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

df=pd.read_csv('../data/raw/California_Houses.csv')

df
```

```
       Median_House_Value  Median_Income  Median_Age  Tot_Rooms
Tot_Bedrooms  \
0                452600.0         8.3252          41        880
129
1                358500.0         8.3014          21       7099
1106
2                352100.0         7.2574          52       1467
190
3                341300.0         5.6431          52       1274
235
4                342200.0         3.8462          52       1627
280
...                   ...            ...         ...        ...
...
20635             78100.0         1.5603          25       1665
374
20636             77100.0         2.5568          18        697
150
20637             92300.0         1.7000          17       2254
485
20638             84700.0         1.8672          18       1860
409
20639             89400.0         2.3886          16       2785
616

       Population  Households  Latitude  Longitude  Distance_to_coast
\
0             322         126     37.88    -122.23        9263.040773

1            2401        1138     37.86    -122.22       10225.733072

2             496         177     37.85    -122.24        8259.085109

3             558         219     37.85    -122.25        7768.086571

4             565         259     37.85    -122.25        7768.086571

...           ...         ...       ...        ...                ...

20635         845         330     39.48    -121.09      162031.481121

20636         356         114     39.49    -121.21      160445.433537
```

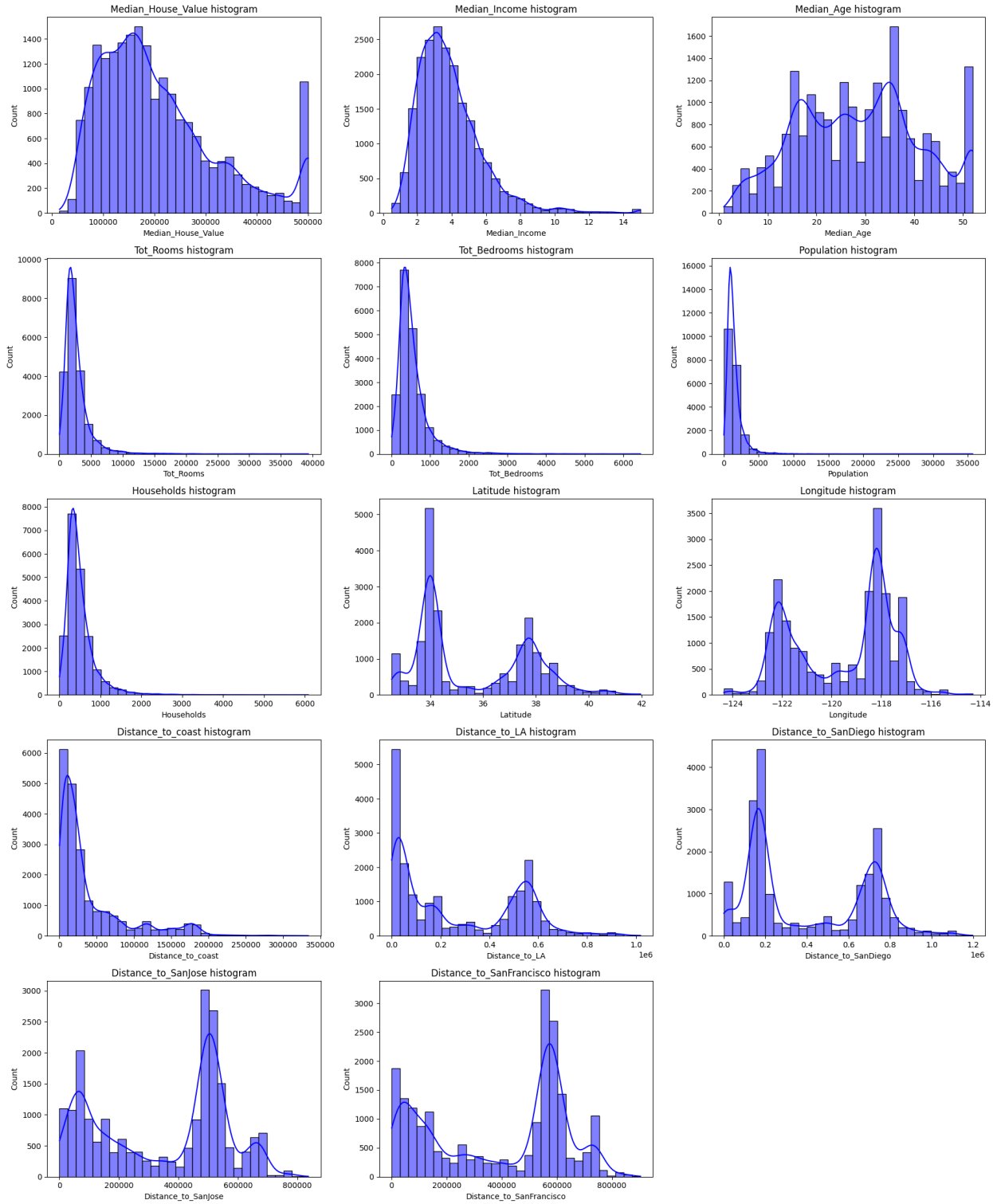
```
20637        1007         433     39.43    -121.22      153754.341182

