Problem Set 1

This problem set is meant to help you familiarize yourself with Python and Pandas. This is meant to be a very gentle introduction -- if you find this problem set to be difficult, you might want to consider taking a more introductory course. Complete the exercises below, and export your completed notebook to a pdf. Submit both the notebook and the pdf file on bCourses.

Before You Start

For this problem set, you should download INF0251-PS1.ipynb (this file!) from bCourses or Github. Create a local copy of the notebook and rename it LASTNAME_FIRSTNAME-PS1.ipynb. Then, if you're using `jupyter` to work with your notebook, edit your renamed file directly in your browser by typing:

```
jupyter notebook <name_of_downloaded_file>
```

You can also upload the notebook to a cloud-based execution environment like Google Colab.

Make sure the following libraries load correctly (hit Ctrl-Enter).

In [1]:
```
#IPython is what you are using now to run the notebook
import IPython
print("IPython version:      %6.6s (need at least 1.0)" % IPython.__version__)

# Numpy is a library for working with Arrays
import numpy as np
print("Numpy version:        %6.6s (need at least 1.7.1)" % np.__version__)

# SciPy implements many different numerical algorithms
import scipy as sp
print("SciPy version:        %6.6s (need at least 0.12.0)" % sp.__version__)

# Pandas makes working with data tables easier
import pandas as pd
print("Pandas version:       %6.6s (need at least 0.11.0)" % pd.__version__)

# Module for plotting
import matplotlib
print("Maplotlib version:    %6.6s (need at least 1.2.1)" % matplotlib.__versi

# SciKit Learn implements several Machine Learning algorithms
import sklearn
print("Scikit-Learn version: %6.6s (need at least 0.13.1)" % sklearn.__version
```

```
IPython version:      8.27.0 (need at least 1.0)
Numpy version:         2.0.2 (need at least 1.7.1)
SciPy version:        1.14.1 (need at least 0.12.0)
Pandas version:        2.2.3 (need at least 0.11.0)
Maplotlib version:     3.9.2 (need at least 1.2.1)
Scikit-Learn version:  1.5.2 (need at least 0.13.1)
```

Working in a group?

List the names of other students with whom you worked on this problem set:

- *Person 1*
- *Person 2*
- ...

Introduction to the assignment

For this assignment, you will be using the California Housing Prices Dataset. Please read about the dataset carefully before continuing -- it is worth investing a few minutes up front otherwise you are likely to be hopelessly confused! We'll be coming back to this dataset repeatedly throughout the semester. Also, if you're new to analyzing data in Python, please make sure to read the relevant readings linked to on Canvas before beginning, otherwise you'll be stabbing in the dark.

Use the following commands to load the dataset:

In [2]:
```
from sklearn.datasets import fetch_california_housing
cal_data = fetch_california_housing()
```

The following commands will provide some basic information about the data:

In [3]:
```
print(cal_data.DESCR)
print(cal_data.keys())
print(cal_data.feature_names)
print(f"cal_data.data shape:   {cal_data.data.shape}")
print(f"caL_data.target shape: {cal_data.target.shape}")
```

```
.. _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).

A 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 surprisingly large values for block groups with few household
s
and many empty houses, such as vacation resorts.

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

.. rubric:: References

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

dict_keys(['data', 'target', 'frame', 'target_names', 'feature_names', 'DESC
R'])
['MedInc', 'HouseAge', 'AveRooms', 'AveBedrms', 'Population', 'AveOccup', 'Lat
itude', 'Longitude']
cal_data.data shape:   (20640, 8)
caL_data.target shape: (20640,)
```

The following commands will put together the features and target into a pandas dataframe:

In [4]:
```
print(cal_data.target)
```

```
[4.526 3.585 3.521 ... 0.923 0.847 0.894]
```

In [5]:
```
cal_df = pd.DataFrame(
    data=cal_data.data,
    columns=cal_data.feature_names
)
cal_df['MedHouseVal'] = cal_data['target']
```

Part 1: Descriptive analysis

1.1: Explore the data

Let's dig into the data a bit to see what we're dealing with. The first thing to do is to make sure you understand how the data is organized, what the data types are, whether there is any missing data, and so forth. Get your bearings on your own, then answer the following questions.

- 1.1.1: How many different variables are there in the dataset, and how many different observations?
- 1.1.2: What is the datatype of each variable?
- 1.1.3: Does the dataset contain any missing values?
- 1.1.4: How would you interpret the row index value?

