A.

Research Question

A justification for the research question is that housing is a unique asset. Both an investment and a consumption good, it is traded in markets that are subject to significant search frictions and information asymmetries. In addition, housing accounts for a large share of wealth in the economy. As a result, changes in house prices can have large effects on aggregate economic activity. this makes housing an ideal asset for the study of a range of questions of broader economic interest (What Can Housing Markets Teach Us about Economics?, n.d.).

The context in which the research question exists is as follows. As a data analyst I will be trying to create a predictive model that can estimate the median house value in California so real estate developers can accurately appraise their property. This study will utilize a multiple linear regression model to analyze the significance of predictor variables and their correlation to the median housing value.

The research question is as follows. Can a predictive model be created on the research data set?

Hypothesis:
Null hypothesis-. A MLR model cannot be constructed on the research data set.
Alternate Hypothesis-. A MLR model can be constructed on the research data set with an accuracy greater than 70%.

B.

Data Collection

The data collected for this study is publicly available information provided by the U.S. census bureau (California Housing Prices, n.d.). The data contains information from the 1990 California census. The data set contains 20,640 rows. The data set contains the following variables of longitude, latitude, housing_median_age, total_rooms, total_bedrooms, population, households, median_income, median_house_value, ocean_proximity. The data set is available through kaggle.com.

The data collecting methodology I used was to use data from government publications such as the U.S. census bureau. One advantage of using this data collection methodology is that the data is from a reliable and credible source. One disadvantage of this data collecting methodology is that it is not always up to date. In this case the data was collected in 1990. I did not encounter any challenges during the collection of this data because it was a publicly available CSV file and easily downloaded from the Internet.

C.

Data Extraction and Preparation

My data extraction and preparation process begins by using the pandas python library to read in the data from a CSV file with the read_csv() method. I then remove the index column by manipulating the pandas DataFrame. I then check for missing values with the isna() method. I use the python ffill() method to impute the missing values. The ffill() method imputes missing data by using the last known value in the column. After the missing values are imputed I check for missing values again. Duplicate rows are then detected with the duplicated() method. The categorical variable is one-hot encoded and a constant is added to the data set for the MLR model. The data set is divided into response and predictor variables. Finally the data is split into training and test sets with 80% training and 20% test.

One-hot encoding was used because it is necessary to encode categorical variables into a numerical format for use with a MLR model. One advantage of one-hot encoding is clear coefficients. This means each binary variable will represent a category for straightforward interpretation of the MLR model. One disadvantage of one-hot encoding is the potential for multicolinearity.

The tools I used for data extraction and preparation are the python programming language python and the pandas library. I used these tools and techniques because the pandas library provides a powerful data structure called a DataFrame for tabular data. The read_csv() method assigns the data to a DataFrame variable. This makes the data easy to manipulate. Pandas also provides many efficient methods for imputation of missing values such as the ffill() method. I chose to use the ffill() method because it allows us to preserve data rather than dropping the whole row. Duplicate rows are detected with only one short line of code making pandas and python an easy choice for data preparation.

One advantage of using these tools and techniques with my data extraction and preparation methods is that the python script is reusable. Once I decide on a method of extracting and preparing the data it can be reused on a different data set. One disadvantage of these tools and techniques when used with my data extraction and preparation methods is that python and pandas require dependency management. This means that as a data analyst I have to make sure that the correct versions of python and it's libraries such as pandas have the correct version installed on the machine I am using. This can be complex and time consuming.

In [1]:
```
#import libraries and read in the data from file.
import pandas as pd
```

```
import statsmodels.api as sm
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
# Assuming your CSV file is named 'data.csv', adjust the file path as needed
file_path = '/home/dj/skewl/Capstone/housing.csv'
pd.set_option('display.max_columns', None)
# Read the data from the CSV file into a DataFrame
df = pd.read_csv(file_path)
#drop index column
df = df.loc[:, ~df.columns.str.contains('Unnamed')]

# Identify missing values using isna() method
missing_values = df.isna().sum()
# Print DataFrame with True for missing values and False for non-missing values
print(missing_values)

#replace missing values in children with ffill  method
df['total_bedrooms'].ffill(inplace=True)

# Identify missing values using isna() method
missing_values = df.isna().sum()
# Print DataFrame with True for missing values and False for non-missing values
print(missing_values)

# Find duplicate rows
duplicate_rows = df.duplicated().sum()

# Print duplicate rows   # found NO duplicate rows here!
print(duplicate_rows)


#split continuous and categorical variables into separate dataframes
dfcon = df[['longitude','latitude','housing_median_age','total_rooms','total_bedrooms','population','households','media
dfcat = df[['ocean_proximity']]
#one-hot encode categorical data and drop first level of each
dfcat_encoded = pd.get_dummies(dfcat,drop_first=True)
#concatenate the columns
data = pd.concat([dfcon, dfcat_encoded], axis=1)
#separate independent and dependent variables
y=data['median_house_value']
del data['median_house_value']
#add constant for intercept
data = sm.add_constant(data)
x=data
#split training and test data
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```

