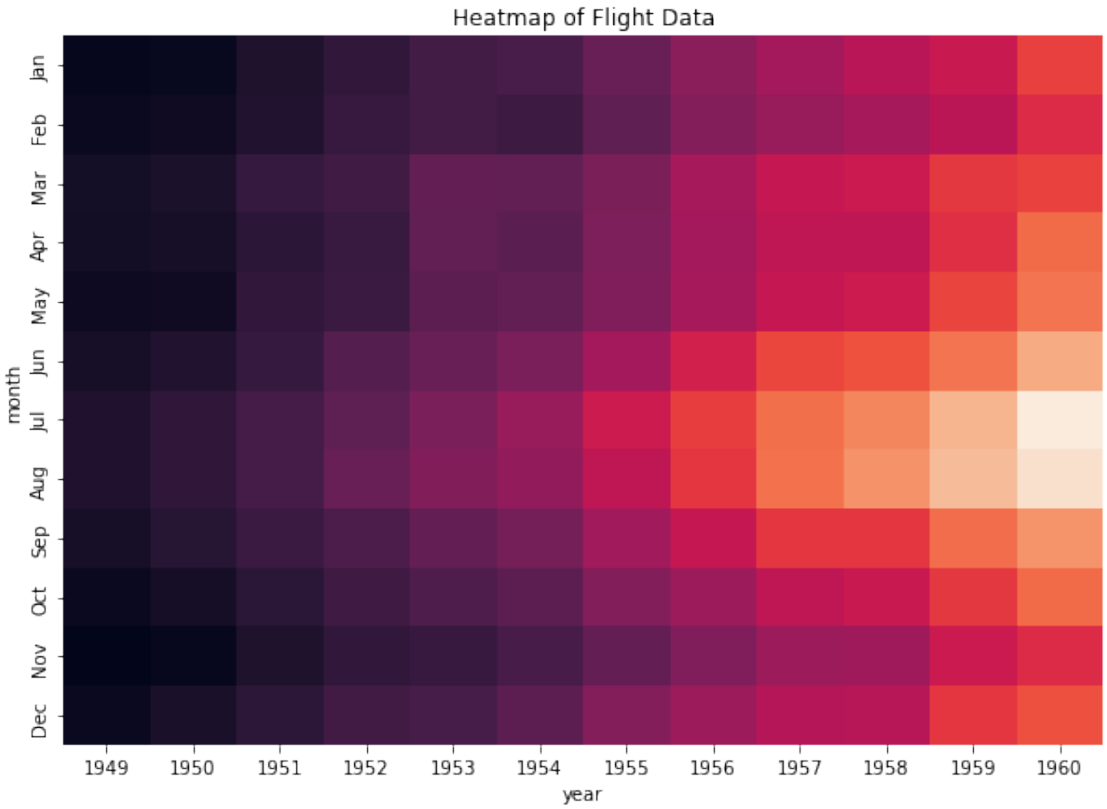
[4]:
```
# Size Customization
flights = flight_data.pivot("year", "month", "passengers")  # Left: Year, x =␣
 ↪Month, right: Passengers
plt.figure(figsize=(10,7))
ax = sns.heatmap(flights)
plt.title("Heatmap of Flight Data")
plt.show()
```



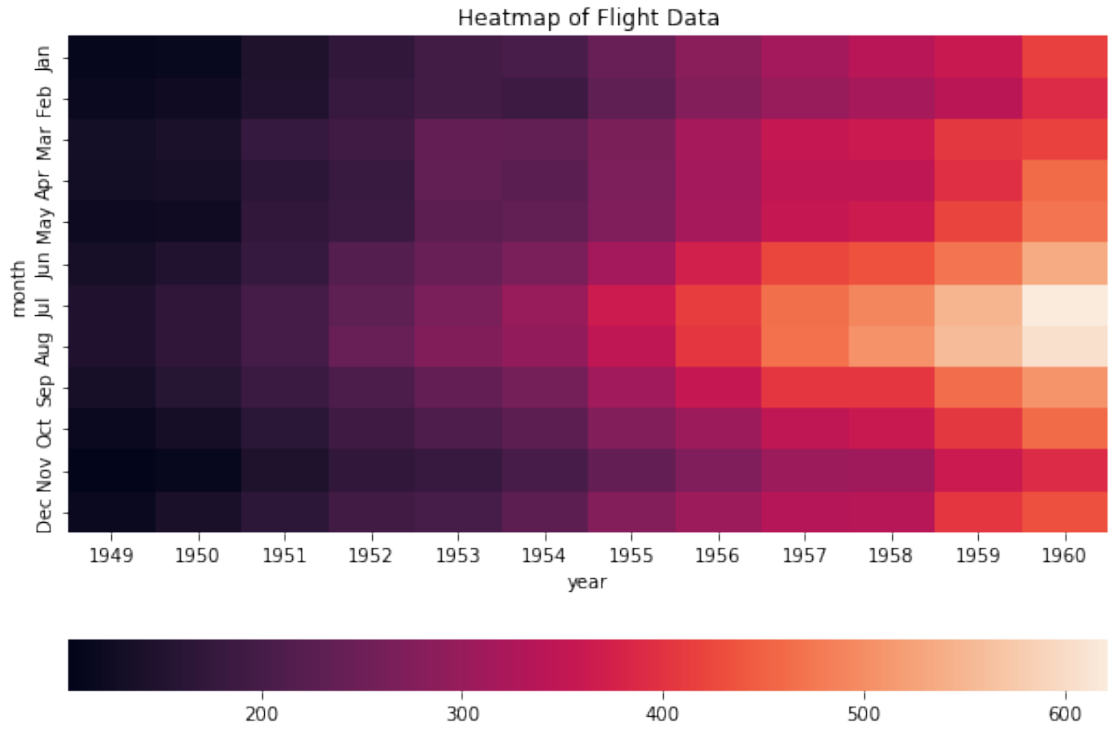
[5]:
```
# Altering Position
flights = flight_data.pivot( "month", "year", "passengers")  # Altering Position
plt.figure(figsize=(10,7))
ax = sns.heatmap(flights)
plt.title("Heatmap of Flight Data")
plt.show()
```



