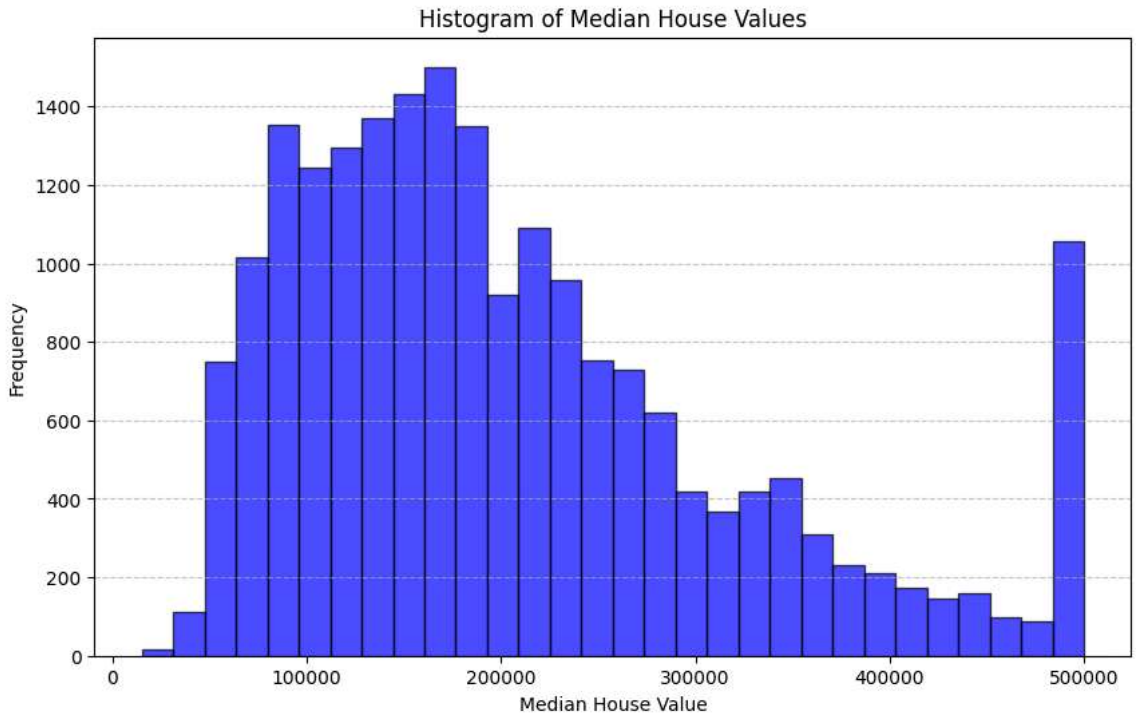
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression


data = pd.read_csv('housing.csv')


data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   longitude           20640 non-null  float64
 1   latitude            20640 non-null  float64
 2   housing_median_age  20640 non-null  float64
 3   total_rooms         20640 non-null  float64
 4   total_bedrooms      20433 non-null  float64
 5   population          20640 non-null  float64
 6   households          20640 non-null  float64
 7   median_income       20640 non-null  float64
 8   median_house_value  20640 non-null  float64
 9   ocean_proximity     20640 non-null  object 
dtypes: float64(9), object(1)
memory usage: 1.6+ MB
```