```
/usr/lib/python3/dist-packages/scipy/__init__.py:146: UserWarning: A NumPy version >=1.17.3 and <1.25.0 is required for
this version of SciPy (detected version 1.26.4
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
longitude               0
latitude                0
housing_median_age      0
total_rooms             0
total_bedrooms        207
population              0
households              0
median_income           0
median_house_value      0
ocean_proximity         0
dtype: int64
longitude             0
latitude              0
housing_median_age    0
total_rooms           0
total_bedrooms        0
population            0
households            0
median_income         0
median_house_value    0
ocean_proximity       0
dtype: int64
0
```

D.

Exploratory data analysis using box plots, bivariate scatter plots, and Q-Q plots.

Box plots, and scatter plots were created using the matplotlib python library. Q-Q plots were created using the statsmodel library. These libraries were chosen for their ease of use and ability to customize the size and labels.

A box plot was used for a bivariate graph to visualize the continuous dependent variable distribution against a categorical variable. Box plots were used to visualize the distribution of the dependent and independent variables. Bivariate scatter plots were used to visualize the correlation between the continuous dependent variable and the other continuous and discrete independent variables.

Box plots were selected because they serve several purposes such as summary of distribution, outlier detection, and comparison between groups. One advantage is that they work well for large datasets such as the research data set. One disadvantage of the box plot is that they lack some of the detail of other graphs because they do not show all data points.

Scatter plots were selected because they provide a way to visualize relationships between two continuous variables. One advantage of a scatter plot is that they can also be used to detect outliers. One disadvantage of a scatter plot is that overplotting can be an issue with large data sets. This causes the points to overlap each other and make the plot difficult to interpret.

Q-Q plots were used to determine the normality of the data set.

Analysis with MLR model and Q-Q plots.

Q-Q plots and the MLR model were created using the statsmodel library. This library was chosen because of the ease of use and the comprehensive statistical output from model.summary().

Q-Q plots were used to visualize the normality of the residuals of the MLR model.

Q-Q plots were selected because they can help assess the normality of a data set. One advantage of a Q-Q plot is that they are easy to interpret. One disadvantage of a Q-Q plot is that they do not provide a formal measure of goodness of fit. They are only a visual representation and lack the statistical significance of hypothesis testing with a Shapiro-Wilk test.

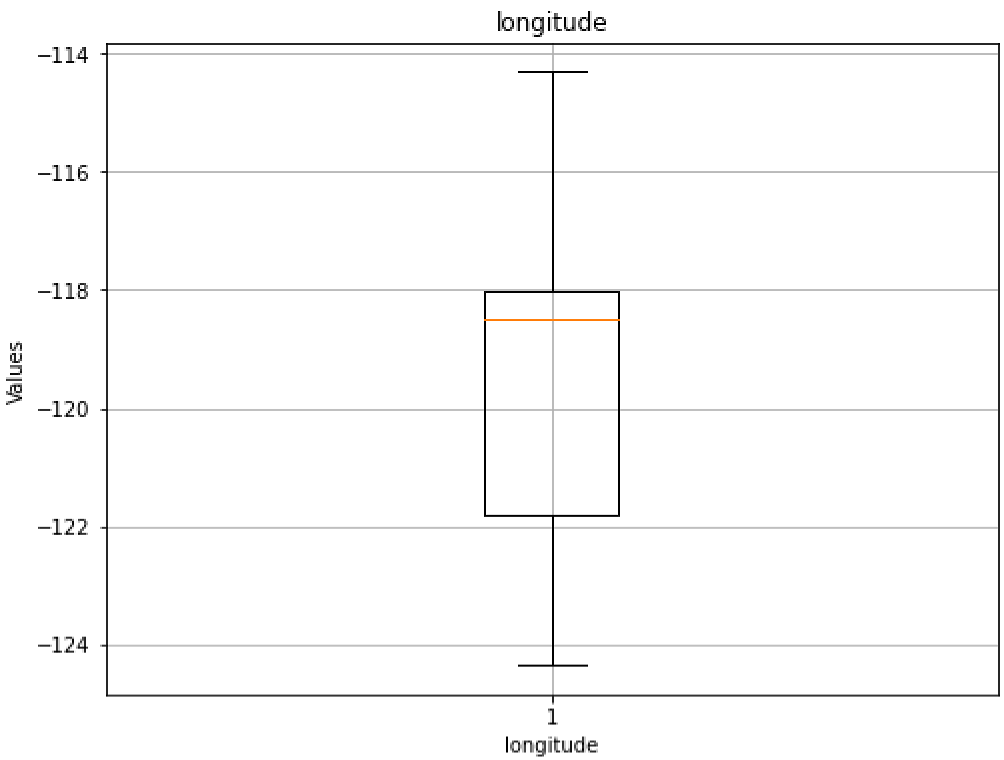
A Multiple Linear Regression model was used to answer the research question. A MLR model was selected because the response variable in the research data set was continuous and the predictor variables were continuous, discrete, and categorical. Multiple linear regression is used to model the relationship between a continuous response variable and continuous or categorical explanatory variables (Multiple Linear Regression, n.d.). One advantage of a MLR model is that the model provides metrics for goodness of fit such as R-squared and adjusted R-squared. One disadvantage of a MLR model is that this technique assumes a linear relationship between the independent and dependent variables.

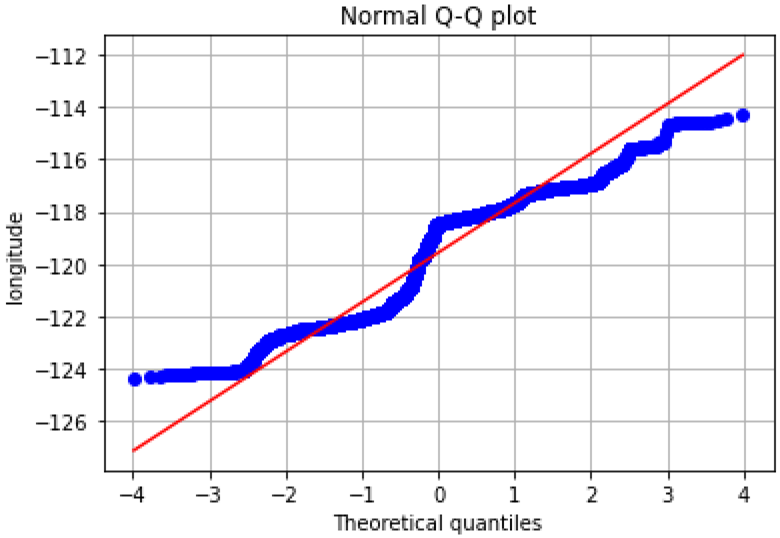
In [2]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import scipy.stats as stats
import statsmodels.api as sm

#function for box plot
def plot_boxplot(data, title="Box Plot", xlabel="Data"):
    plt.figure(figsize=(8, 6))  # Adjust figure size if needed
    plt.boxplot(data)
    plt.title(title)
    plt.xlabel(xlabel)
    plt.ylabel("Values")
```

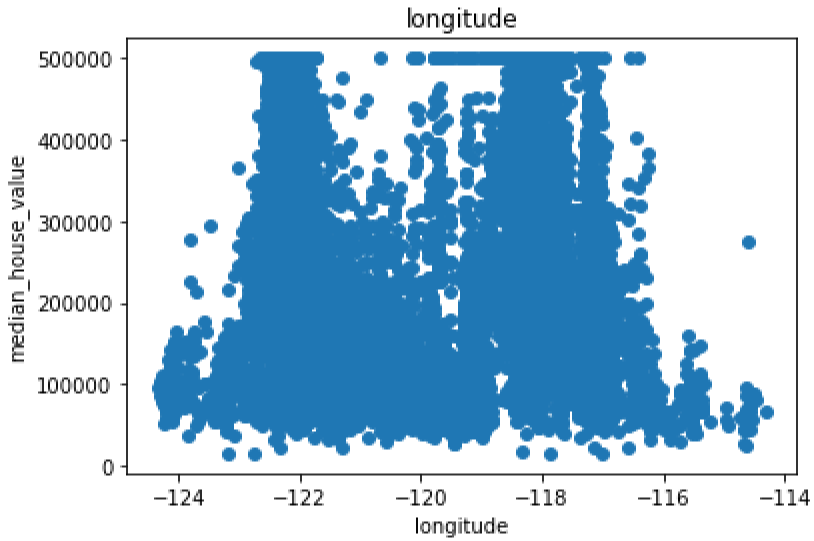
```
    plt.grid(True)  # Add grid lines for better readability
    plt.show()

#function for qq plot
def qq_plot(data, column_name='ylabel'):
    stats.probplot(data, dist="norm", plot=plt)
    plt.title('Normal Q-Q plot')
    plt.xlabel('Theoretical quantiles')
    plt.ylabel(column_name)
    plt.grid(True)
    plt.show()

#function for bivariate scatter plot
def scatter_plot(x,  title='Scatter Plot', xlabel=''):

    # Plot the scatter plot
    plt.scatter(x, df['median_house_value'])

    # Set labels and title
    plt.title(title)
    plt.xlabel(xlabel)
    plt.ylabel('median_house_value')

    # Show plot
    plt.show()

def box_plot(indep):
    # Box plot for categorical and continuous variable
    df.boxplot(column='median_house_value', by=indep)
    plt.title('Box Plot',y=.5)
    plt.xlabel(indep)
    plt.ylabel('median_house_value')
    plt.show()


#EDA for longitude

plot_boxplot(df['longitude'], title="longitude", xlabel="longitude")

qq_plot(df['longitude'],'longitude')

scatter_plot(df['longitude'], title='longitude', xlabel='longitude')

#EDA for latitude

plot_boxplot(df['latitude'], title="latitude", xlabel="latitude")

qq_plot(df['latitude'],'latitude')
```