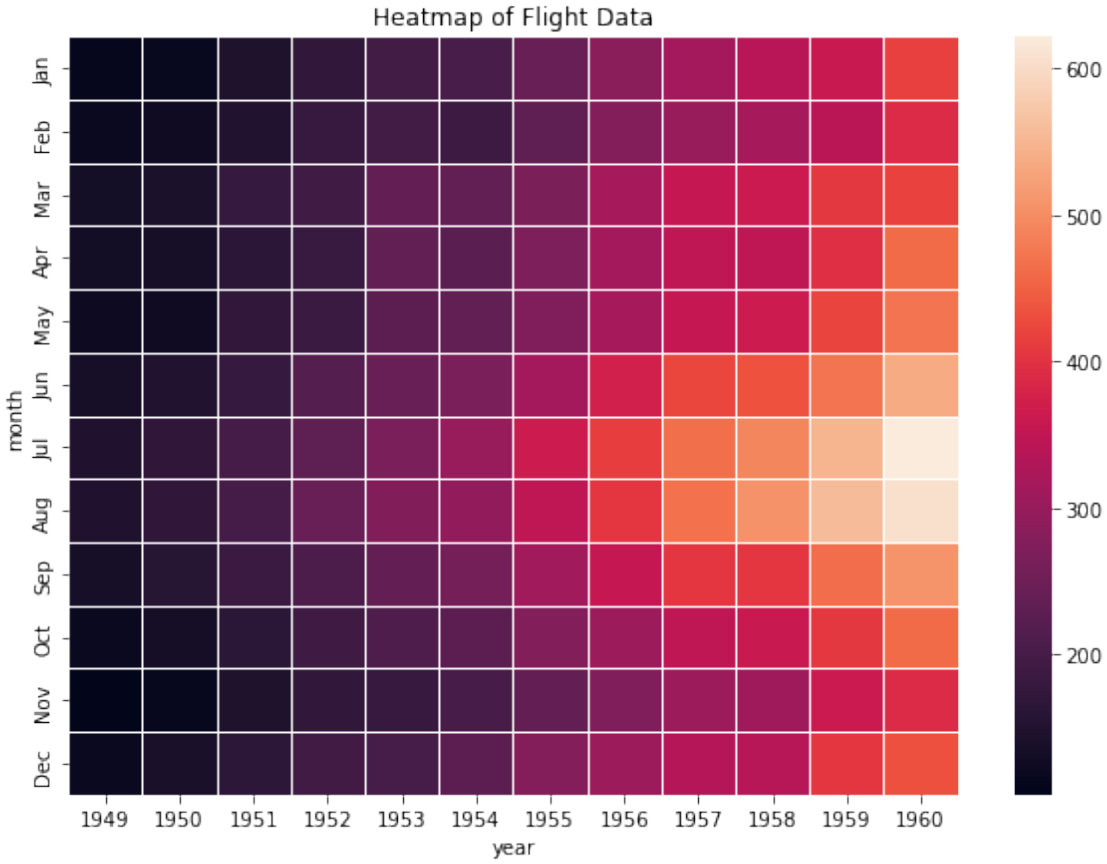
[6]:
```
flights = flight_data.pivot( "month", "year", "passengers")
plt.figure(figsize=(10,7))
ax = sns.heatmap(flights, cbar=False)          # Remove Color Bar
plt.title("Heatmap of Flight Data")
plt.show()
```



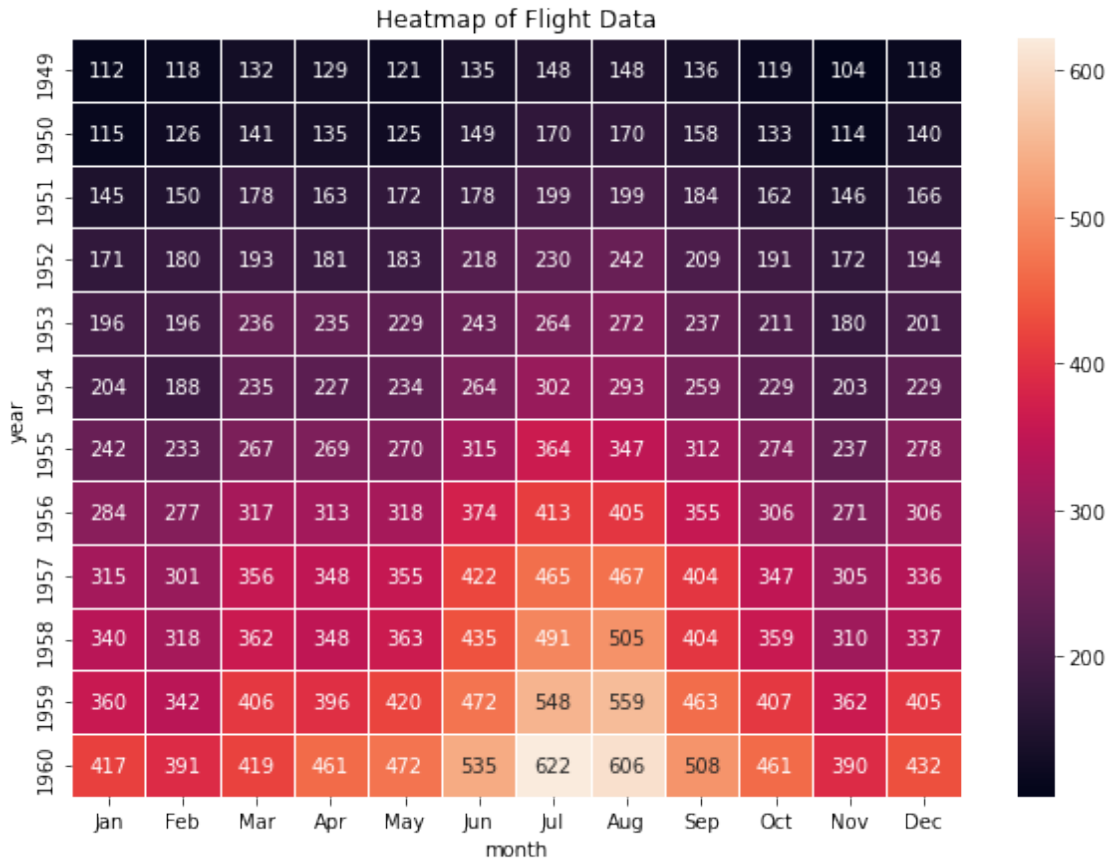
[7]:
```
flights = flight_data.pivot( "month", "year", "passengers")
plt.figure(figsize=(10,7))
ax = sns.heatmap(flights, cbar_kws={"orientation":"horizontal"})     # Change␣
 ↪color bar position
plt.title("Heatmap of Flight Data")
plt.show()
```



[8]:
```
flights = flight_data.pivot( "month", "year", "passengers")
plt.figure(figsize=(10,7))
ax = sns.heatmap(flights, linewidths=0.5)     # Linewidth
plt.title("Heatmap of Flight Data")
plt.show()
```



