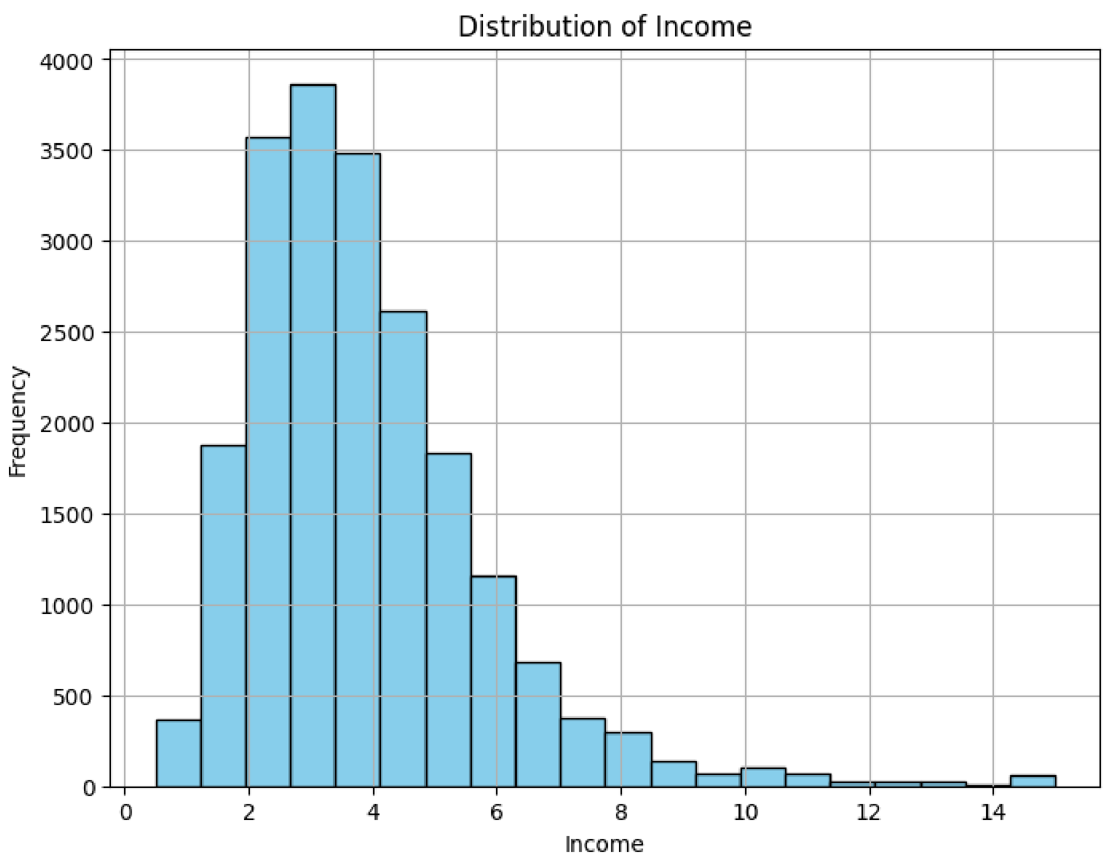
mandatory-project-advanced

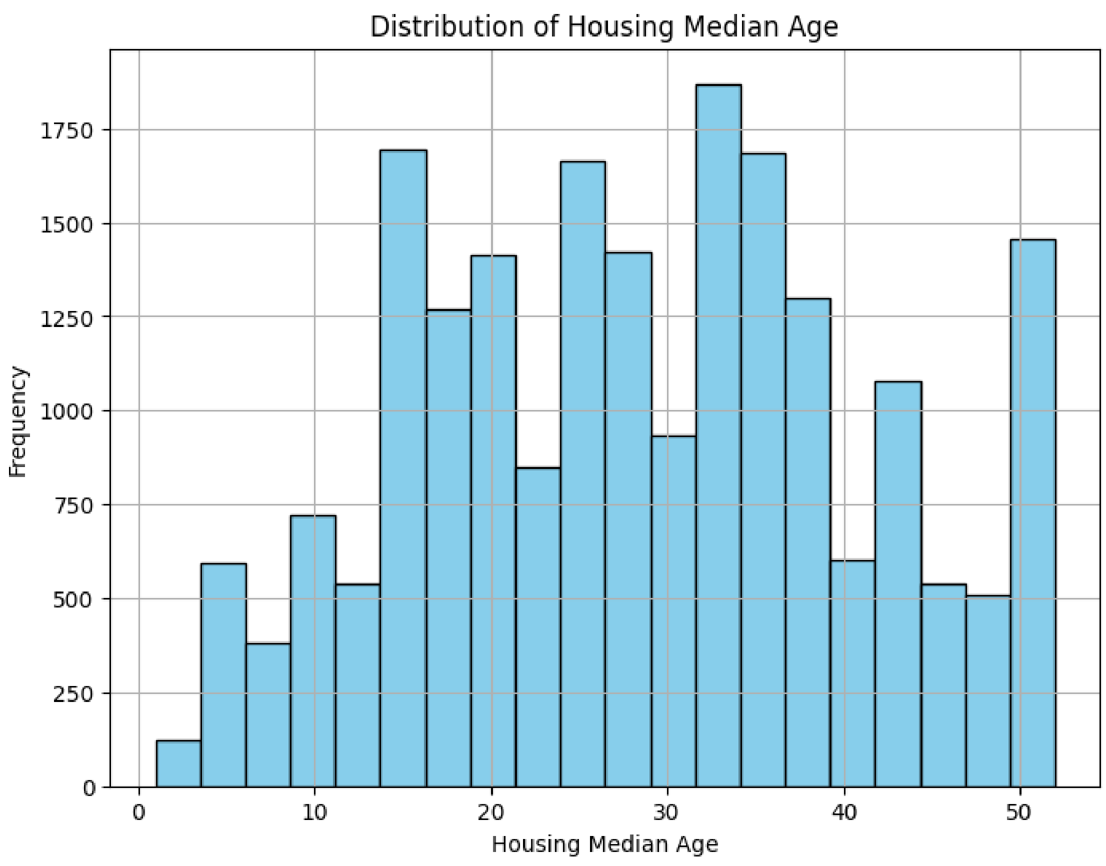
July 27, 2023

[45]:
```python
#1. What is the average median income of the data set and check the␣
↪distribution of data using appropriate plots.
    # Please explain the distribution of the plot.
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Read the data from the CSV file
data = pd.read_excel("dataset.xlsx")

# Extract the 'income' column
income = data["median_income"]

# Calculate the average median income
average_income = np.mean(income)

# Print the average median income
print("Average Median Income:", average_income)

# Plot the distribution of income using a histogram
plt.figure(figsize=(8, 6))
plt.hist(income, bins=20, edgecolor='black', color='skyblue')
plt.xlabel('Income')
plt.ylabel('Frequency')
plt.title('Distribution of Income')
plt.grid(True)
plt.show()
```