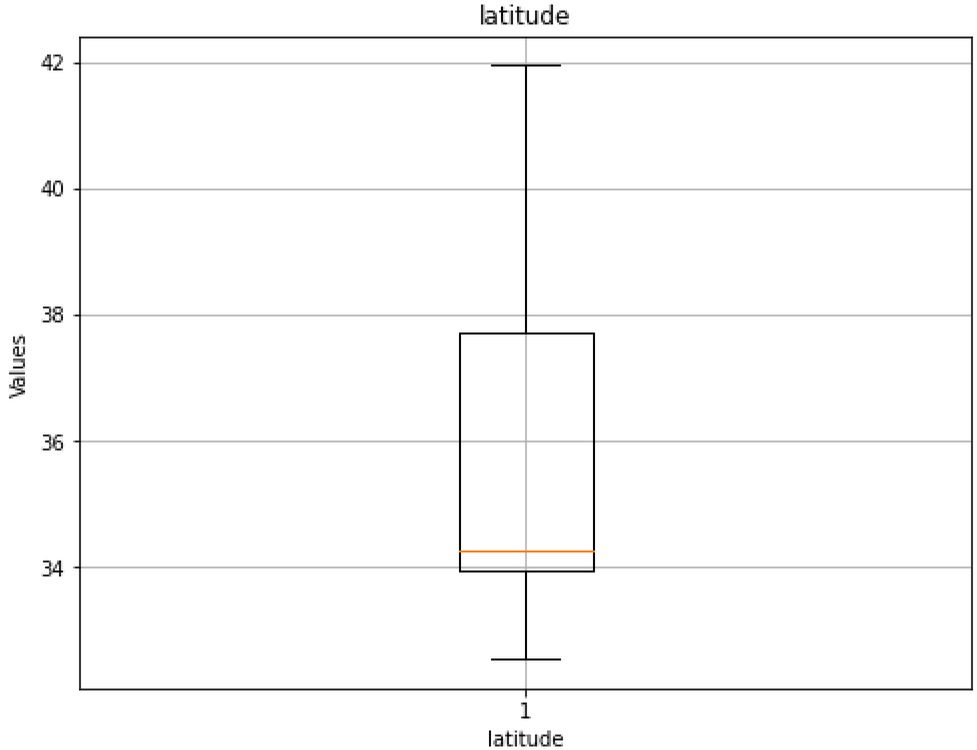
```
scatter_plot(df['latitude'], title='latitude', xlabel='latitude')

#EDA for housing_median_age
plot_boxplot(df['housing_median_age'], title="housing_median_age", xlabel="housing_median_age")
qq_plot(df['housing_median_age'],'housing_median_age')
scatter_plot(df['housing_median_age'], title='housing_median_age', xlabel='housing_median_age')


#EDA for toal rooms
plot_boxplot(df['total_rooms'], title="total_rooms", xlabel="total_rooms")
qq_plot(df['total_rooms'],'total_rooms')
scatter_plot(df['total_rooms'], title='total_rooms', xlabel='total_rooms')
#EDA for toal bedrooms
plot_boxplot(df['total_bedrooms'], title="total_bedrooms", xlabel="total_bedrooms")
qq_plot(df['total_bedrooms'],'total_bedrooms')
scatter_plot(df['total_bedrooms'], title='total_bedrooms', xlabel='total_bedrooms')

#EDA for  population
plot_boxplot(df['population'], title="population", xlabel="population")
qq_plot(df['population'],'population')
scatter_plot(df['population'], title='population', xlabel='population')

#EDA for  households
plot_boxplot(df['households'], title="households", xlabel="households")
qq_plot(df['households'],'households')
scatter_plot(df['households'], title='households', xlabel='households')
```

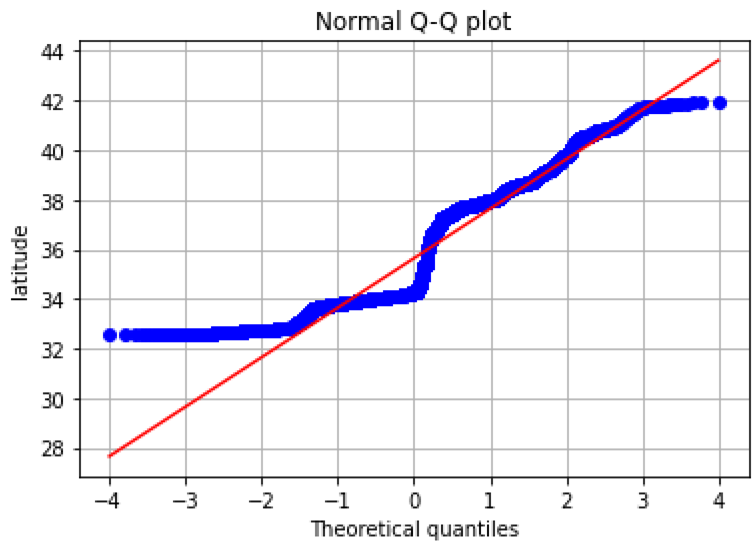
```
#EDA for median_income

plot_boxplot(df['median_income'], title="median_income", xlabel="median_income")

qq_plot(df['median_income'],'median_income')

scatter_plot(df['median_income'], title='median_income', xlabel='median_income')


#EDA for median_house_value

plot_boxplot(df['median_house_value'], title="median_house_value", xlabel="median_house_value")

qq_plot(df['median_house_value'],'median_house_value')


#EDA for ocean_proximity
box_plot('ocean_proximity')
```

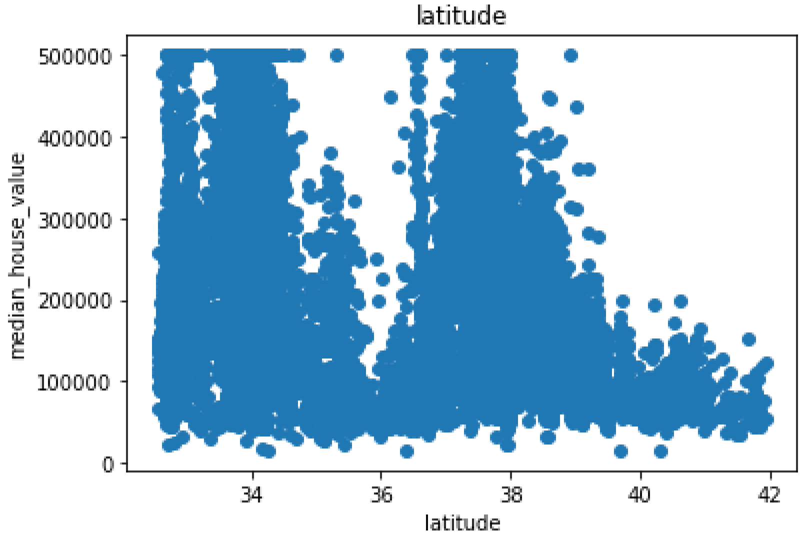


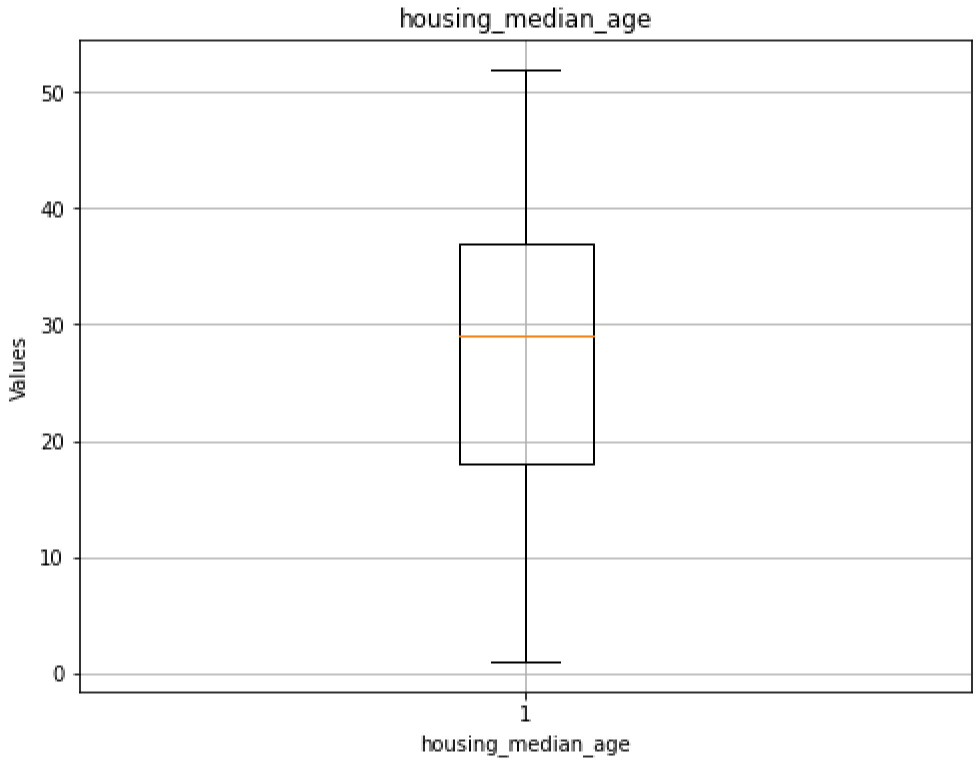

Normal Q-Q plot
−112
−114
−116
−118
−120
−122
−124
−126
longitude
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles

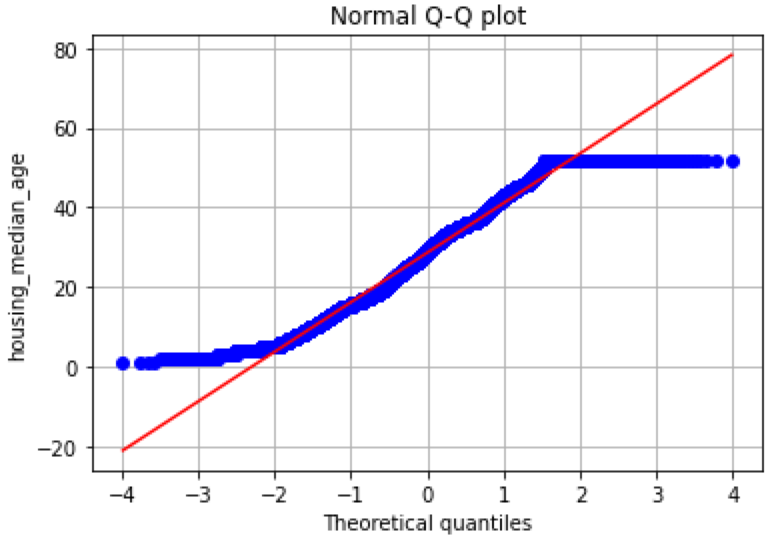

longitude
500000
400000
300000
200000
100000
0
median_house_value
−124
−122
−120
−118
−116
−114
longitude

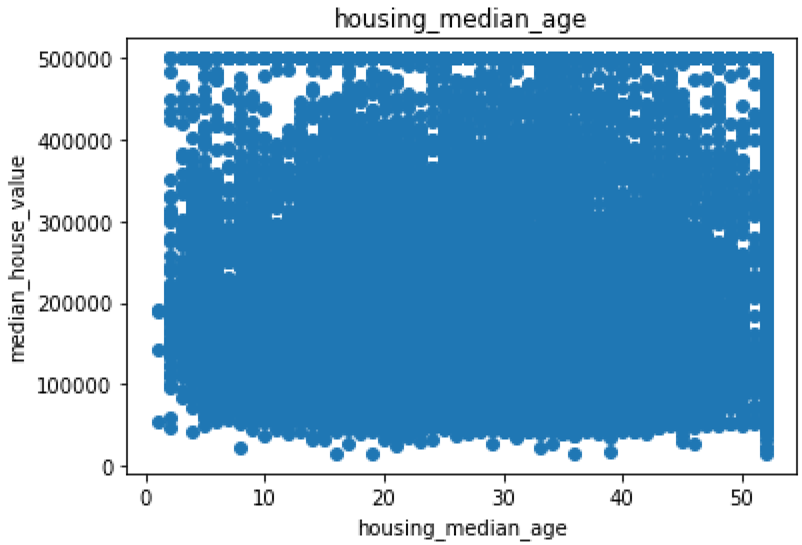

latitude
42
40
38
36
34
Values
1
latitude

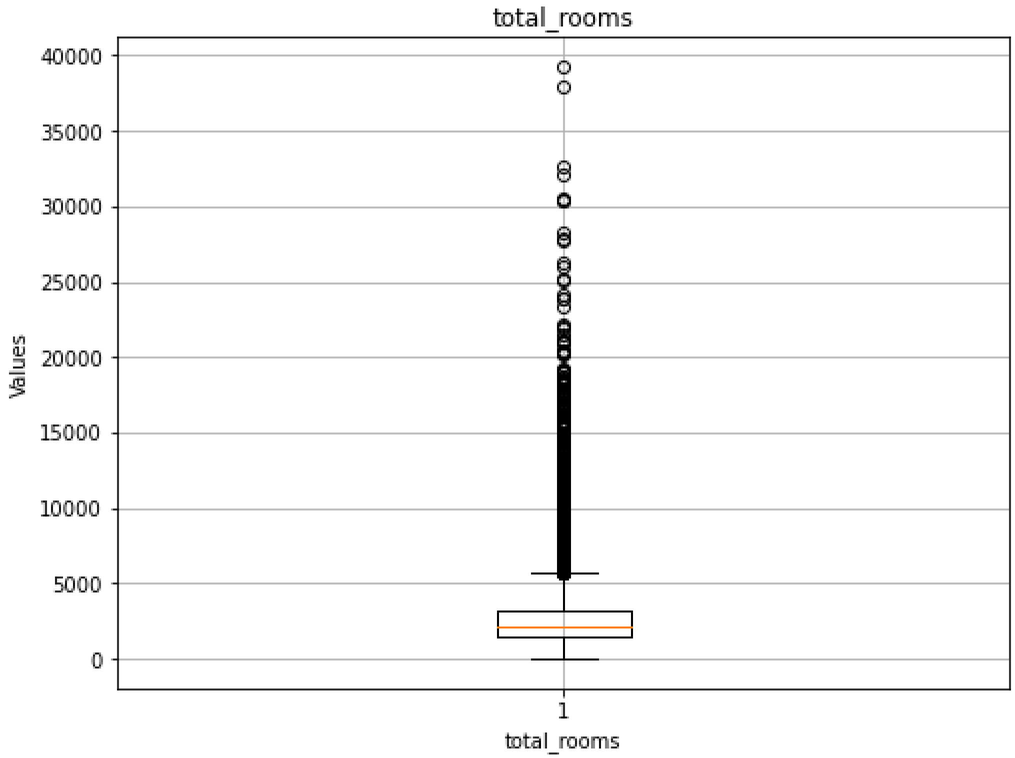

Normal Q-Q plot
44
42
40
38
36
34
32
30
28
latitude
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles

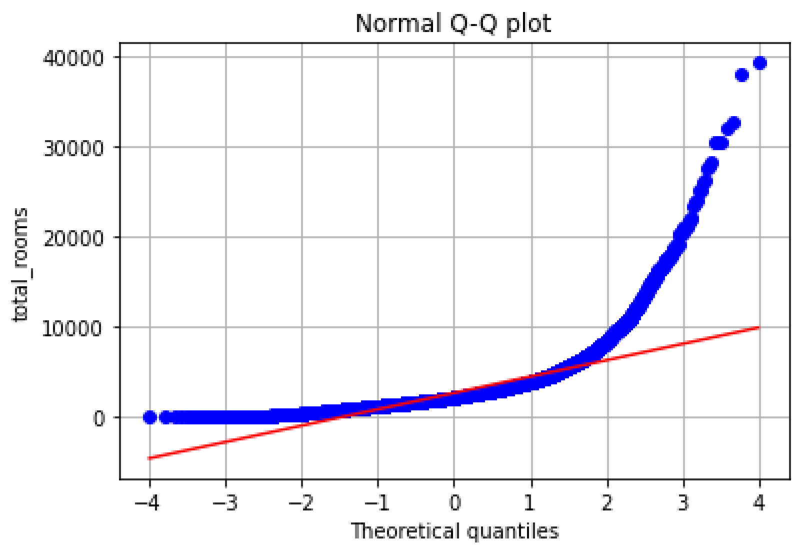

latitude
median_house_value
500000
400000
300000
200000
100000
0
34
36
38
40
42
latitude