In [6]:
```
print(f"- 1.1.1: Number of variables: {cal_df.shape[1]}; including the target
print(f"- 1.1.1: Number of observations: {cal_df.shape[0]}")
print(f"\n- 1.1.2:datattype:\n{cal_df.dtypes}\n")
print(f"- 1.1.3: Any missing value? \n{cal_df.isna().sum()}")
print(f"\n- 1.1.4: An example:\n{cal_df.loc[0]}\n")
```

```
- 1.1.1: Number of variables: 9; including the target variable
- 1.1.1: Number of observations: 20640

- 1.1.2:datattype:
MedInc          float64
HouseAge        float64
AveRooms        float64
AveBedrms       float64
Population      float64
AveOccup        float64
Latitude        float64
Longitude       float64
MedHouseVal     float64
dtype: object

- 1.1.3: Any missing value?
MedInc          0
HouseAge        0
AveRooms        0
AveBedrms       0
Population      0
AveOccup        0
Latitude        0
Longitude       0
MedHouseVal     0
dtype: int64

- 1.1.4: An example:
MedInc           8.325200
HouseAge        41.000000
AveRooms         6.984127
AveBedrms        1.023810
Population     322.000000
AveOccup         2.555556
Latitude        37.880000
Longitude     -122.230000
MedHouseVal      4.526000
Name: 0, dtype: float64
```

- 1.1.1: Number of variables: **9**, including the target variable and Number of observations: **20640**
- 1.1.2: **float64**
- 1.1.3: No, there's no missing value.
- 1.1.4: Row index value do not carry any specifric detail by it self; it is just used as a mean for data access and manupulation (like a unique identifier). In this case, it starts from 0 and ends with 20639. The example above shows the values for index 0.

1.2: Answer some basic questions

- 1.2.1: What is the average population per block group? What was California's total population in 1990 according to the Census?
- 1.2.2: What are the median house values in California's block groups with the lowest and highest populations?

- 1.2.3: Suggest 5 block groups that are likely to be vacation resorts. Do a quick Google search to validate your results.
- 1.2.4: How many census block groups are west of the city of Berkeley (lat: 37.871666, long: -122.272781)?
- 1.2.5: What fraction of block groups have an average number of household members greater or equal to 10?

In [7]:
```
print(f"- 1.2.1: Ave population: {cal_df['Population'].mean()} and total popul

print(f"- 1.2.2: For the lowest population: {cal_df.loc[cal_df['Population'].i

thrs_01 = cal_df['AveOccup'].quantile(0.8)
thrs_02 = cal_df['Population'].quantile(0.2)
vac_resorts = cal_df###[(cal_df['AveOccup'] > thrs_01)]# & (cal_df['Population
thrs_03 = cal_df['AveRooms'].quantile(0.8)
vac_resorts = vac_resorts[vac_resorts['AveRooms'] > thrs_03]
thrs_04 = cal_df['HouseAge'].quantile(0.2)
vac_resorts = vac_resorts[vac_resorts['HouseAge'] < thrs_04]
thrs_05 = cal_df['MedHouseVal'].quantile(0.8)
vac_resorts = vac_resorts[vac_resorts['MedHouseVal'] > thrs_05]
print(f"\n- 1.2.3:\n {vac_resorts.head()}")

west_ber = cal_df[cal_df['Longitude'] < -122.272781].shape[0]
print(f"\n- 1.2.4: {west_ber}")

members = cal_df[cal_df['AveOccup'] >= 10].shape[0] / cal_df.shape[0]

print(f"- 1.2.5: {members}")
```

```
- 1.2.1: Ave population: 1425.4767441860465 and total population: 29421840.0
- 1.2.2: For the lowest population: 3.5 and highest population: 1.344

- 1.2.3:
      MedInc  HouseAge  AveRooms  AveBedrms  Population  AveOccup  Latitude  \
570  7.6110       5.0  6.855776   1.061442      7427.0  2.732524     37.72
576  7.2634      12.0  7.133034   1.018934      5781.0  2.880419     37.77
577  7.0568       5.0  7.023438   0.912109      1738.0  3.394531     37.73
706  6.2579      10.0  6.443323   1.029503      3827.0  2.971273     37.65
863  5.8151       6.0  6.402616   1.042151      2071.0  3.010174     37.58

     Longitude  MedHouseVal
570    -122.24        3.507
576    -122.06        3.416
577    -122.06        4.125
706    -122.04        3.155
863    -122.00        2.956

- 1.2.4: 2167
- 1.2.5: 0.0017926356589147287
```