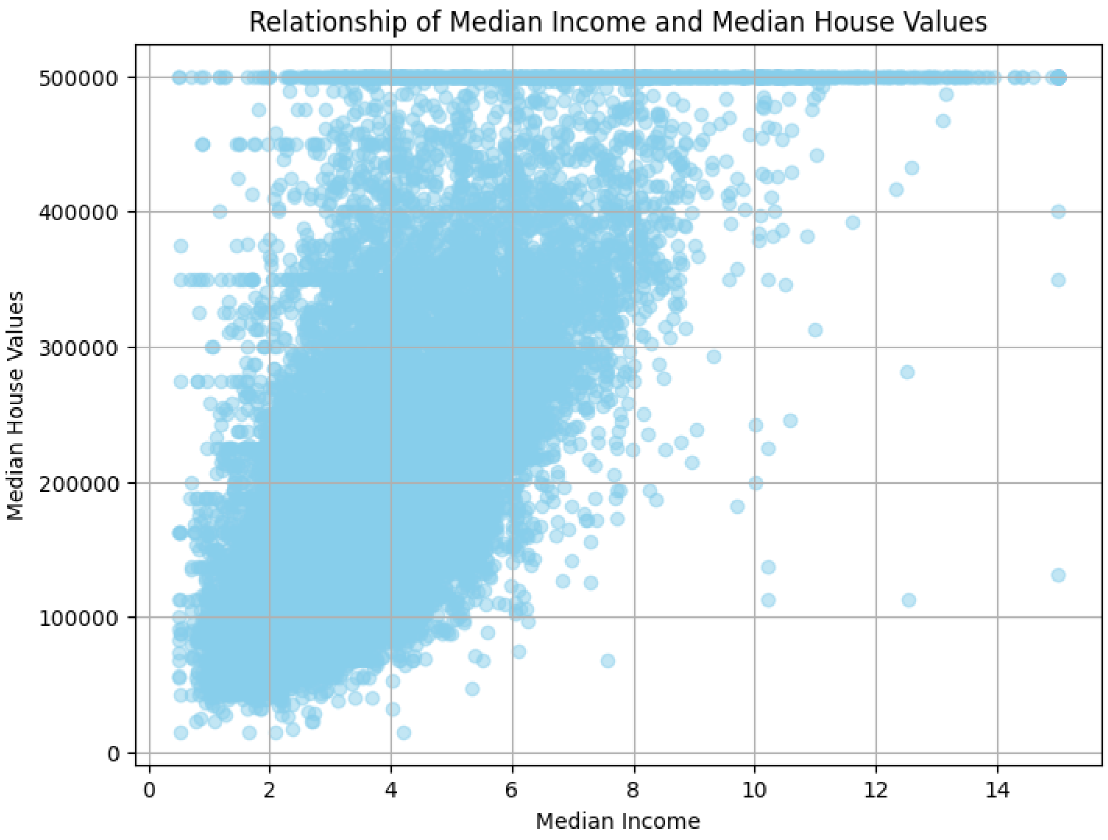
```
Average Median Income: 3.8706710029069766
```



[46]:
```
#2. Draw an appropriate plot to see the distribution of housing_median_age and␣
 ↪explain your observations.
import pandas as pd
import matplotlib.pyplot as plt

#As i already uplaod the dataset by name = data

# Extract the 'housing_median_age' column from the DataFrame
To_get_housing_median_age = data['housing_median_age']

# Create a histogram to visualize the distribution of 'housing_median_age'
plt.figure(figsize=(8, 6))
plt.hist(To_get_housing_median_age, bins=20, edgecolor='black', color='skyblue')
plt.xlabel('Housing Median Age')
plt.ylabel('Frequency')
plt.title('Distribution of Housing Median Age')
plt.grid()
plt.show()
```



[44]:
```
#3. Show with the help of visualization, how median_income and␣
 ↪median_house_values are related?

import pandas as pd
import matplotlib.pyplot as plt

# Assuming you have already loaded your dataset into a pandas DataFrame named␣
 ↪'data'

# Extract the 'median_income' and 'median_house_values' columns from the␣
 ↪DataFrame
median_income = data['median_income']
median_house_values= data['median_house_value']

# Create a scatter plot to visualize the relationship between 'median_income'␣
 ↪and 'median_house_values'
plt.figure(figsize=(8, 6))
plt.scatter(median_income, median_house_values, color='skyblue', alpha=0.5)
plt.xlabel('Median Income')
plt.ylabel('Median House Values')
plt.title('Relationship of Median Income and Median House Values')
plt.grid()
plt.show()
```



[ ]:
```
#4.Create a data set by deleting the corresponding examples from the data set␣
 ↪for which total_bedrooms are not available.
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_excel("dataset.xlsx")

# Assuming you have already loaded your dataset into a pandas DataFrame named␣
 ↪'data'

# Create a new dataset by removing rows where 'total_bedrooms' are not available
new_data = data.dropna(subset=['total_bedrooms'])

# Optional: Reset the index of the new dataset to have continuous index numbers
new_data.reset_index(drop=True, inplace=True)
print(new_data.reset_index)
```