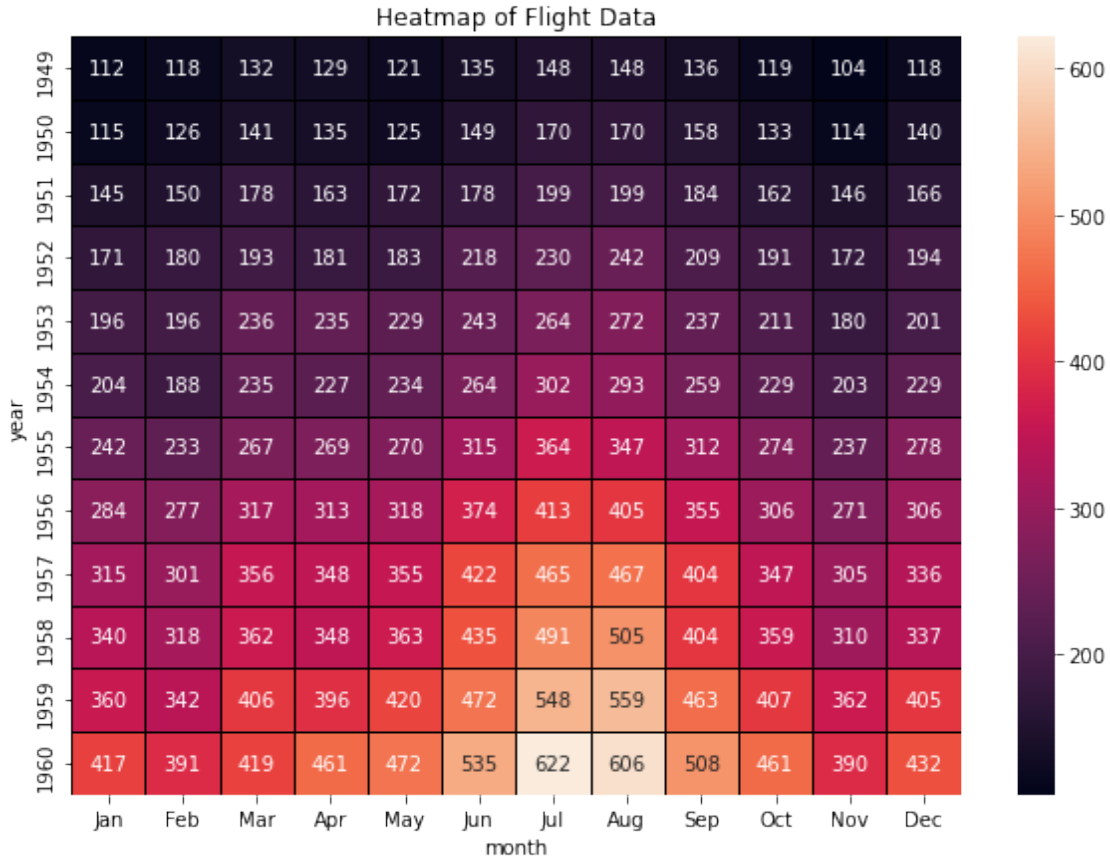
[9]:
```
flights = flight_data.pivot( "year", "month", "passengers")
plt.figure(figsize=(10,7))
ax = sns.heatmap(flights, linewidths=0.5, annot=True, fmt="d")     # format
plt.title("Heatmap of Flight Data")
plt.show()
```



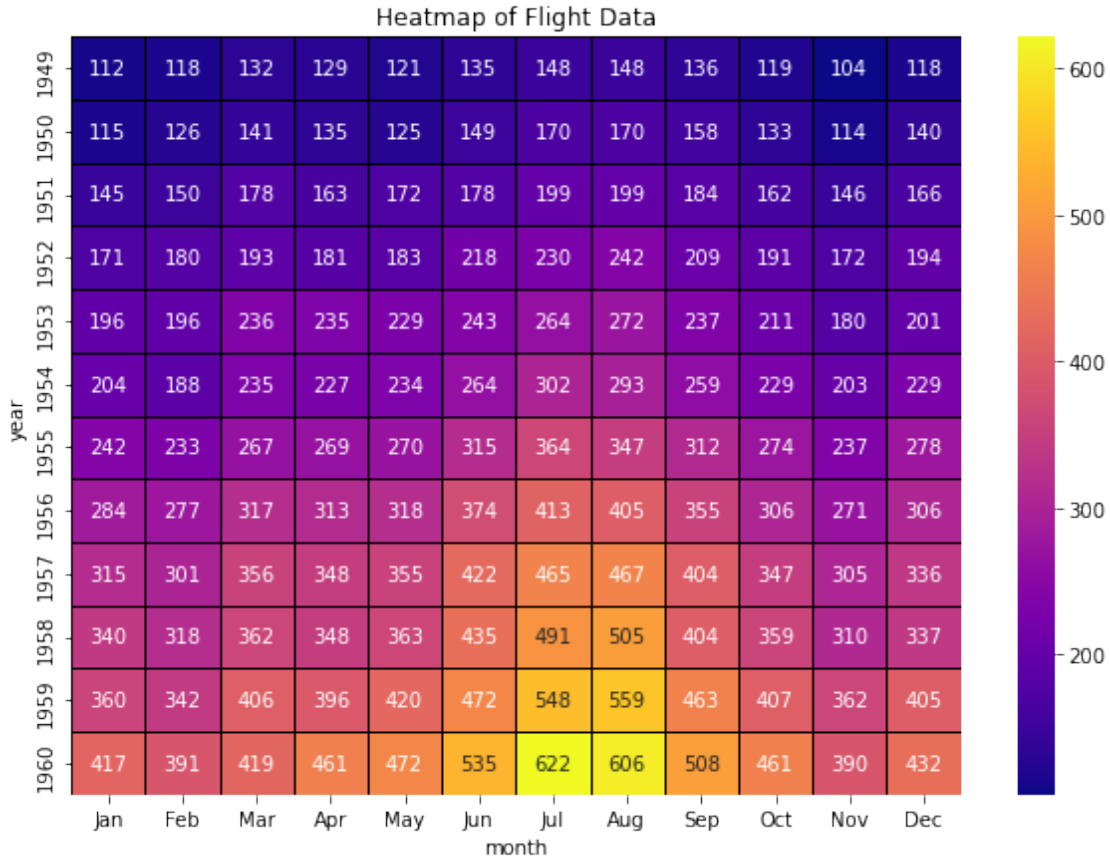
[10]:
```
flight_data.describe()
```

[10]:
```
              year  passengers
count   144.000000  144.000000
mean   1954.500000  280.298611
std       3.464102  119.966317
min    1949.000000  104.000000
25%    1951.750000  180.000000
50%    1954.500000  265.500000
75%    1957.250000  360.500000
max    1960.000000  622.000000
```