20638         741         349     39.43    -121.32      152005.022239

20639        1387         530     39.37    -121.24      146866.196892


       Distance_to_LA  Distance_to_SanDiego  Distance_to_SanJose  \
0       556529.158342         735501.806984         67432.517001
1       554279.850069         733236.884360         65049.908574
2       554610.717069         733525.682937         64867.289833
3       555194.266086         734095.290744         65287.138412
4       555194.266086         734095.290744         65287.138412
...               ...                   ...                  ...
20635   654530.186299         830631.543047        248510.058162
20636   659747.068444         836245.915229        246849.888948
20637   654042.214020         830699.573163        240172.220489
20638   657698.007703         834672.461887        238193.865909
20639   648723.337126         825569.179028        233282.769063

       Distance_to_SanFrancisco
0                   21250.213767
1                   20880.600400
2                   18811.487450
3                   18031.047568
4                   18031.047568
...                          ...
20635             222619.890417
20636             218314.424634
20637             212097.936232
20638             207923.199166
20639             205473.376575

[20640 rows x 14 columns]

df.duplicated().sum()

np.int64(0)

df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 14 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Median_House_Value        20640 non-null  float64
 1   Median_Income             20640 non-null  float64
 2   Median_Age                20640 non-null  int64
 3   Tot_Rooms                 20640 non-null  int64
```

```
 4   Tot_Bedrooms               20640 non-null  int64
 5   Population                 20640 non-null  int64
 6   Households                 20640 non-null  int64
 7   Latitude                   20640 non-null  float64
 8   Longitude                  20640 non-null  float64
 9   Distance_to_coast          20640 non-null  float64
 10  Distance_to_LA             20640 non-null  float64
 11  Distance_to_SanDiego       20640 non-null  float64
 12  Distance_to_SanJose        20640 non-null  float64
 13  Distance_to_SanFrancisco  20640 non-null  float64
dtypes: float64(9), int64(5)
memory usage: 2.2 MB

df.describe()

       Median_House_Value  Median_Income    Median_Age
Tot_Rooms  \
count        20640.000000   20640.000000  20640.000000  20640.000000

mean        206855.816909       3.870671     28.639486   2635.763081

std         115395.615874       1.899822     12.585558   2181.615252

min          14999.000000       0.499900      1.000000      2.000000

25%         119600.000000       2.563400     18.000000   1447.750000

50%         179700.000000       3.534800     29.000000   2127.000000

75%         264725.000000       4.743250     37.000000   3148.000000

max         500001.000000      15.000100     52.000000  39320.000000


       Tot_Bedrooms    Population    Households      Latitude
Longitude  \
count  20640.000000  20640.000000  20640.000000  20640.000000
20640.000000
mean     537.898014   1425.476744    499.539680     35.631861   -
119.569704
std      421.247906   1132.462122    382.329753      2.135952
2.003532
min        1.000000      3.000000      1.000000     32.540000   -
124.350000
25%      295.000000    787.000000    280.000000     33.930000   -
121.800000
50%      435.000000   1166.000000    409.000000     34.260000   -
118.490000
75%      647.000000   1725.000000    605.000000     37.710000   -
118.010000
max     6445.000000  35682.000000   6082.000000     41.950000   -
```