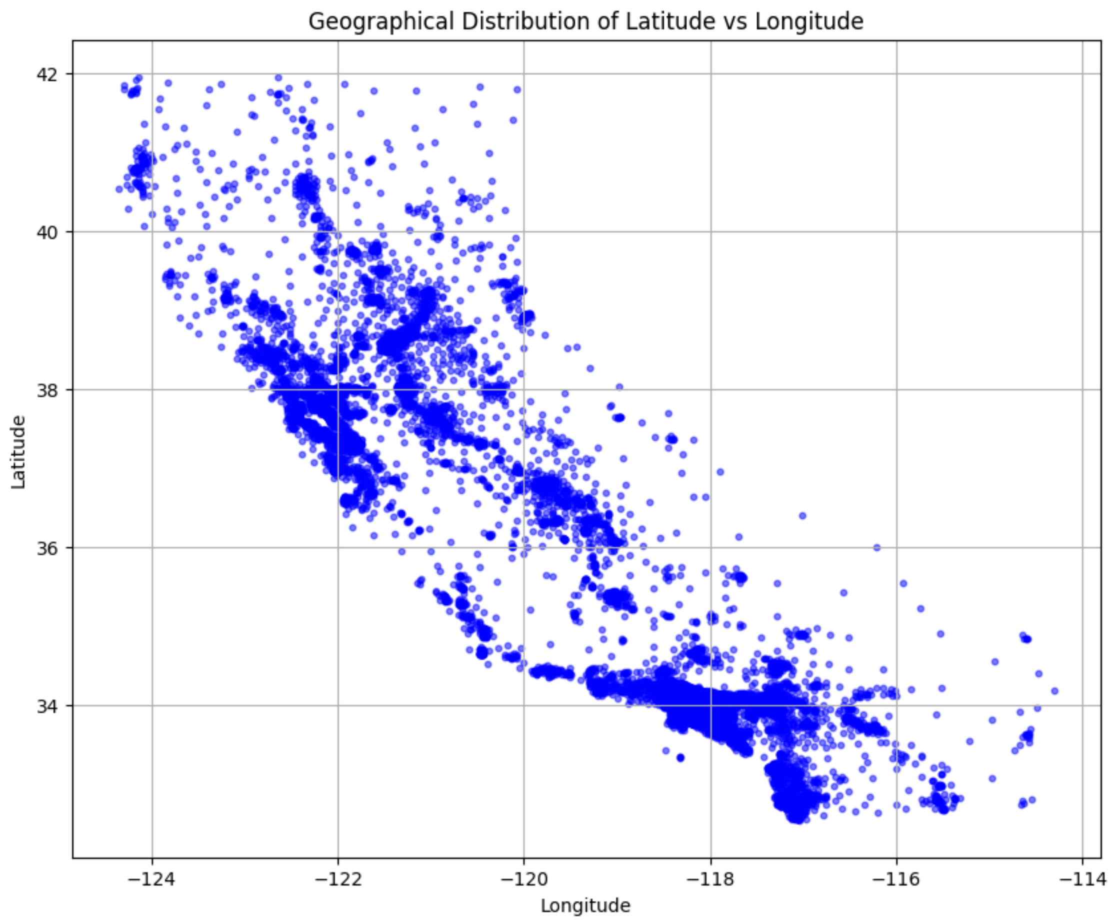
```
# Now, the 'new_data' DataFrame contains the dataset with rows where␣
 ↪'total_bedrooms' are available.
# You can use 'new_data' for further analysis.
```

[27]:
```
#5. Create a data set by filling the missing data with the mean value of the␣
 ↪total_bedrooms in the original data set.

import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_excel("dataset.xlsx")
# Assuming you have already loaded your dataset into a pandas DataFrame named␣
 ↪'data'

# Calculate the mean value of 'total_bedrooms' from the original dataset
mean_total_bedrooms = data['total_bedrooms'].mean()

# Create a new dataset by filling the missing 'total_bedrooms' with the mean␣
 ↪value
new_data = data.copy()  # Create a copy of the original dataset to avoid␣
 ↪modifying it directly
new_data['total_bedrooms'].fillna(mean_total_bedrooms, inplace=True)

# Now, the 'new_data' DataFrame contains the dataset with missing␣
 ↪'total_bedrooms' filled with the mean value.
# You can use 'new_data' for further analysis.
```