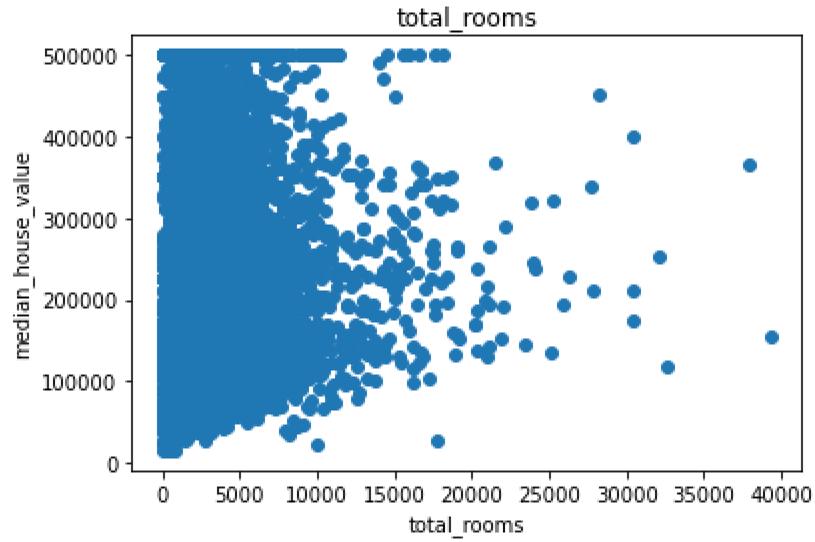

housing_median_age
Values
50
40
30
20
10
0
1
housing_median_age

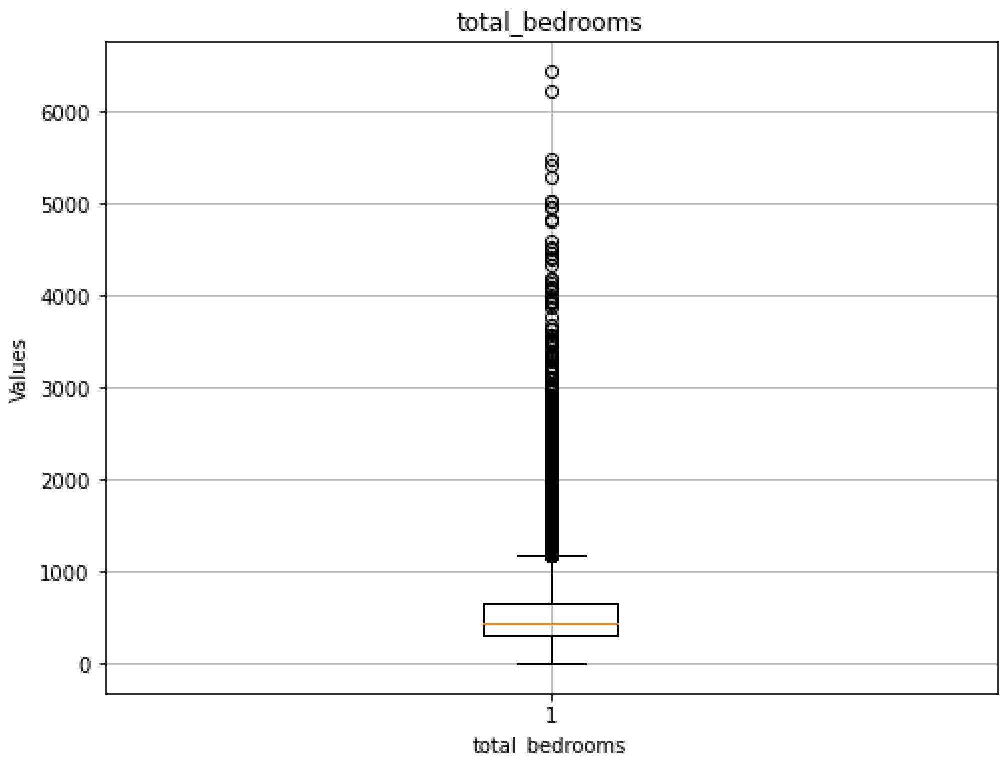

Normal Q-Q plot
housing_median_age
Theoretical quantiles
80
60
40
20
0
−20
−4
−3
−2
−1
0
1
2
3
4

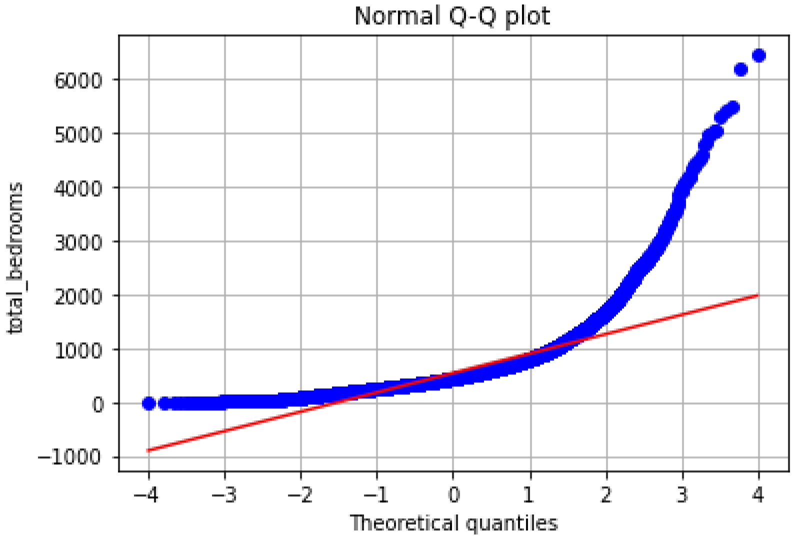

housing_median_age
median_house_value
housing_median_age
500000
400000
300000
200000
100000
0
0
10
20
30
40
50

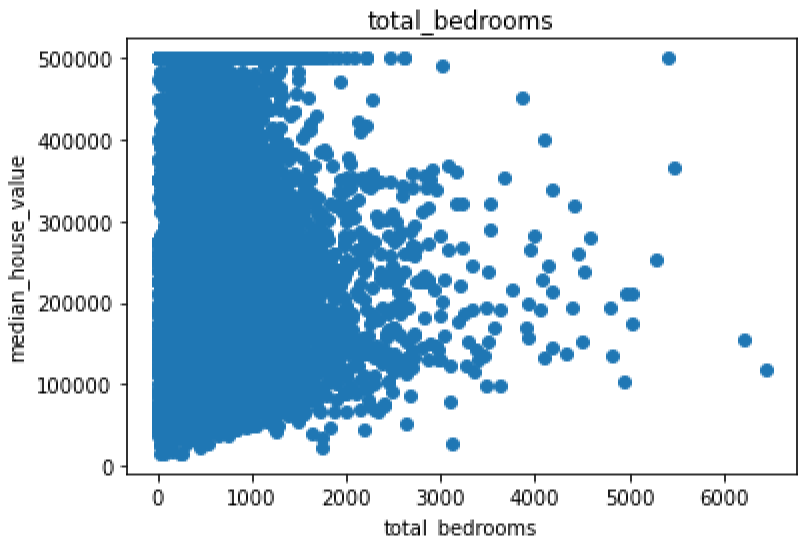

total_rooms
Values
40000
35000
30000
25000
20000
15000
10000
5000
0
1
total_rooms

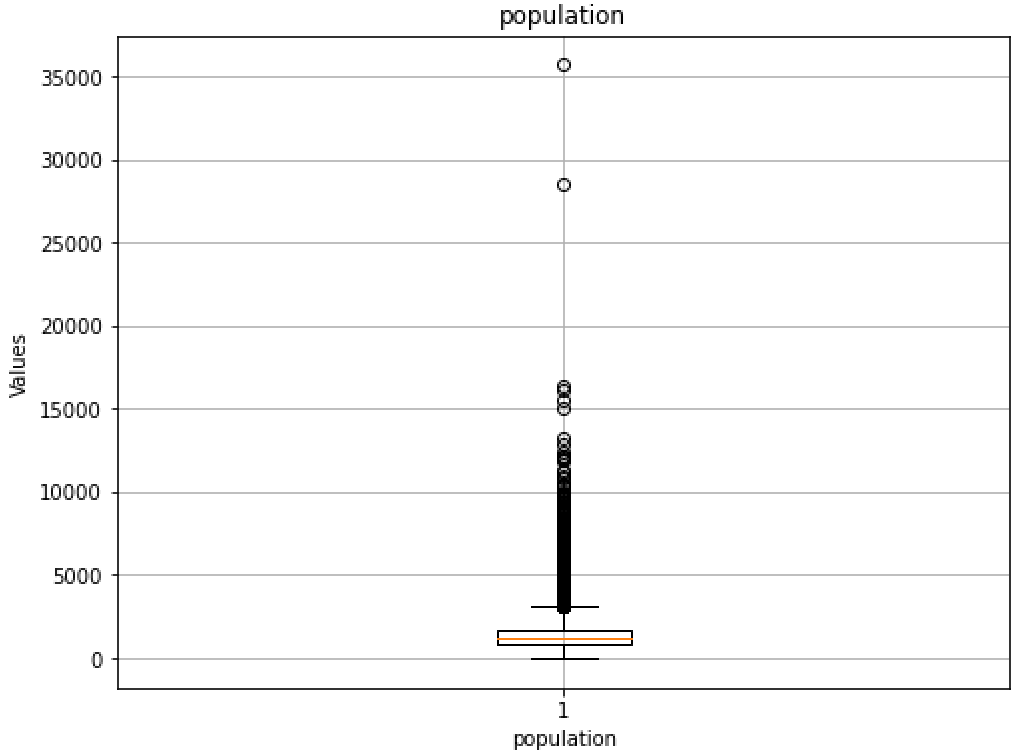

Normal Q-Q plot
total_rooms
40000
30000
20000
10000
0
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles

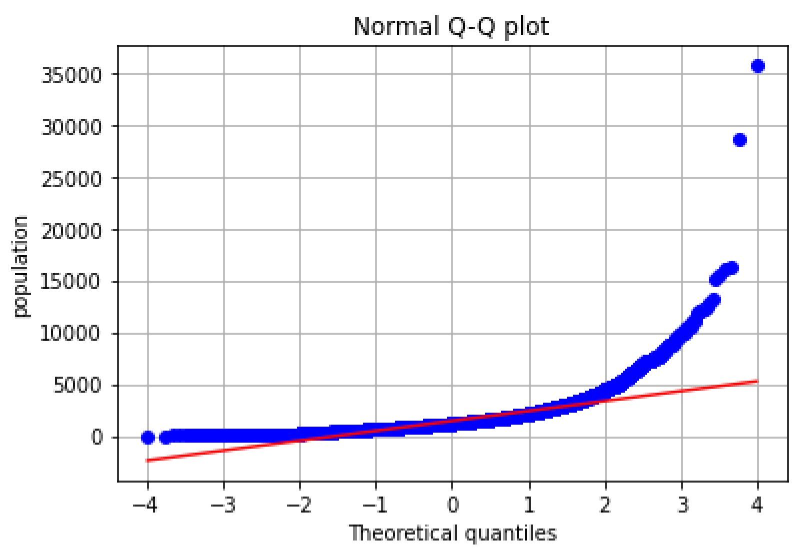

total_rooms
median_house_value
total_rooms
0
100000
200000
300000
400000
500000
0
5000
10000
15000
20000
25000
30000
35000
40000

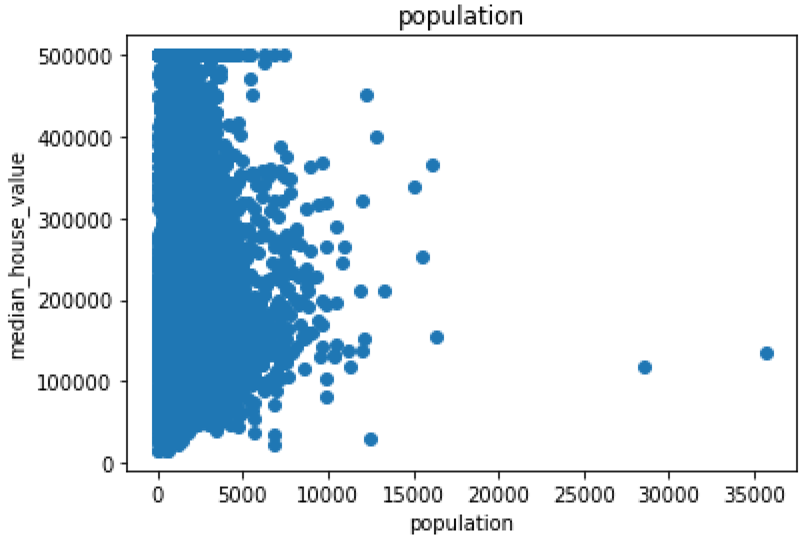

total_bedrooms
Values
total_bedrooms
0
1000
2000
3000
4000
5000
6000
1


Normal Q-Q plot
total_bedrooms
Theoretical quantiles
6000
5000
4000
3000
2000
1000
0
−1000
−4
−3
−2
−1
0
1
2
3
4

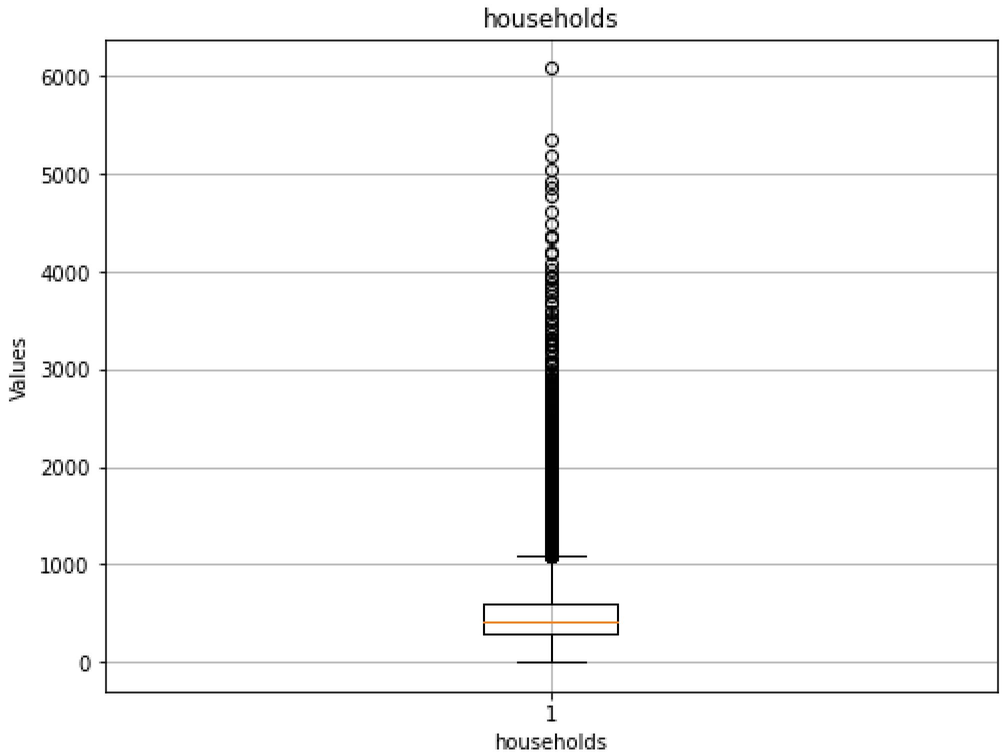

total_bedrooms
median_house_value
total_bedrooms
500000
400000
300000
200000
100000
0
0
1000
2000
3000
4000
5000
6000