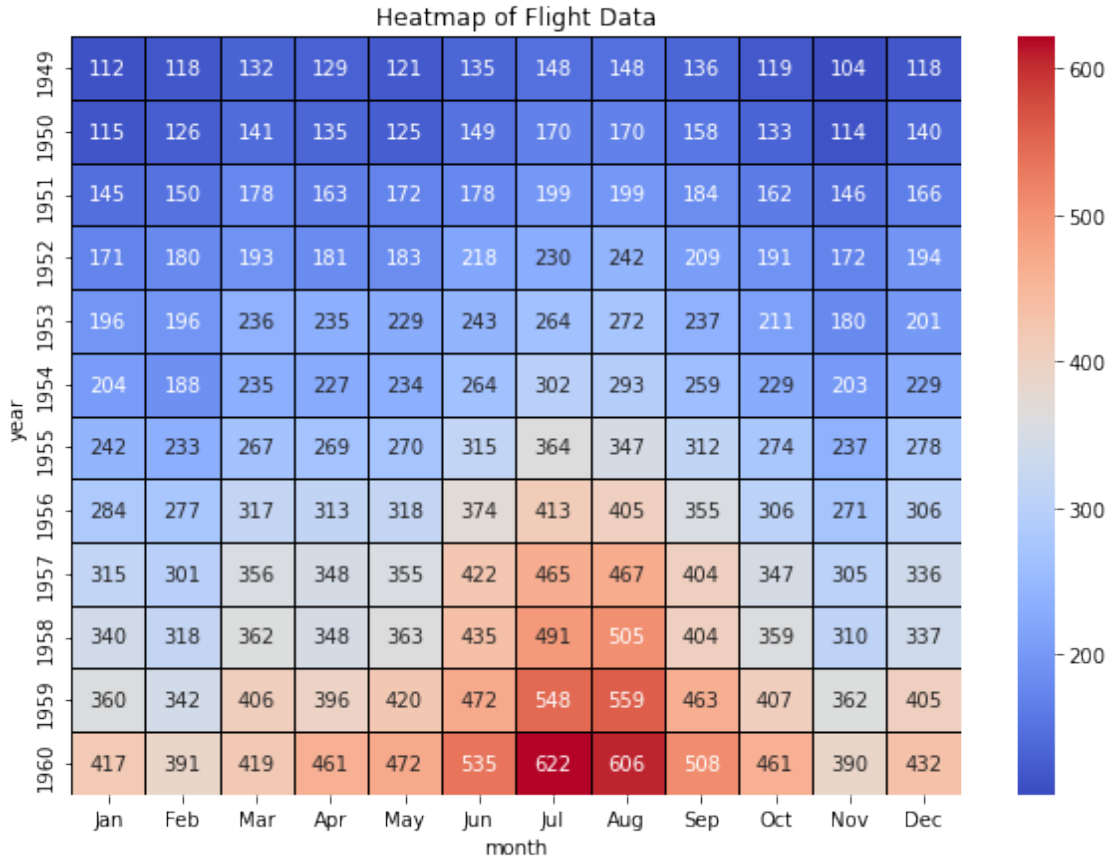
[11]:
```
flights = flight_data.pivot( "year", "month", "passengers")
plt.figure(figsize=(10,7))
ax = sns.heatmap(flights, linewidths=0.5, linecolor='black', annot=True,␣
 ↪fmt="d")     # format
plt.title("Heatmap of Flight Data")
plt.show()
```



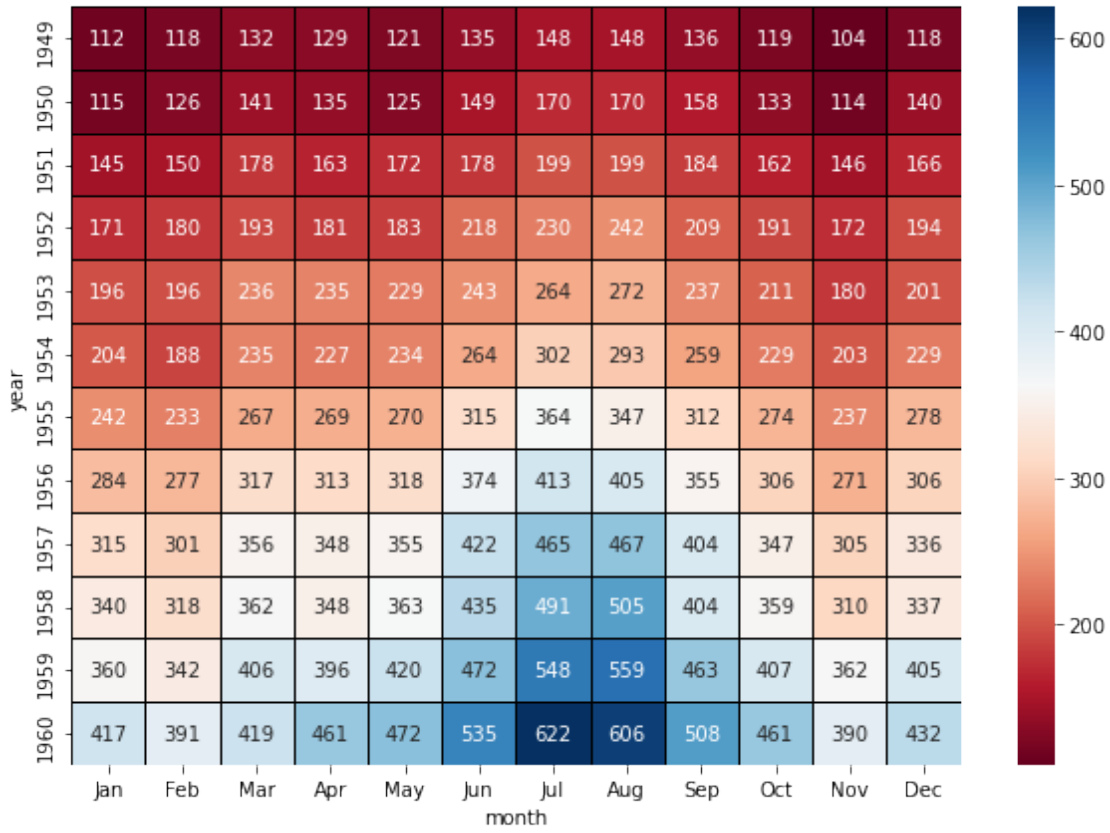
[12]:
```
flights = flight_data.pivot( "year", "month", "passengers")
plt.figure(figsize=(10,7))
ax = sns.heatmap(flights, linewidths=0.5, linecolor='black', annot=True,␣
 ↪fmt="d", cmap='plasma')     # Change heat map color by color map/ cmap
plt.title("Heatmap of Flight Data")
plt.show()
```