```
data.columns
```

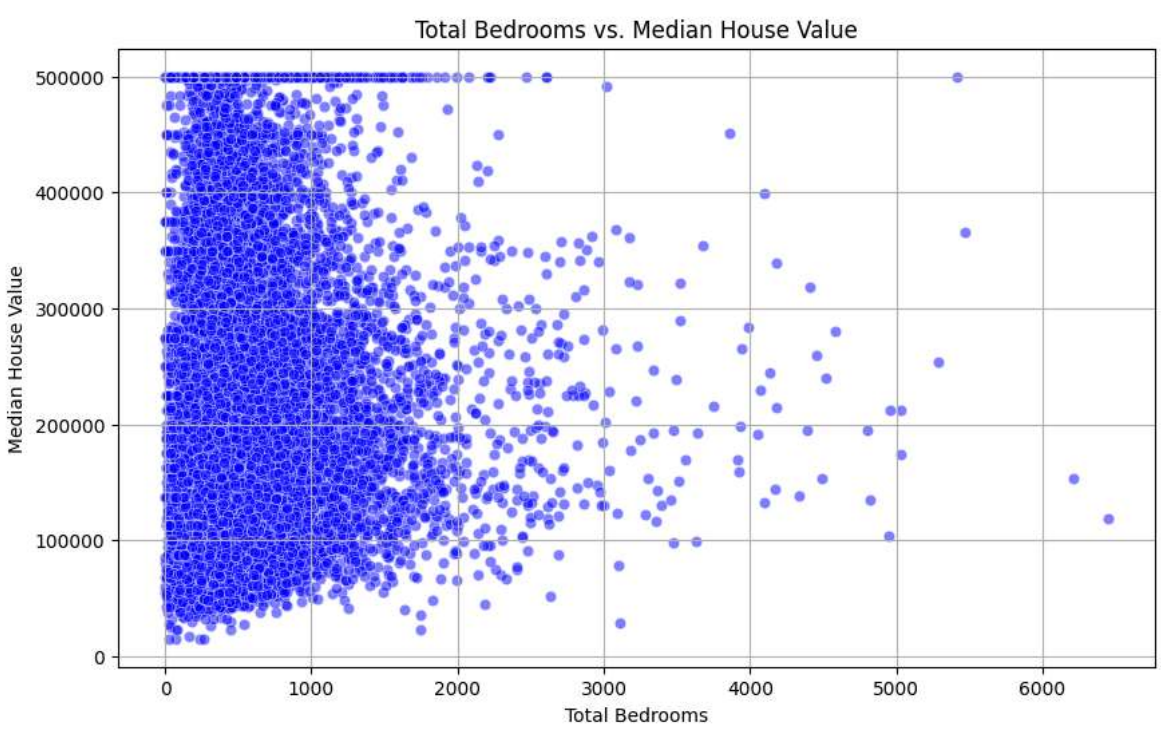
```
Index(['longitude', 'latitude', 'housing_median_age', 'total_rooms',
       'total_bedrooms', 'population', 'households', 'median_income',
       'median_house_value', 'ocean_proximity'],
      dtype='object')
```

```
median_house_values = data['median_house_value']

# Create a histogram
plt.figure(figsize=(10, 6))
plt.hist(median_house_values, bins=30, color='blue', edgecolor='black', alpha=0.7)
plt.title('Histogram of Median House Values')
plt.xlabel('Median House Value')
plt.ylabel('Frequency')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