- 1.2.1: Average population approximately was **1425** and total population was **29421840**.
- 1.2.2: For the lowest population: **3.5** and highest population: **1.344**

- 1.2.3: I decided to go for newer houses so that they might have fewer changes.

  first row: not a resort

second row: **RANCHO DE LOS AMIGOS**

third row: not a resort

forth row: not a resort

fifth row: not a resort

- 1.2.4: **2167**
- 1.2.5: **0.00179**

1.3: Summary statistics

Create a clean, organized table that shows just the following information (no more, no less) for each variable in the dataset. Note that your table should have K rows (one for each variable) and 7 columns, ordered as below:

- The name of the variable
- The number of observations with non-missing values
- The mean of the variable
- The standard deviation of the variable
- The minimum value of the variable
- The median of the variable
- The maximum value of the variable

In [8]:
```
table = pd.DataFrame({
    'No. of non-missing values': cal_df.notnull().sum(),
    'Mean': cal_df.mean(),
    'Std dev.': cal_df.std(),
    'Min': cal_df.min(),
    'Median': cal_df.median(),
    'Max': cal_df.max()
                    })
table.index.name = 'Varriable name'
# table = table.reset_index()
table#.round(2)
```

Out[8]:

Varriable name	No. of non-missing values	Mean	Std dev.	Min	Median	Max
MedInc	20640	3.870671	1.899822	0.499900	3.534800	15.000100
HouseAge	20640	28.639486	12.585558	1.000000	29.000000	52.000000
AveRooms	20640	5.429000	2.474173	0.846154	5.229129	141.909091
AveBedrms	20640	1.096675	0.473911	0.333333	1.048780	34.066667
Population	20640	1425.476744	1132.462122	3.000000	1166.000000	35682.000000
AveOccup	20640	3.070655	10.386050	0.692308	2.818116	1243.333333
Latitude	20640	35.631861	2.135952	32.540000	34.260000	41.950000
Longitude	20640	-119.569704	2.003532	-124.350000	-118.490000	-114.310000
MedHouseVal	20640	2.068558	1.153956	0.149990	1.797000	5.000010

1.4 Simple Linear Regression

Estimate a linear regression of the median house value (the dependent variable) on the population (the independent variable), with no other control variables. Interpret the coefficients and standard errors. Based on this analysis, can you conclude anything about the causal effect of decreasing the population on the median housing value?

In [9]:
```
import statsmodels.api as sm

y = cal_df['MedHouseVal']
x = cal_df['Population']
x = sm.add_constant(x)
model = sm.OLS(y, x).fit()

print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:            MedHouseVal   R-squared:                       0.001
Model:                            OLS   Adj. R-squared:                  0.001
Method:                 Least Squares   F-statistic:                     12.55
Date:                Sun, 26 Jan 2025   Prob (F-statistic):           0.000398
Time:                        19:58:00   Log-Likelihood:                -32236.
No. Observations:               20640   AIC:                         6.448e+04
Df Residuals:                   20638   BIC:                         6.449e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          2.1044      0.013    163.012      0.000       2.079       2.130
Population -2.512e-05   7.09e-06     -3.542      0.000    -3.9e-05   -1.12e-05
==============================================================================
Omnibus:                     2387.069   Durbin-Watson:                   0.308
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             3301.867
Skew:                           0.967   Prob(JB):                         0.00
Kurtosis:                       3.311   Cond. No.                     2.93e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correct
ly specified.
[2] The condition number is large, 2.93e+03. This might indicate that there ar
e
strong multicollinearity or other numerical problems.
```

Based on the results, intercept is 2.1044 and population (slope) coefficient is -2.512e-05 with a p-value of 0 which is less than 0.05 (statitstically significant). The std error of the population is 7.09e-06. This small std error relative to the coefficiants can suggest that the prediction is precise. Also t-statistical coefficient is -3.542. We know that statistical significance does not imply a strong or meaningful effect in practical terms. R-sqrd is 0 >> this suggest that population alone is not a good predictor of median housing vallues.

Thus, we cannot conclude causality from this study. From this OLS regression analysis we cannot prove causation but we can say that the statistically significant negatiove relationship between populatioan and housing values suggest a correlation.