population
Values
1
population

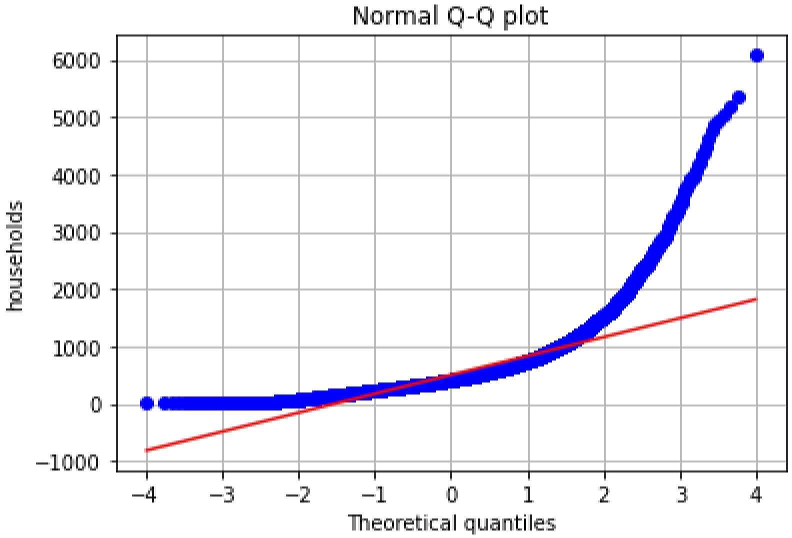

Normal Q-Q plot
population
Theoretical quantiles

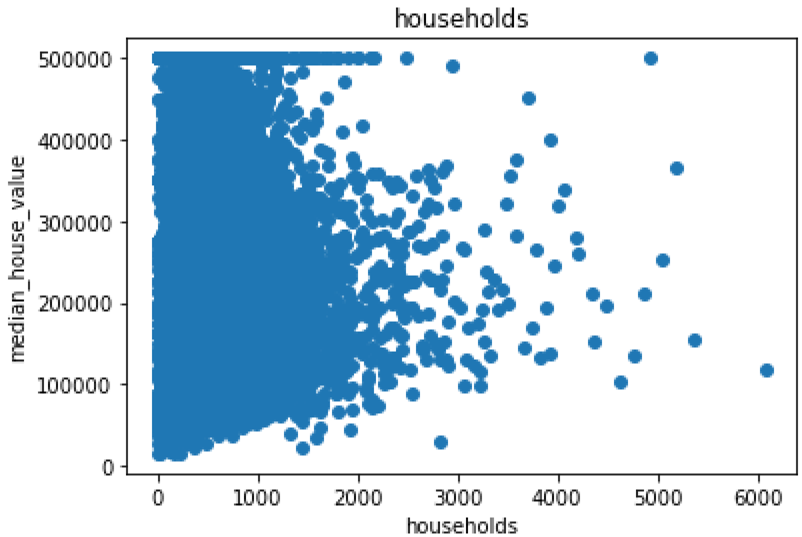


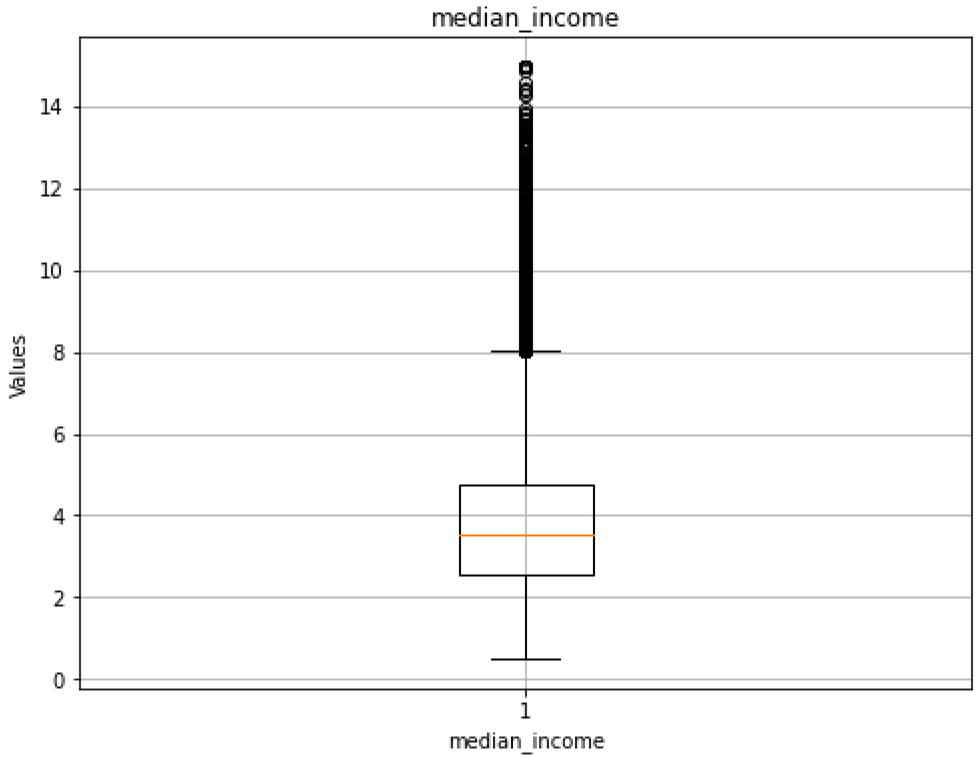
population
median_house_value
population



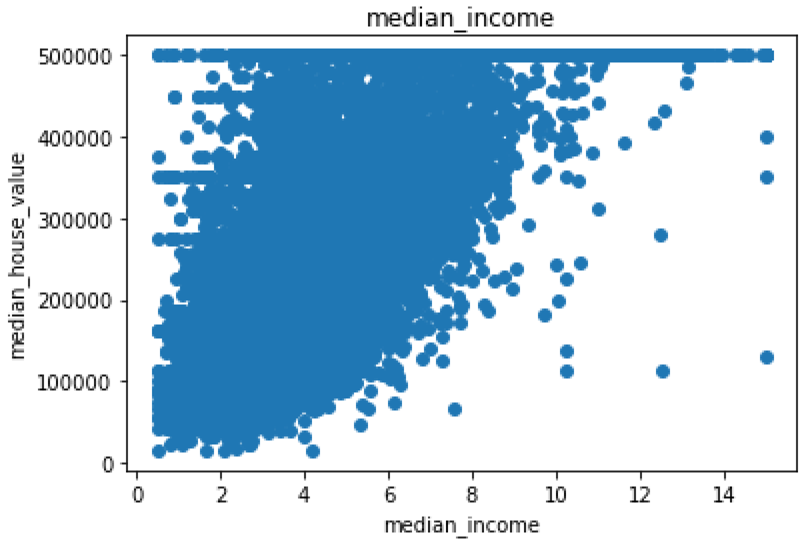
households
Values
households

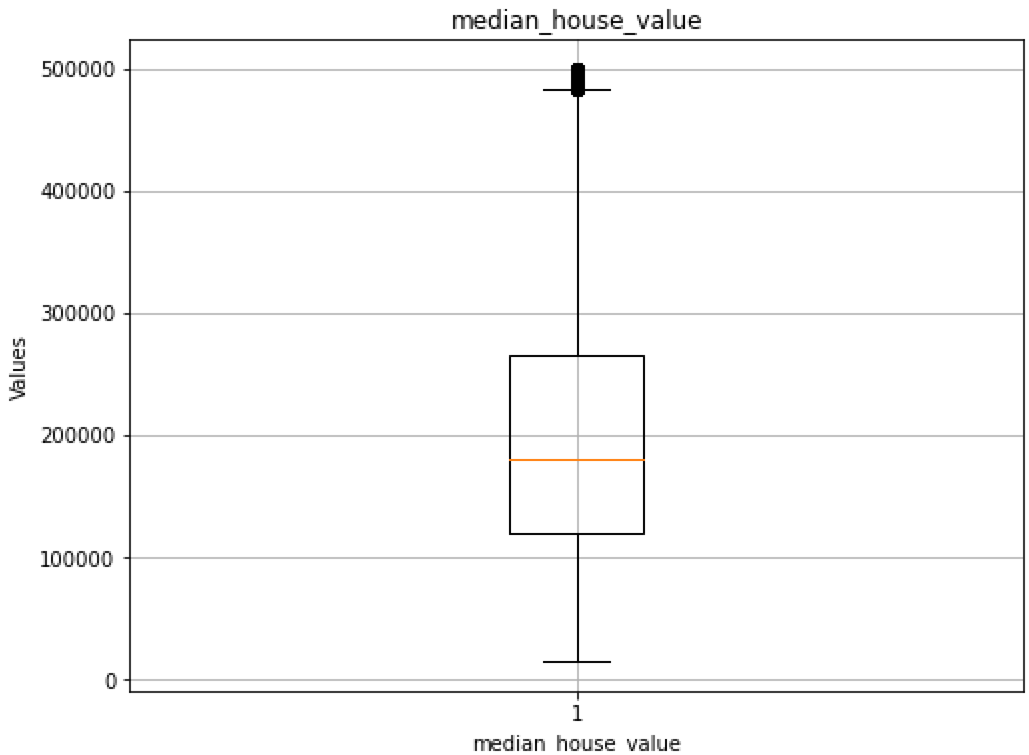

Normal Q-Q plot
households
Theoretical quantiles
6000
5000
4000
3000
2000
1000
0
−1000
−4
−3
−2
−1
0
1
2
3
4

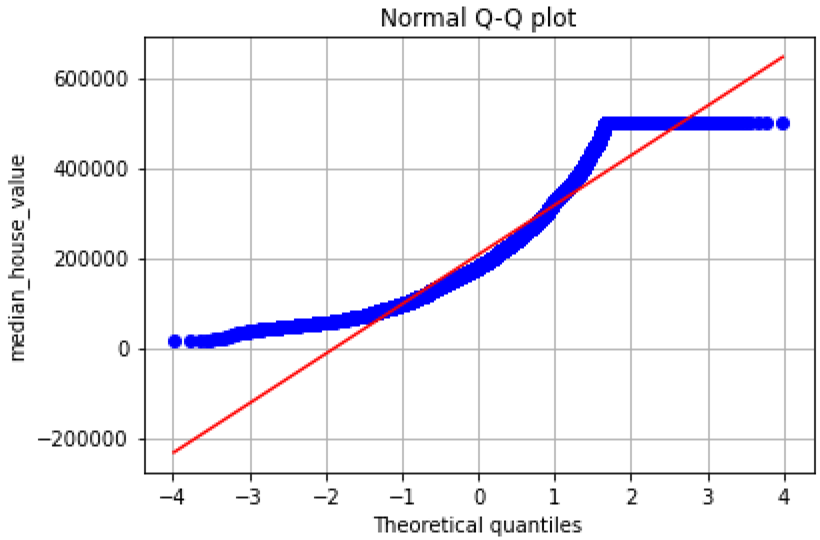

households
median_house_value
households
500000
400000
300000
200000
100000
0
0
1000
2000
3000
4000
5000
6000