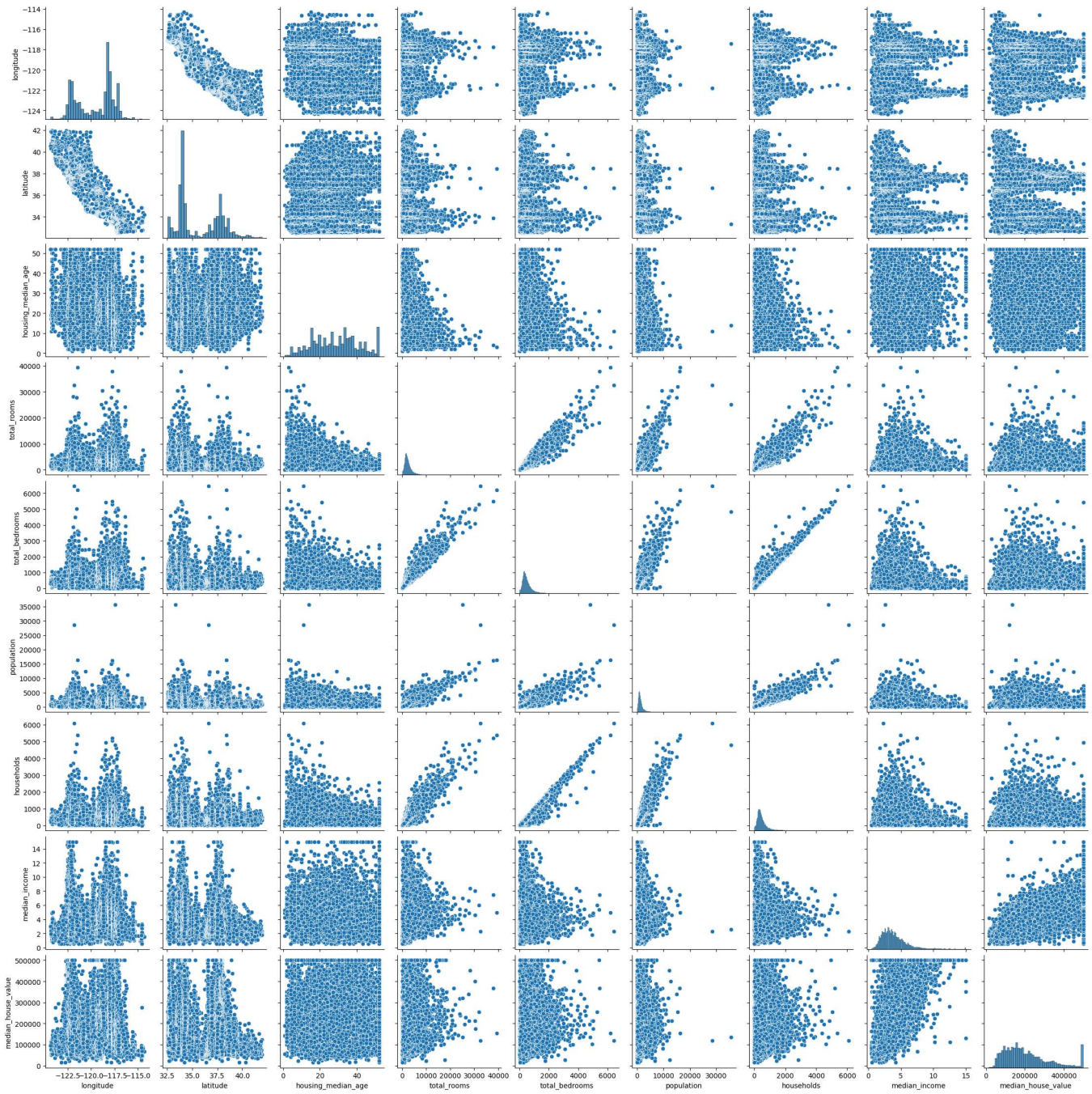



```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='total_bedrooms', y='median_house_value', data=data, alpha=0.5, color='blue')
plt.title('Total Bedrooms vs. Median House Value')
```

```
plt.xlabel('Total Bedrooms')
plt.ylabel('Median House Value')
plt.grid(True)
plt.show()
```




```
sns.pairplot(data)
plt.figure(figsize=(8,5))
plt.show()
```