Part 2: Histograms and Scatterplots

2.1: Histogram of housing prices

Below you will find some very basic code to make a very basic histogram of median housing prices (the "target" variable) for your dataset. Your first task is to make this graph look pretty by doing the following:

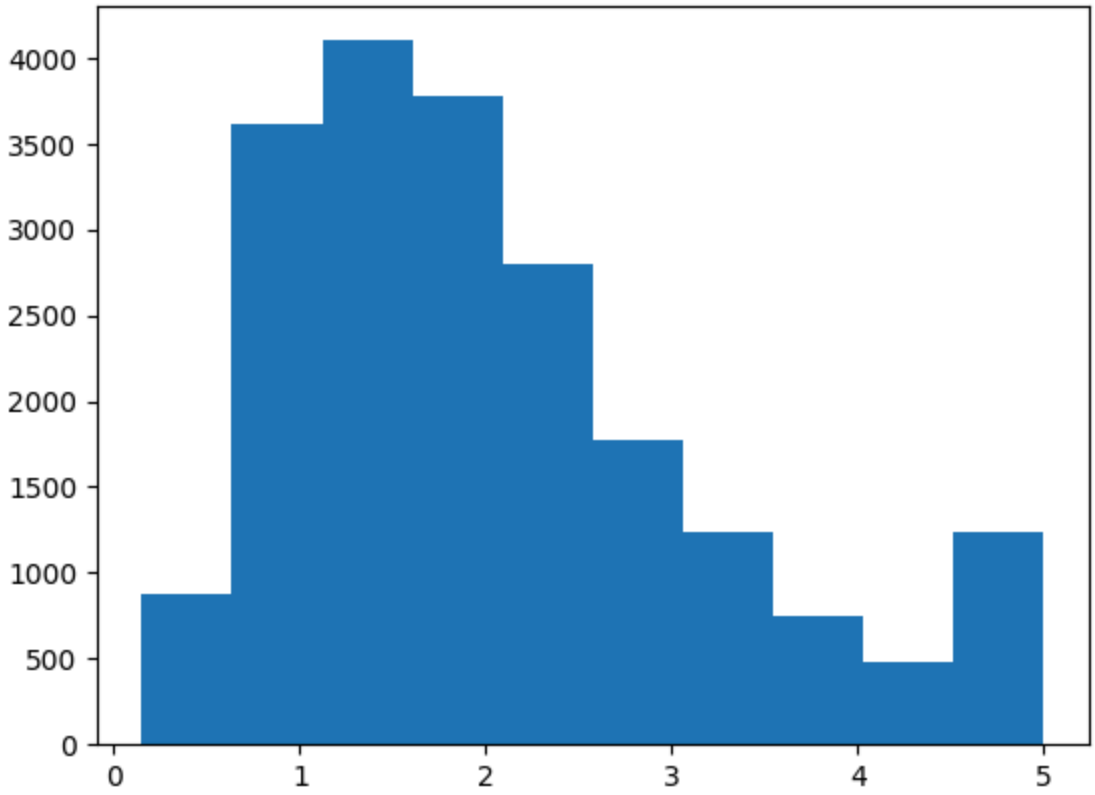
1. Add appropriate labels to the x and y axes, make sure to include units
2. Change the bin width on the histogram to be $5,000

3. Remove the axes on the top and right side of the plot window
4. Change the color of the bars to be green
5. Add an appropriate title

In [10]:
```
# prepare IPython to work with matplotlib and import the library to something
# %matplotlib inline
import matplotlib.pyplot as plt

# edit the code below to make the graph look good
plt.hist(cal_df['MedHouseVal'])
```

Out[10]:
```
(array([ 877., 3612., 4099., 3771., 2799., 1769., 1239.,  752.,  479.,
        1243.]),
 array([0.14999 , 0.634992, 1.119994, 1.604996, 2.089998, 2.575   ,
        3.060002, 3.545004, 4.030006, 4.515008, 5.00001 ]),
 <BarContainer object of 10 artists>)
```



2.2: Histogram of average occupancy

Now use your histograming skills to create a fine looking histogram of the average number of household members ("AveOccup"). In the same figure, plot the mean and median values of this variable. (Hint: applying a very common transformation to the data might make things easier).