```
114.310000
```

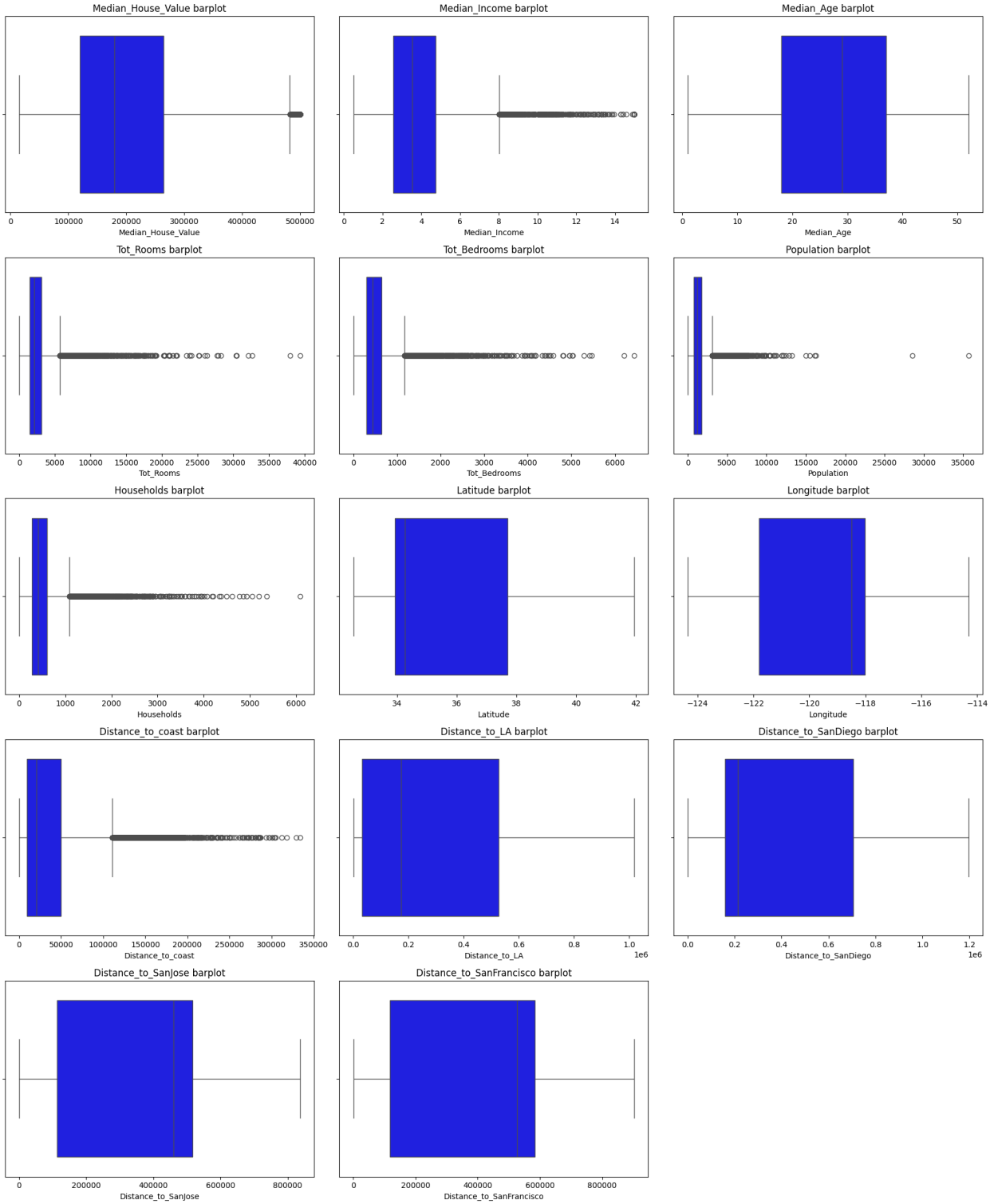
```
       Distance_to_coast  Distance_to_LA  Distance_to_SanDiego  \
count       20640.000000    2.064000e+04          2.064000e+04
mean        40509.264883    2.694220e+05          3.981649e+05
std         49140.039160    2.477324e+05          2.894006e+05
min           120.676447    4.205891e+02          4.849180e+02
25%          9079.756762    3.211125e+04          1.594264e+05
50%         20522.019101    1.736675e+05          2.147398e+05
75%         49830.414479    5.271562e+05          7.057954e+05
max        333804.686371    1.018260e+06          1.196919e+06