[13]:
```
flights = flight_data.pivot( "year", "month", "passengers")
plt.figure(figsize=(10,7))
ax = sns.heatmap(flights, linewidths=0.5, linecolor='black', annot=True,␣
 ↪fmt="d", cmap='coolwarm')     # Change heat map color by color map/ cmap
plt.title("Heatmap of Flight Data")
plt.show()
```



[14]:
```
flight_data = flight_data.pivot('year', 'month', 'passengers')
plt.figure(figsize=(10,7))
ax = sns.heatmap(flight_data, linewidths=0.5, linecolor='black', annot=True,␣
↪fmt="d", cmap='RdBu')    # more map >> cmap='magma'
```



2 Correlation Matrix

[17]:
```
from sklearn.datasets import fetch_california_housing
housing_data = fetch_california_housing()
```

[18]:
```
df = pd.DataFrame(housing_data.data)
df.columns = housing_data.feature_names
df['price'] = housing_data.target
df.head()
```

[18]:

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	\
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	

	Longitude	price
0	-122.23	4.526
1	-122.22	3.585

```
2    -122.24  3.521
3    -122.25  3.413
4    -122.25  3.422
```

[21]:
```
print(housing_data.DESCR) # Know About the Data or Data description
```

```
.. _california_housing_dataset:

California Housing dataset
--------------------------

**Data Set Characteristics:**

    :Number of Instances: 20640

    :Number of Attributes: 8 numeric, predictive attributes and the target

    :Attribute Information:
        - MedInc        median income in block group
        - HouseAge      median house age in block group
        - AveRooms      average number of rooms per household
        - AveBedrms     average number of bedrooms per household
        - Population    block group population
        - AveOccup      average number of household members
        - Latitude      block group latitude
        - Longitude     block group longitude

    :Missing Attribute Values: None

This dataset was obtained from the StatLib repository.
https://www.dcc.fc.up.pt/~ltorgo/Regression/cal_housing.html

The target variable is the median house value for California districts,
expressed in hundreds of thousands of dollars ($100,000).

This dataset was derived from the 1990 U.S. census, using one row per census
block group. A block group is the smallest geographical unit for which the U.S.
Census Bureau publishes sample data (a block group typically has a population
of 600 to 3,000 people).

An household is a group of people residing within a home. Since the average
number of rooms and bedrooms in this dataset are provided per household, these
columns may take surpinsingly large values for block groups with few households
and many empty houses, such as vacation resorts.

It can be downloaded/loaded using the
:func:`sklearn.datasets.fetch_california_housing` function.
```

```
.. topic:: References

    - Pace, R. Kelley and Ronald Barry, Sparse Spatial Autoregressions,
      Statistics and Probability Letters, 33 (1997) 291-297
```

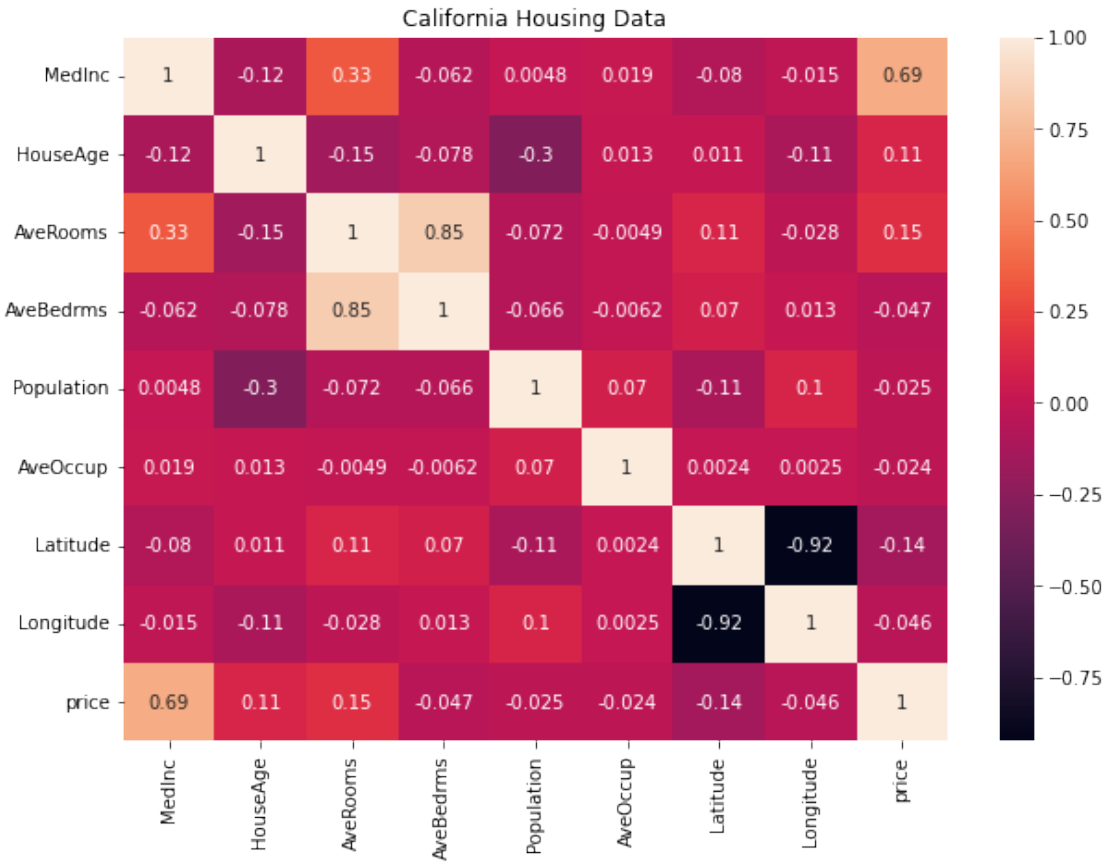
[24]:
```
df.corr()  # Data Correlation
```