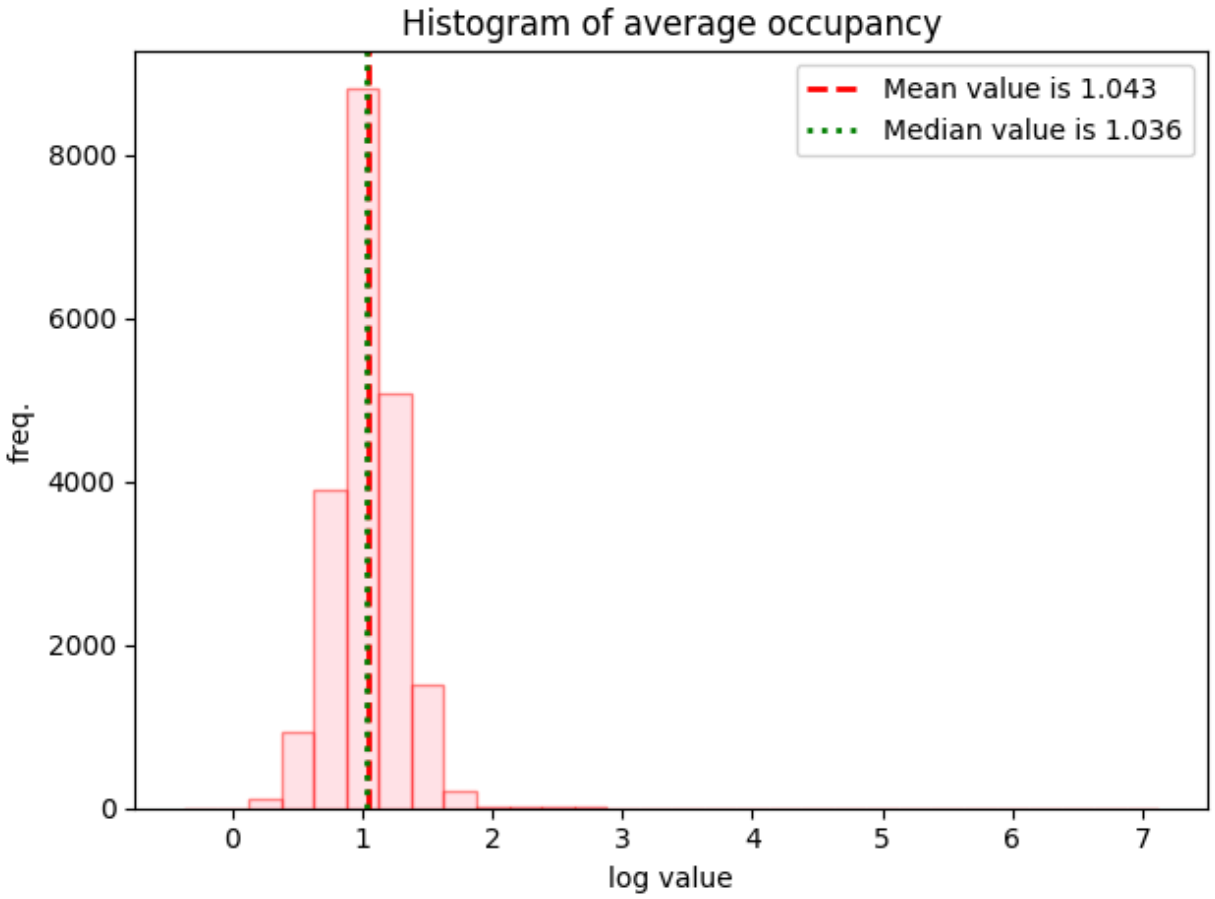
In [11]:
```
log_value = np.log(cal_df['AveOccup'])
mean_value = log_value.mean()
median_value = log_value.median()

plt.hist(log_value, bins=30, color='pink', edgecolor='red', alpha=0.45)

plt.axvline(mean_value, color='red', linestyle='dashed', linewidth=2, label=f'
plt.axvline(median_value, color='green', linestyle='dotted', linewidth=2, labe
```

```
plt.title('Histogram of average occupancy')
plt.xlabel('log value')
plt.ylabel('freq.')
plt.legend()

plt.tight_layout()
```