       Distance_to_SanJose  Distance_to_SanFrancisco
count         20640.000000              20640.000000
mean         349187.551219             386688.422291
std          217149.875026             250122.192316
min             569.448118                456.141313
25%          113119.928682             117395.477505
50%          459758.877000             526546.661701
75%          516946.490963             584552.007907
max          836762.678210             903627.663298
```

```
plt.figure(figsize=(18, 30))

for i, bf in enumerate(df.columns):
    plt.subplot(7, 3, i+1)
    sns.histplot(x=bf, data=df, kde=True, color='blue', bins=30)

    plt.title(f'{bf} histogram')
plt.tight_layout()
plt.show()
```




```
plt.figure(figsize=(18, 30))
for i, bf in enumerate(df.columns):
    plt.subplot(7, 3, i+1)
    sns.boxplot(x=bf, data=df,  color='blue')
```

```
    plt.title(f'{bf} barplot')
plt.tight_layout()
plt.show()
```



Based on the feature distributions:

- **Log/Power Transform for Skewed Features**:

-- Median_Income, Tot_Rooms, Tot_Bedrooms, Population, Households, and all Distance_to_* features.

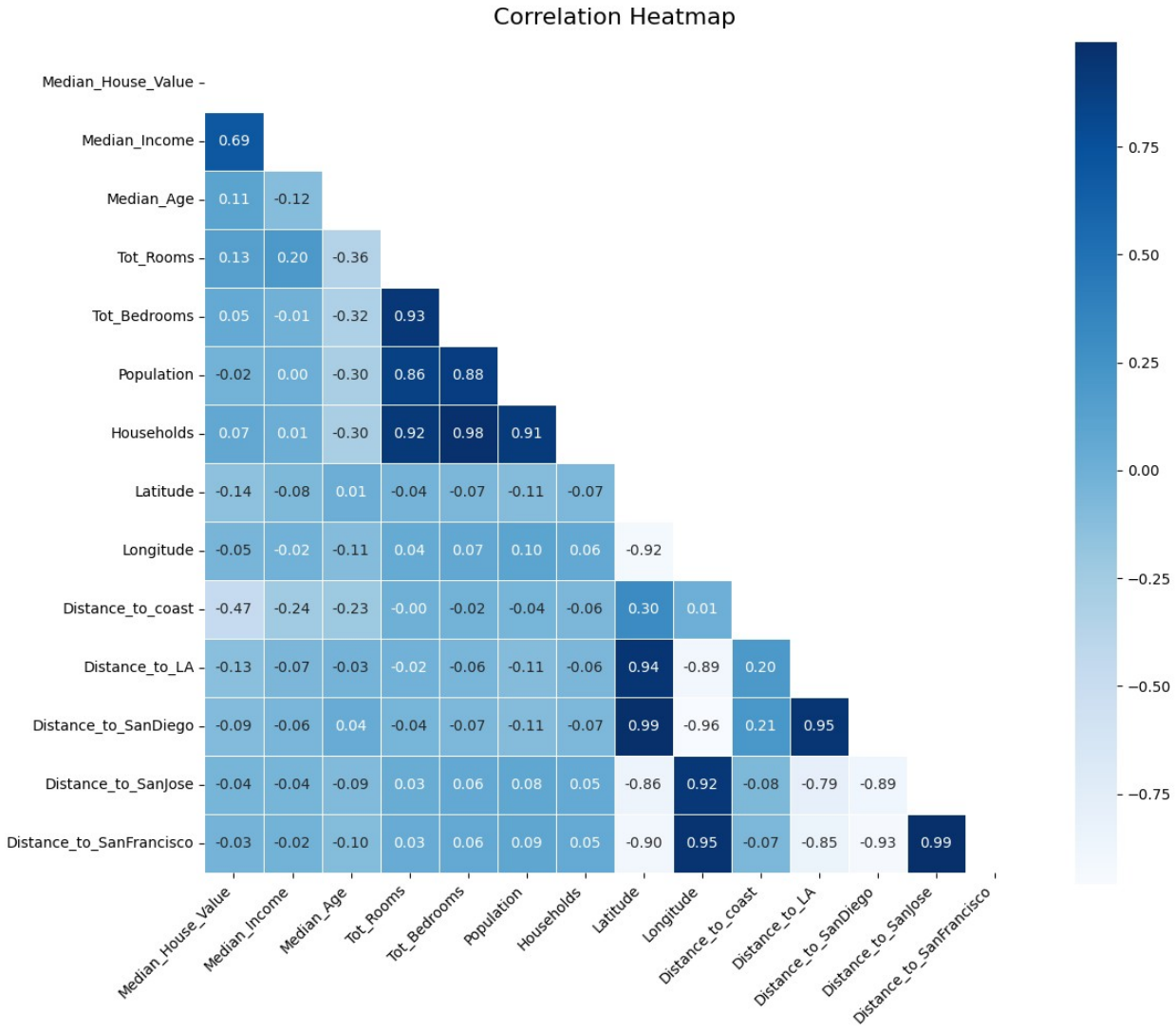
- **Handle Other Features Separately**:

-- Median_Age ,Latitude and Longitude : Don't transform

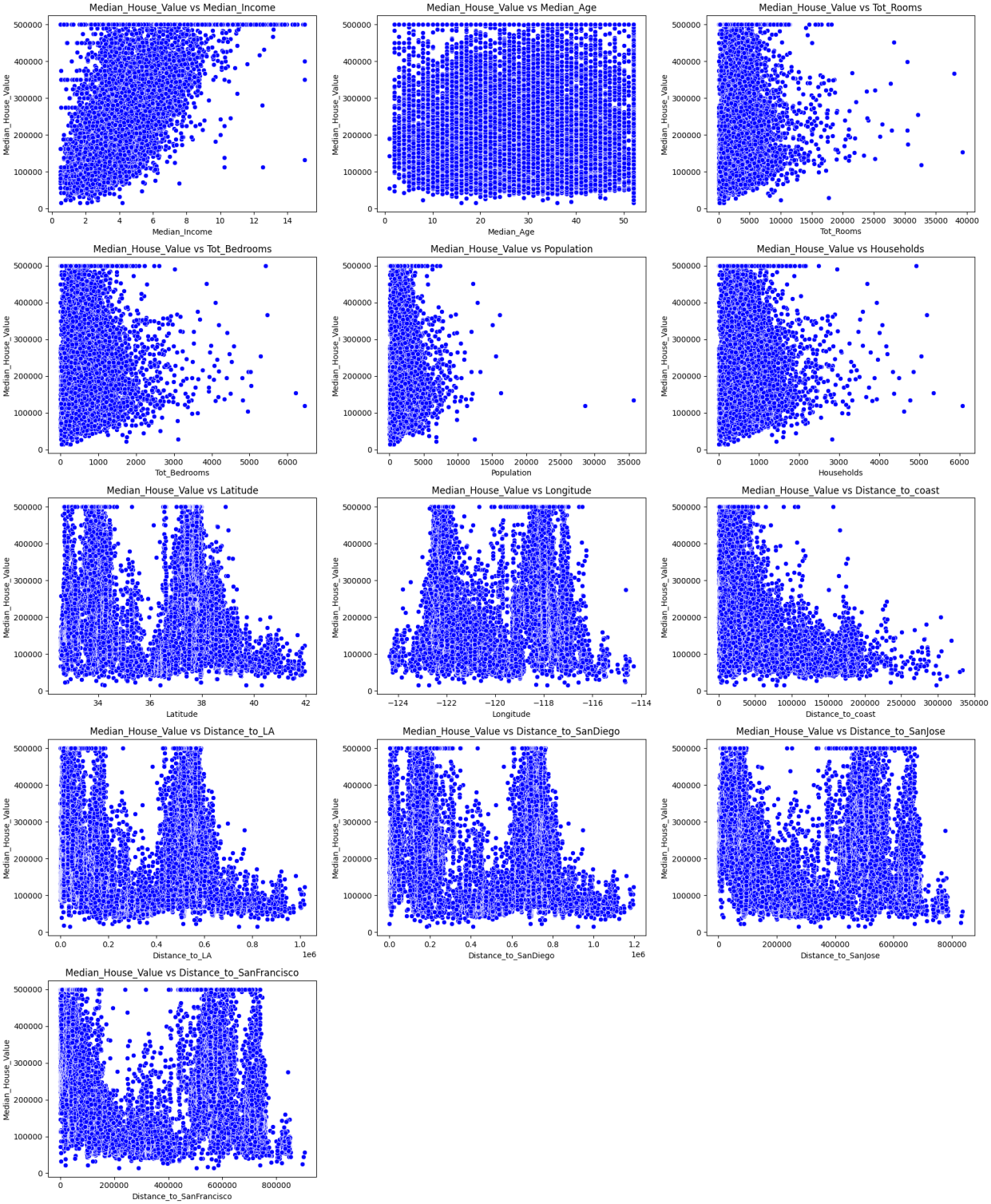
- **Final Scaling**:

-- After transformations we will apply `StandardScaler` on all the features.

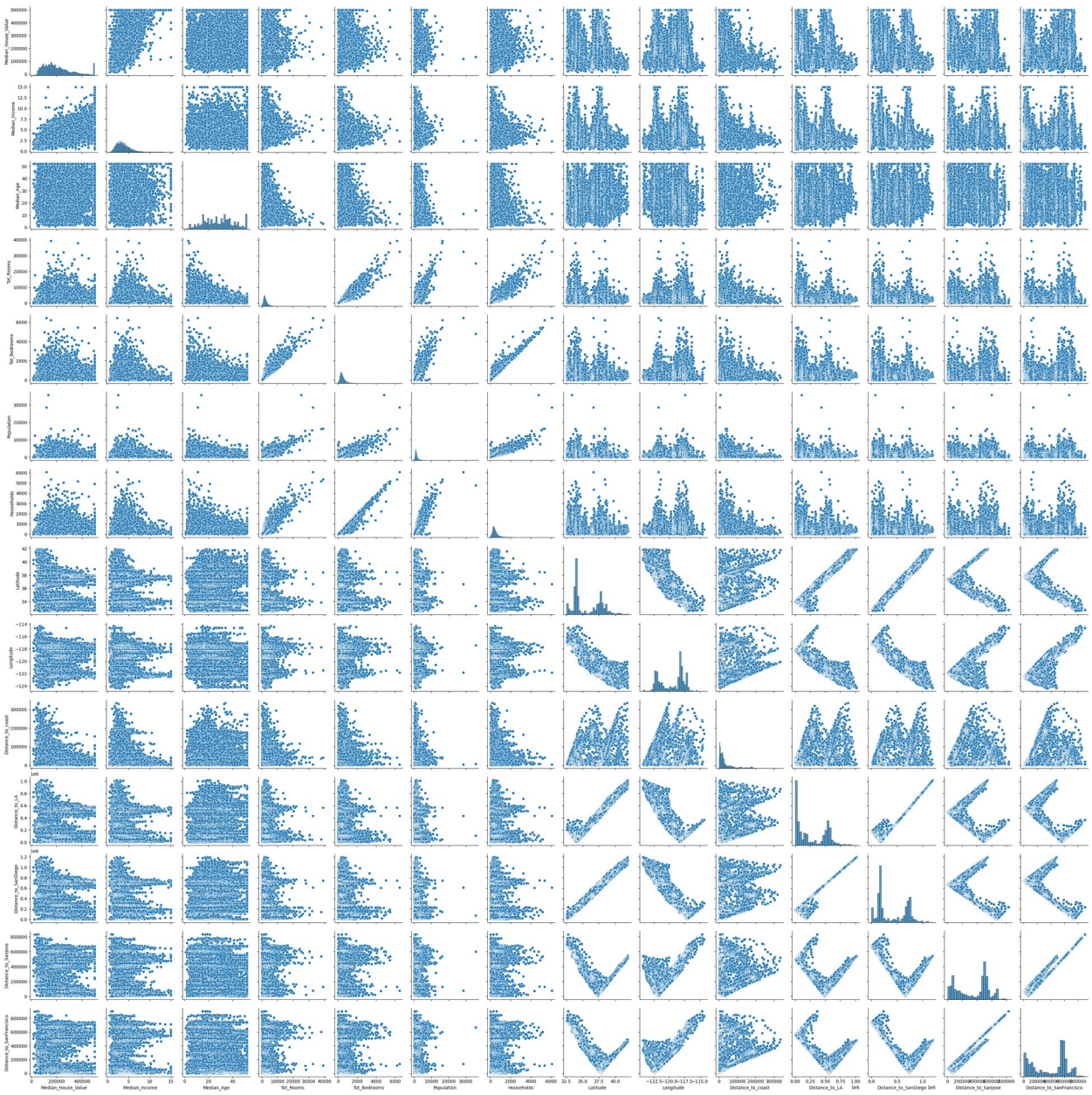
```
corr = df.corr()