[37]:
```
#6. Write a programming construct (create a user defined function) to calculate␣
 ↪the median value of the data set wherever required.
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_excel("dataset.xlsx")
def calculate_median(data):
    # Sort the data in ascending order
    sorted_data = sorted(data)

    # Calculate the number of elements in the dataset
    n = len(sorted_data)

    # Check if the number of elements is odd or even
    if n % 2 == 1:
        # If the number of elements is odd, return the middle value
        median = sorted_data[n // 2]
    else:
        # If the number of elements is even, calculate the average of the two␣
 ↪middle values
        middle_index1 = n // 2 - 1
        middle_index2 = n // 2
        median = (sorted_data[middle_index1] + sorted_data[middle_index2]) / 2

    return median
# Sample dataset
data = ["total_bedrooms"]

# Calculate the median using the user-defined function
median_value = calculate_median(data)
print("Median:", median_value)
```

```
Median: total_bedrooms
```

[32]:
```
#7. Plot latitude versus longitude and explain your observations.

import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_excel("dataset.xlsx")
# Assuming you have already loaded your dataset into a pandas DataFrame named␣
↪'data'

# Extract the 'latitude' and 'longitude' columns from the DataFrame
latitude = data['latitude']
longitude = data['longitude']

# Create a scatter plot to visualize latitude versus longitude
plt.figure(figsize=(10, 8))
plt.scatter(longitude, latitude, s=10, c='blue', alpha=0.5)
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.title('Geographical Distribution of Latitude vs Longitude')
plt.grid(True)
plt.show()
```



[42]:
```
#8.. Create a data set for which the ocean_proximity is ‘Near ocean’.
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_excel("dataset.xlsx")

# Assuming you have already loaded your dataset into a pandas DataFrame named␣
 ↪'data'

# Create a new dataset with rows where 'ocean_proximity' is 'Near Ocean'
near_ocean_data = data[data['ocean_proximity'] == 'Near Ocean']

# Optional: Reset the index of the new dataset to have continuous index numbers
near_ocean_data.reset_index(drop=True, inplace=True)

# Now, the 'near_ocean_data' DataFrame contains the subset of the dataset with␣
 ↪'ocean_proximity' as 'Near Ocean'.
# You can use 'near_ocean_data' for further analysis.
```

[41]:
```
#9.Find the mean and median of the median income for the data set created in␣
 ↪question

import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_excel("dataset.xlsx")
# Assuming you have the dataset in the DataFrame named 'near_ocean_data'

# Calculate the mean of the 'median_income' column
mean_median_income = data['median_income'].mean()

# Calculate the median of the 'median_income' column
median_median_income =data['median_income'].median()

# Print the mean and median
print("Mean of Median Income:", mean_median_income)
print("Median of Median Income:", median_median_income)
```

```
Mean of Median Income: 3.8706710029069766
Median of Median Income: 3.5347999999999997
```

[43]:
```
#10.Please create a new column named total_bedroom_size. If the total bedrooms␣
 ↪is 10 or less, it should be quoted as small.
#   If the total bedrooms is 11 or more but less than 1000, it should be␣
 ↪medium, otherwise it should be considered large.

import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_excel("dataset.xlsx")
import pandas as pd

# Assuming you have already loaded your dataset into a pandas DataFrame named␣
 ↪'data'

# Define a function to categorize the total bedrooms based on size
def categorize_bedroom_size(total_bedrooms):
    if total_bedrooms <= 10:
        return 'Small'
    elif 11 <= total_bedrooms < 1000:
        return 'Medium'
    else:
        return 'Large'

# Apply the function to create the new column 'total_bedroom_size'
data['total_bedroom_size'] = data['total_bedrooms'].
 ↪apply(categorize_bedroom_size)
```

```
# Now, the 'data' DataFrame contains the new column 'total_bedroom_size' with␣
↪the categorized sizes.
# You can use 'data' for further analysis.
```