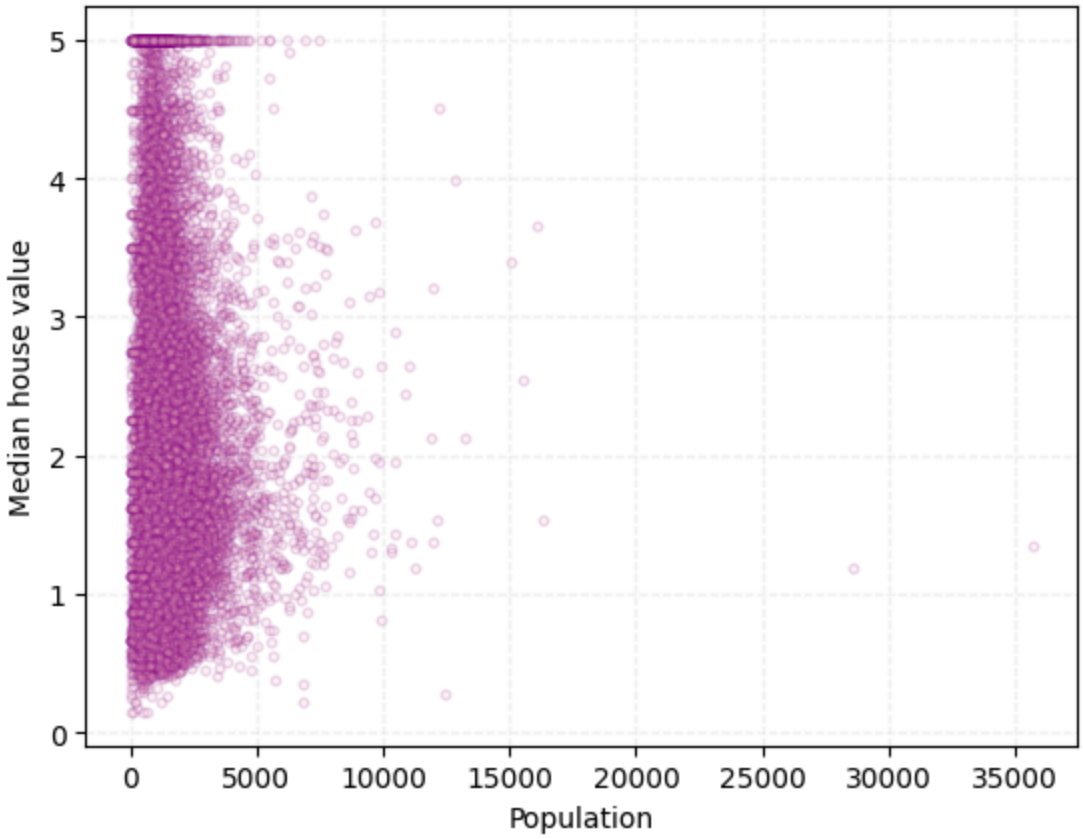


2.3: Scatter plot of housing prices and population

Use matplotlib to create a scatter plot that shows the relationship between the block group median house value (y-axis) and the block group population (x-axis). Properly label your axes, and make sure that your graphic looks polished and professional.

In [12]:
```
plt.scatter(cal_df['Population'], cal_df['MedHouseVal'], color='pink', alpha=0

plt.xlabel('Population')
plt.ylabel('Median house value')
# plt.tight_layout()
plt.grid(True, linestyle='--', alpha=0.15)
```



2.4: Interpret

What do you observe in the above scatter plot? Does there appear to be a relationship between media house value and population in California? Calculate the correlation between these two variables. Do you think this relationship is causal, or just a correlation? Justify your position and compare to your answer in section 1.4.

In [13]:
```
correlation = cal_df['Population'].corr(cal_df['MedHouseVal'])
print(f"Pearson correlation: {correlation:.4f}")
```

```
Pearson correlation: -0.0246
```

With a Pearson correlation coefficient clode to zero we can conclude that there is almost no linear relationship between the median housing values and population. This very weak negative correlation confirms that any association between population and housing value is negligible. We see in the scatter plot, datapoints are mostly clustered toward the lower populations (lower than 5000), which aligns with the obtained correlation coefficient. The maximum median housing value is also capped at 5, which might be related to the upper limit in the dataset. in higher populations, we could see some scattered datapoints, but still no clear pattern.

Given correlation does not imply causation, the relationship is almost certainly not causal based on the evidence.