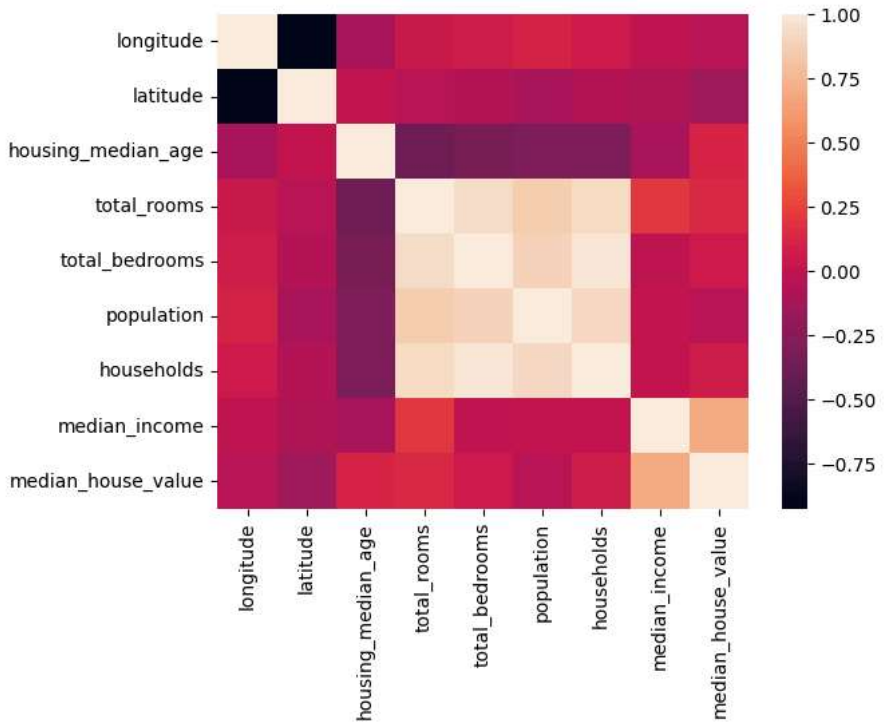


<Figure size 800x500 with 0 Axes>

```
# Remove non-numeric columns
numeric_data = data.select_dtypes(exclude=['object'])

# Calculate the correlation matrix
corr_matrix = numeric_data.corr()
```

```
# Plot the heatmap
sns.heatmap(corr_matrix)
```

```
<Axes: >
```




```
x=data[['longitude', 'latitude', 'housing_median_age', 'total_rooms',
       'total_bedrooms', 'population', 'households', 'median_income',
       'median_house_value', 'ocean_proximity']]
```

```
y=data['median_house_value']
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=42)
```

```
x_train
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_val
7061	-118.02	33.93	35.0	2400.0	398.0	1218.0	408.0	4.1312	19380(
14689	-117.09	32.79	20.0	2183.0	534.0	999.0	496.0	2.8631	16970(
17323	-120.14	34.59	24.0	1601.0	282.0	731.0	285.0	4.2026	25980(
10056	-121.00	39.26	14.0	810.0	151.0	302.0	138.0	3.1094	13610(
15750	-122.45	37.77	52.0	3188.0	708.0	1526.0	664.0	3.3068	50000
...	...	...	...	...	...	...	...	...	
11284	-117.96	33.78	35.0	1330.0	201.0	658.0	217.0	6.3700	22920(
11964	-117.43	34.02	33.0	3084.0	570.0	1753.0	449.0	3.0500	9780(
5390	-118.38	34.03	36.0	2101.0	569.0	1756.0	527.0	2.9344	22210(
860	-121.96	37.58	15.0	3575.0	597.0	1777.0	559.0	5.7192	28350(
15795	-122.42	37.77	52.0	4226.0	1315.0	2619.0	1242.0	2.5755	32500(

14448 rows × 10 columns

```
from sklearn.linear_model import LinearRegression
```

```
from sklearn.ensemble import RandomForestRegressor
```

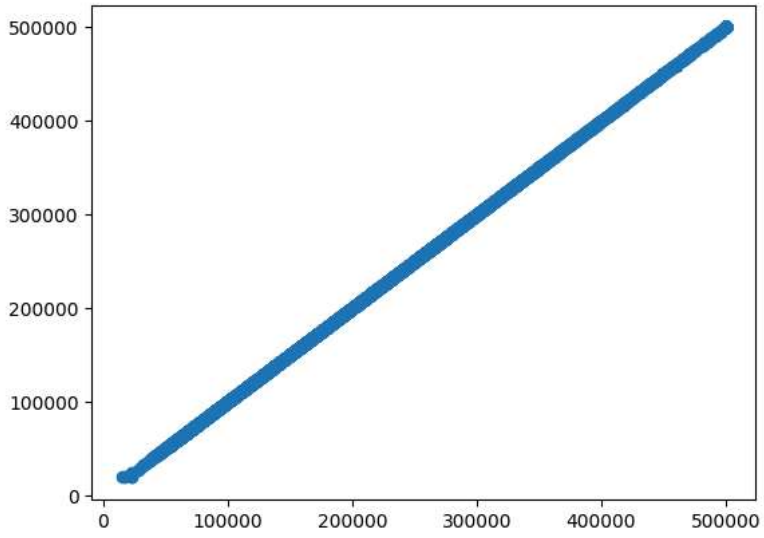
```
rg=LinearRegression()
```

```
x_train_encoded = pd.get_dummies(x_train)
x_test_encoded = pd.get_dummies(x_test)

# Instancier le modèle
rg = RandomForestRegressor()

# Entraîner le modèle
rg.fit(x_train_encoded, y_train)
```

```
▾ RandomForestRegressor
RandomForestRegressor()
```

```
predictions = rg.predict(x_test_encoded)
```

```
scatter=plt.scatter(y_test,predictions)
```




```
sns.displot(y_test-predictions,bins=50);
```