[24]:
```
              MedInc  HouseAge  AveRooms  AveBedrms  Population  AveOccup  \
MedInc      1.000000 -0.119034  0.326895  -0.062040    0.004834  0.018766
HouseAge   -0.119034  1.000000 -0.153277  -0.077747   -0.296244  0.013191
AveRooms    0.326895 -0.153277  1.000000   0.847621   -0.072213 -0.004852
AveBedrms  -0.062040 -0.077747  0.847621   1.000000   -0.066197 -0.006181
Population  0.004834 -0.296244 -0.072213  -0.066197    1.000000  0.069863
AveOccup    0.018766  0.013191 -0.004852  -0.006181    0.069863  1.000000
Latitude   -0.079809  0.011173  0.106389   0.069721   -0.108785  0.002366
Longitude  -0.015176 -0.108197 -0.027540   0.013344    0.099773  0.002476
price       0.688075  0.105623  0.151948  -0.046701   -0.024650 -0.023737

              Latitude  Longitude     price
MedInc     -0.079809  -0.015176  0.688075
HouseAge    0.011173  -0.108197  0.105623
AveRooms    0.106389  -0.027540  0.151948
AveBedrms   0.069721   0.013344 -0.046701
Population -0.108785   0.099773 -0.024650
AveOccup    0.002366   0.002476 -0.023737
Latitude    1.000000  -0.924664 -0.144160
Longitude  -0.924664   1.000000 -0.045967
price      -0.144160  -0.045967  1.000000
```

[28]:
```
plt.figure(figsize=(10,7))
ax = sns.heatmap(df.corr(), annot=True)   # 0.85 is most correlated values;␣
 ↪whereas 0-92 is leaset corraled values
plt.title('California Housing Data')
```

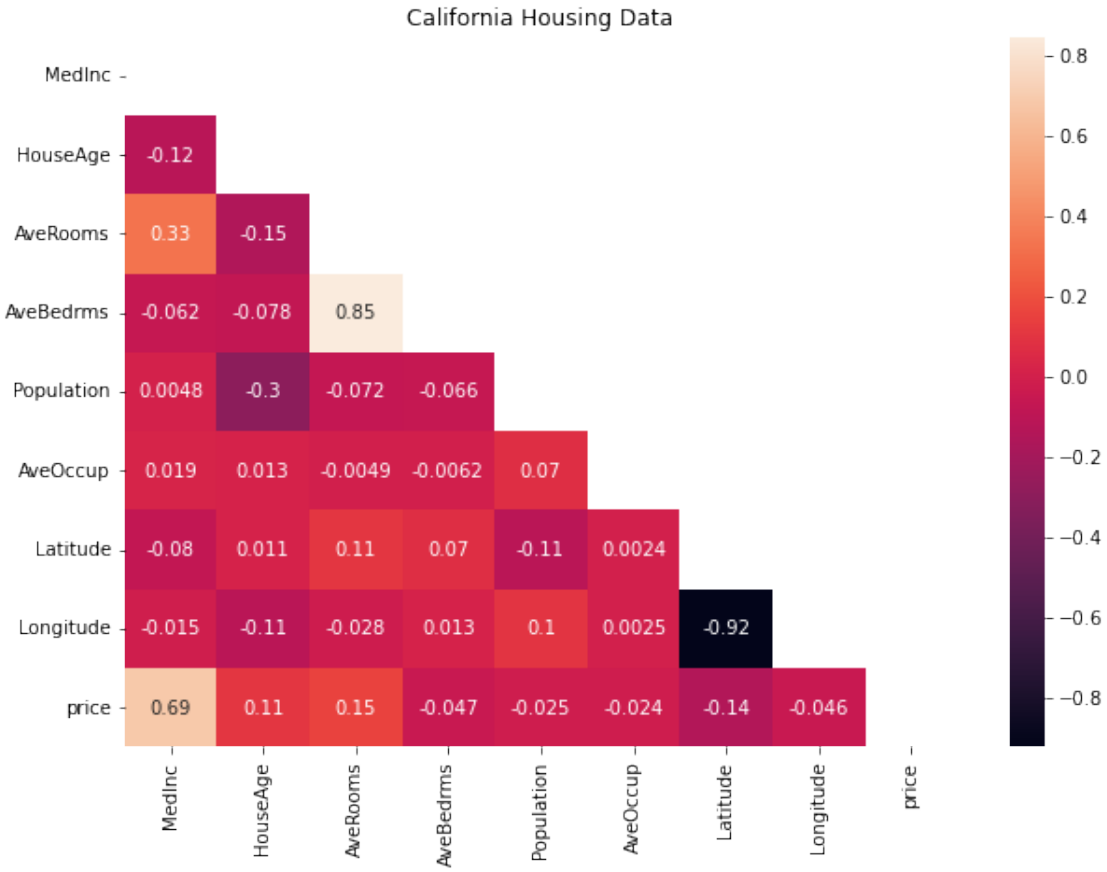
[28]:
```
Text(0.5, 1.0, 'California Housing Data')
```



2.0.1 Lower half Corrlation

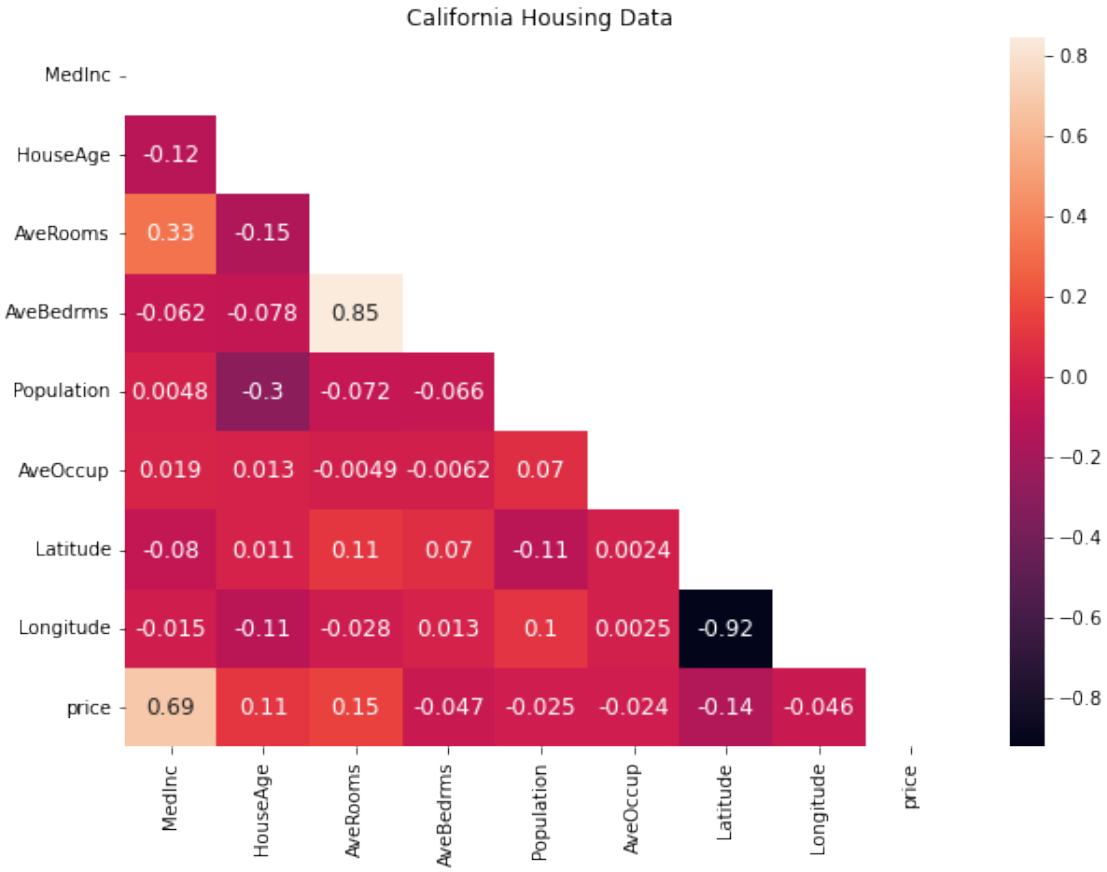
[29]:
```
plt.figure(figsize=(10,7))
mask = np.triu(np.ones_like(df.corr()))  # lower half by mask
ax = sns.heatmap(df.corr(), annot=True, mask= mask)
plt.title('California Housing Data')
```