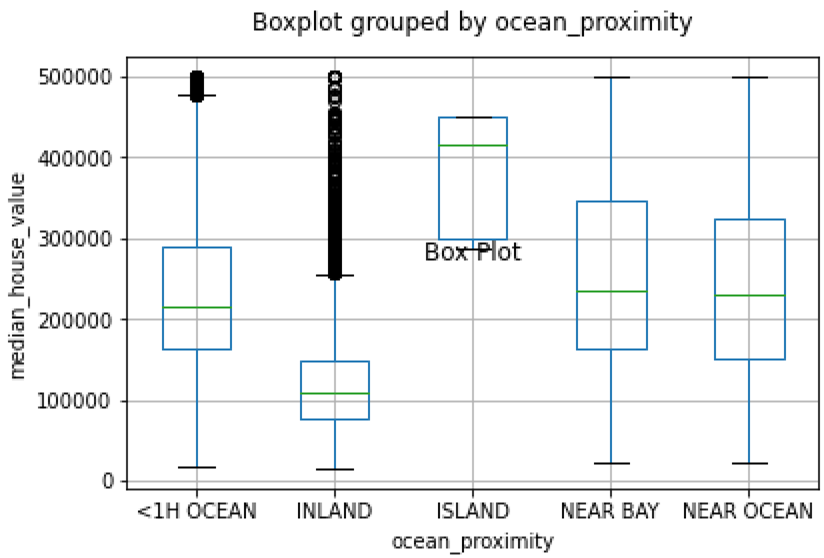

median_income
Values
14
12
10
8
6
4
2
0
1
median_income


Normal Q-Q plot
median_income
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−4
−3
−2
−1
0
1
2
3
4
Theoretical quantiles


median_income
median_house_value
500000
400000
300000
200000
100000
0
0
2
4
6
8
10
12
14
median_income


median_house_value
Values
500000
400000
300000
200000
100000
0
1
median_house_value



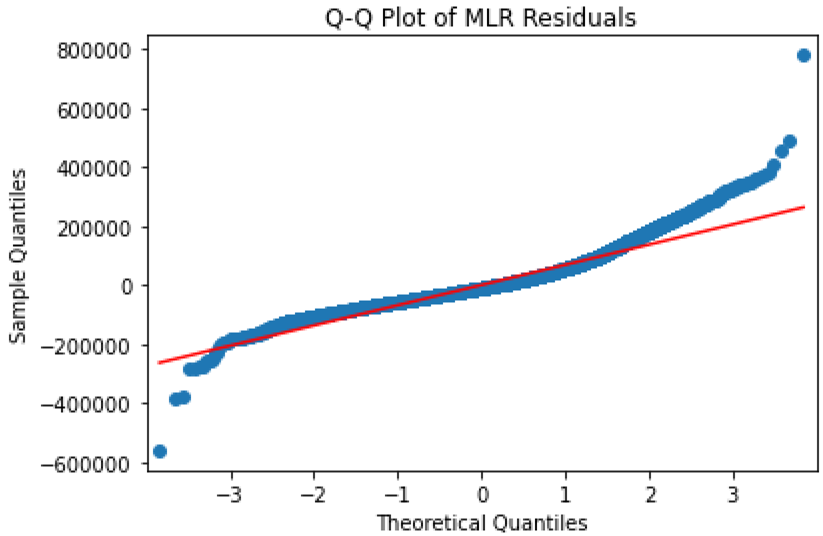


In [3]:

```
# Create an instance of the MLR model
# Fit the model on the training data
model = sm.OLS(y_train, x_train).fit()

print(model.summary())
# Get the predicted values
y_pred = model.predict(x_test)

# Calculate R-squared
r_squared = model.rsquared
print("R-squared:", r_squared)

# Calculate Mean Squared Error (MSE)
```

```
mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error:", mse)
# Calculate residuals
residuals = model.resid

# Create Q-Q plot
sm.qqplot(residuals, line='s')
plt.title('Q-Q Plot of MLR Residuals')
plt.show()
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:     median_house_value   R-squared:                       0.650
Model:                            OLS   Adj. R-squared:                  0.649
Method:                 Least Squares   F-statistic:                     2550.
Date:                Fri, 24 May 2024   Prob (F-statistic):               0.00
Time:                        20:40:18   Log-Likelihood:            -2.0727e+05
No. Observations:               16512   AIC:                         4.146e+05
Df Residuals:                   16499   BIC:                         4.147e+05
Df Model:                          12
Covariance Type:            nonrobust
==============================================================================================
                                 coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------------
const                      -2.276e+06   9.73e+04    -23.394      0.000   -2.47e+06   -2.08e+06
longitude                  -2.684e+04   1127.047    -23.813      0.000    -2.9e+04   -2.46e+04
latitude                   -2.547e+04   1111.486    -22.914      0.000   -2.76e+04   -2.33e+04
housing_median_age          1102.1851     48.605     22.676      0.000    1006.914    1197.456
total_rooms                   -6.0215      0.886     -6.796      0.000      -7.758      -4.285
total_bedrooms               102.7894      7.697     13.355      0.000      87.703     117.876
population                   -38.1729      1.188    -32.129      0.000     -40.502     -35.844
households                    48.2528      8.375      5.761      0.000      31.836      64.669
median_income               3.947e+04    375.091    105.238      0.000    3.87e+04    4.02e+04
ocean_proximity_INLAND     -3.979e+04   1933.681    -20.576      0.000   -4.36e+04    -3.6e+04
ocean_proximity_ISLAND      1.361e+05   3.43e+04      3.972      0.000    6.89e+04    2.03e+05
ocean_proximity_NEAR BAY   -5136.6422   2111.676     -2.432      0.015   -9275.756    -997.529
ocean_proximity_NEAR OCEAN  3431.1401   1751.612      1.959      0.050      -2.208    6864.488
==============================================================================
Omnibus:                     4119.707   Durbin-Watson:                   1.967
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            16516.873
Skew:                           1.189   Prob(JB):                         0.00
Kurtosis:                       7.284   Cond. No.                     7.21e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 7.21e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
R-squared: 0.6496648627123223
Mean Squared Error: 4936200740.484659
```



E.

Data Summary and Implications.

The implications of my data analyis are that I have failed to reject the null hypothesis. A MLR model was created on the research data set with an R-squared value of 0.649. This does not meet the accuracy criteria of 70% to reject the null hypothesis. This is consistent with the the Q-Q plot of the residuals which does not meet the normality assumption of MLR. The model was not reduced because there were no coefficients with a P value greater than .05. One limitation of my analysis is my lack of domain specific knowledge. This limits my ability to detect and remove outliers without skewing the data significantly. The course of action recommended based on the results of my analysis is to fail to reject the null hypothesis and assume that a MLR model cannot be constructed on the research data set. This model should not be used to predict house prices in California until the accuracy can be increased to 70% or greater.

One approach for future study of the data set would be to work with a domain expert to more accurately identify outliers without skewing the data set. This may help increase the accuracy of the model to 70% or greater.

A second approach for future study of the data set would be to apply logarithmic transformations to the skewed predictors to normalize them. This may increase the accuracy of the model to greater than 70%. Logarithmic transformation is a convenient means of transforming a highly skewed variable into a more normalized dataset. When modeling variables with non-linear relationships, the chances of producing errors may also be skewed negatively (DEV Community, 2019).

Citations

California Housing Prices. (n.d.). Www.kaggle.com. https://www.kaggle.com/datasets/camnugent/california-housing-prices

DEV Community. (2019, April 19). Logarithmic Transformation in Linear Regression Models: Why & When. The DEV Community; dev.to. https://dev.to/rokaandy/logarithmic-transformation-in-linear-regression-models-why-when-3a7c

Multiple Linear Regression. (n.d.). Www.jmp.com. https://www.jmp.com/en_us/statistics-knowledge-portal/what-is-multiple-regression.html

What Can Housing Markets Teach Us about Economics? (n.d.). NBER. https://www.nber.org/reporter/2016number4/what-can-housing-markets-teach-us-about-economics