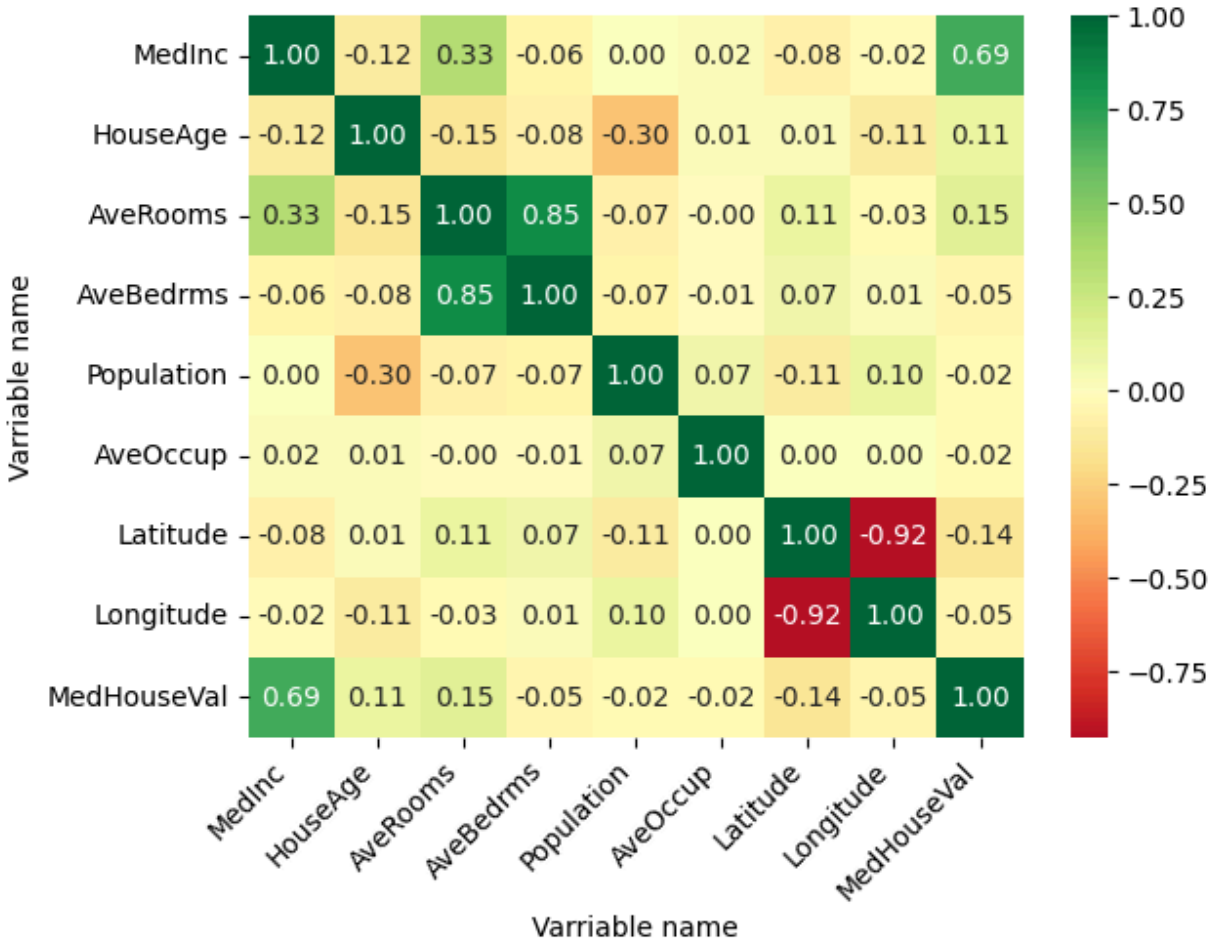
2.5 Correlation Matrix

Calculate the correlation of every pair of variables in the dataset. Create a *K* x *K* matrix where the value in the (*i*,*j*) cell is the correlation between the *i*th and *j*th variable. Show off your skills by coloring the cell so that large positive correlations appear green and large negative correlations appear red (use a gradient to make this pretty). What two variables appear to me most positively and negatively correlated? Explain these results.