[29]: Text(0.5, 1.0, 'California Housing Data')



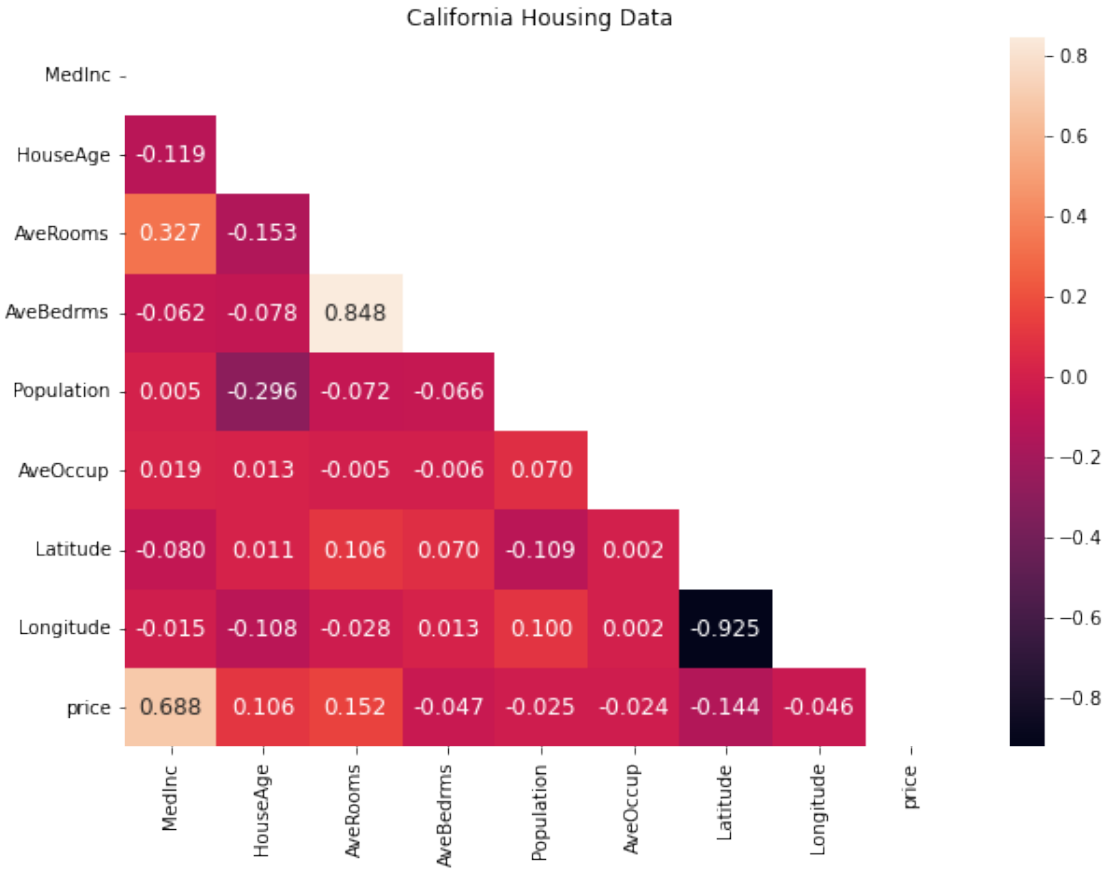
[31]:
```
plt.figure(figsize=(10,7))
mask = np.triu(np.ones_like(df.corr()))  # lower half by mask
ax = sns.heatmap(df.corr(), annot=True, mask= mask, annot_kws={'fontsize':12}) ␣
 ↪# Incrase font size
plt.title('California Housing Data')
```

[31]:
```
Text(0.5, 1.0, 'California Housing Data')
```



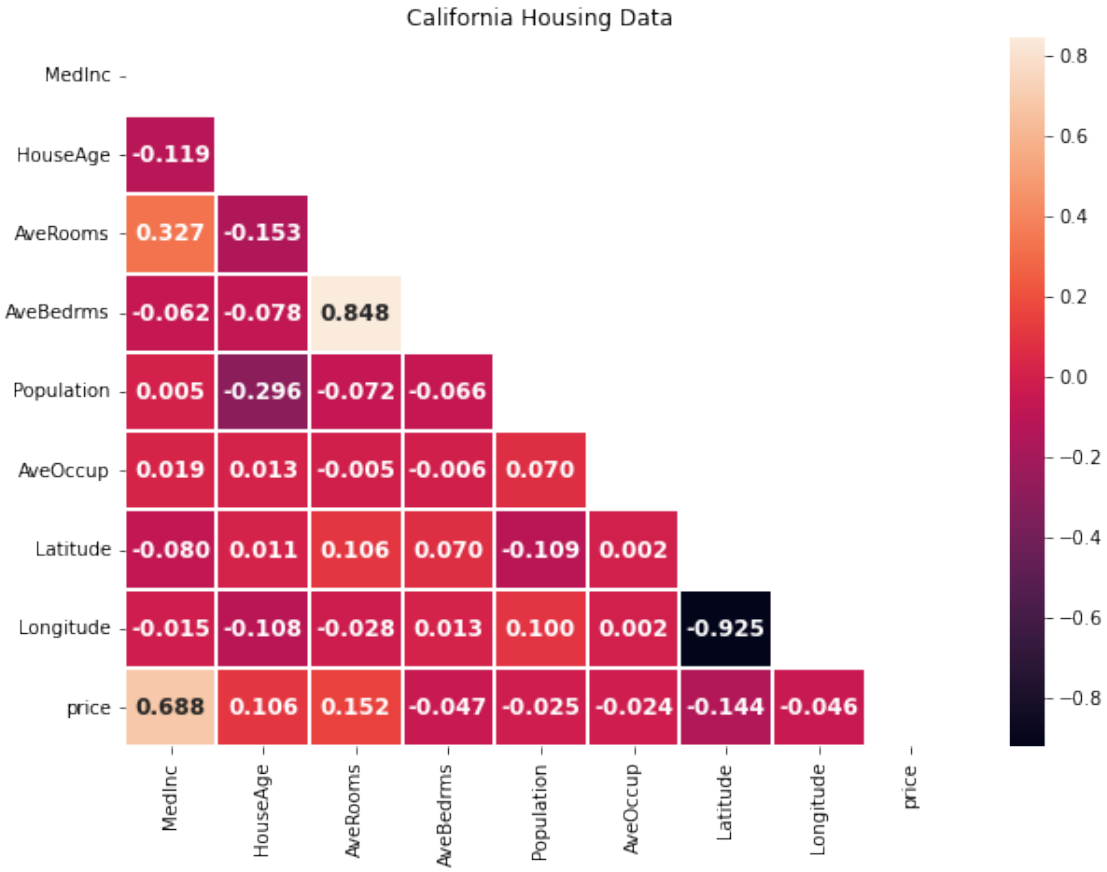
[33]:
```
plt.figure(figsize=(10,7))
mask = np.triu(np.ones_like(df.corr()))  # lower half by mask
ax = sns.heatmap(df.corr(), annot=True, mask= mask, annot_kws={'fontsize':12},␣
↪fmt="0.3f")  # all value at 3 decimal point
plt.title('California Housing Data')
```

[33]:
```
Text(0.5, 1.0, 'California Housing Data')
```



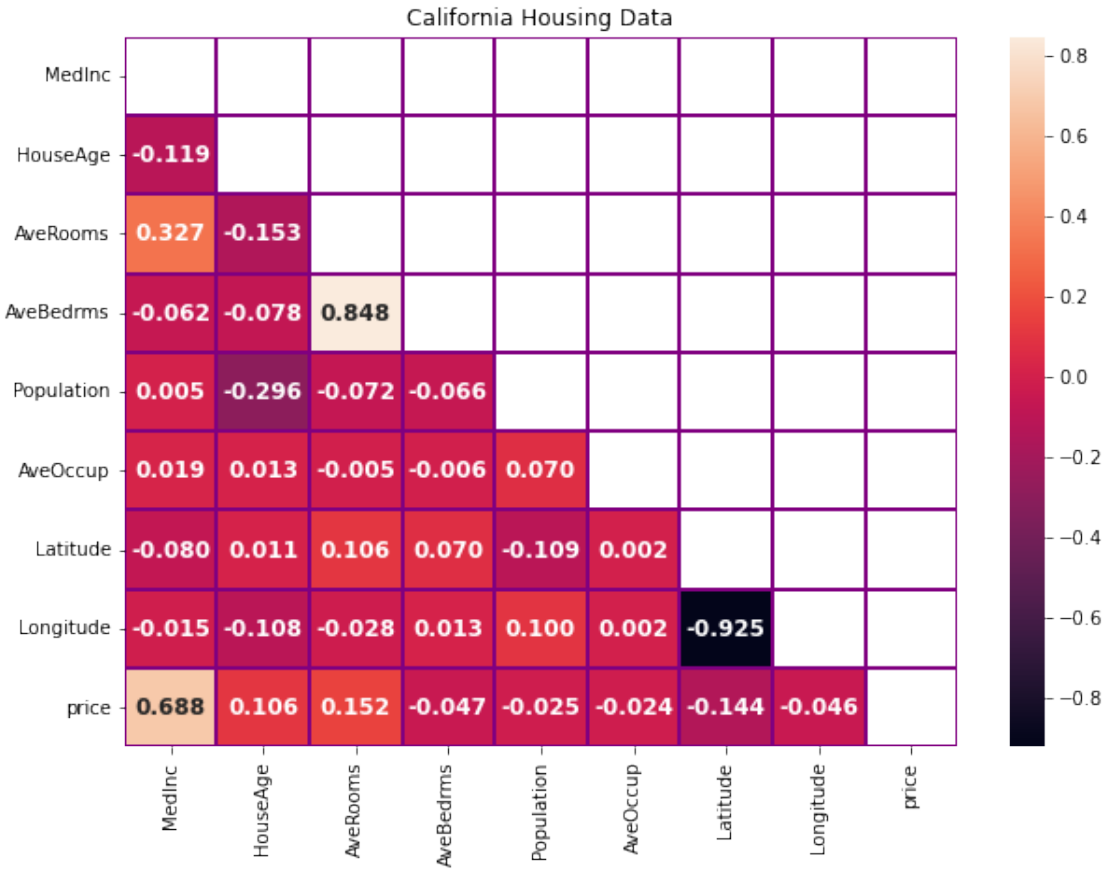
[36]:
```
plt.figure(figsize=(10,7))
mask = np.triu(np.ones_like(df.corr()))  # lower half by mask
ax = sns.heatmap(df.corr(), annot=True, mask= mask, annot_kws={'fontsize':12,␣
 ↪'fontweight':'bold'}, linewidth=0.3, fmt="0.3f")  # bold
plt.title('California Housing Data')
```

[36]:
```
Text(0.5, 1.0, 'California Housing Data')
```



[38]:
```
plt.figure(figsize=(10,7))
mask = np.triu(np.ones_like(df.corr()))  # lower half by mask
ax = sns.heatmap(df.corr(), annot=True, mask= mask, annot_kws={'fontsize':12,␣
↪'fontweight':'bold'}, linewidth=0.3,linecolor='purple', fmt="0.3f")
plt.title('California Housing Data')
```

[38]:
```
Text(0.5, 1.0, 'California Housing Data')
```


California Housing Data
MedInc
HouseAge
AveRooms
AveBedrms
Population
AveOccup
Latitude
Longitude
price
-0.119
0.327 -0.153
-0.062 -0.078 0.848
0.005 -0.296 -0.072 -0.066
0.019 0.013 -0.005 -0.006 0.070
-0.080 0.011 0.106 0.070 -0.109 0.002
-0.015 -0.108 -0.028 0.013 0.100 0.002 -0.925
0.688 0.106 0.152 -0.047 -0.025 -0.024 -0.144 -0.046
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8