mask = np.triu(np.ones_like(corr, dtype=bool))
plt.figure(figsize=(12, 10))


sns.heatmap( corr,mask=mask,  annot=True,fmt=".2f",
cmap='Blues',cbar=True,square=True,linewidths=0.5,annot_kws={"size":
10})

plt.title('Correlation Heatmap', fontsize=16, pad=20)
plt.xticks(rotation=45, ha='right', fontsize=10)
plt.yticks(rotation=0, fontsize=10)

plt.tight_layout()
plt.show()
```




```
plt.figure(figsize=(18, 30))

for i, bf in
enumerate(df.drop(columns=['Median_House_Value']).columns):
    plt.subplot(7, 3, i+1)
    sns.scatterplot(data=df, x=bf, y='Median_House_Value',
color='blue')
    plt.title(f'Median_House_Value vs {bf}')
    plt.xlabel(bf)
    plt.ylabel('Median_House_Value')
plt.tight_layout()
plt.show()
```




```
sns.pairplot(df)
plt.show()
```



Analysis of Multicollinearity

There is strong evidence of severe multicollinearity in the dataset. This will significantly impact the regression models.

```
* `Households` and `Tot_Bedrooms`: **0.98**
* `Households` and `Tot_Rooms`: **0.92**
* `Tot_Rooms` and `Tot_Bedrooms`: **0.93**
* `Population` and `Households`: **0.91**
* `Longitude` and `Distance_to_SanDiego`: **0.99**
* `Latitude` and `Distance_to_Sanjose`: **0.95**
* `Longitude` and `Latitude`: **-0.92**
```

so we will try to remove to manually remove the highly correlated features and keeping just one from each group to solve the multicollinearity problem.