In [14]:
```
import seaborn as sns

corr_matrix = cal_df.corr()
sns.heatmap(corr_matrix, annot=True, fmt=".2f", cmap="RdYlGn",center=0)

plt.xticks(rotation=45, ha="right")
plt.show()
```

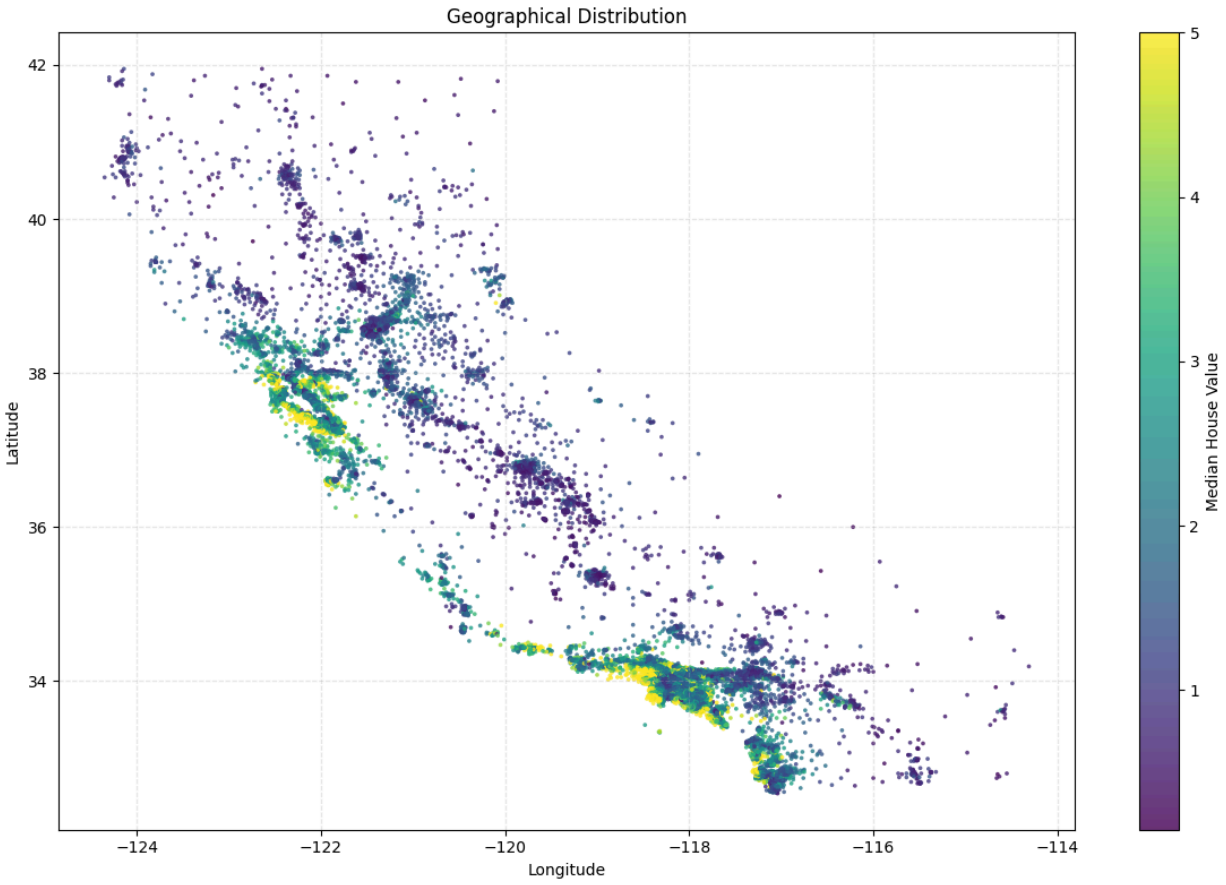


The most positively correlated value is seen between the average number of rooms and average number of bedrooms per household.Other values include the corelation of MedHouseVal and MedInc. The most negatively correlated value is seen between the longtitude and latitude, as we expected. Other values with high negative values isPopulation and median house age in block group.

2.6 Create your own (creative and effective) visualization

Use another type of graph or chart to illustrate an interesting pattern in the data. Be creative in your visualization, and make sure to produce a "publication-quality" graph. Points will be given for useful and creative graphs; points will be deducted for confusing issues like unlabeled axes. If you're new to data visualization, this guide is a good place to start.

In [15]:
```
#scatter plot of locations on California map
plt.figure(figsize=(12, 8))
plt.scatter(cal_df["Longitude"],cal_df["Latitude"],
    c=cal_df["MedHouseVal"], cmap="viridis", s=7, alpha=0.8, edgecolors="none"
plt.colorbar(label="Median House Value")
plt.title("Geographical Distribution")
plt.xlabel("Longitude")
plt.ylabel("Latitude")
plt.grid(linestyle="--", alpha=0.3)
plt.tight_layout()
plt.show()
```



In [ ]: