Loading Libraries and importing files

In [4]:
```
### Loading Libraries and importing the Housing.csv file
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

housing = pd.read_csv('/Users/siam/Desktop/Coding Assessment/Given_Files/ho
using.csv')
```

Task 1

In [5]:
```
# Viewing the file housing
print(housing)
```

```
       longitude  latitude  housing_median_age  total_rooms  total_bedroom
s  \
0        -122.23     37.88                41.0        880.0           129.
0
1        -122.22     37.86                21.0       7099.0          1106.
0
2        -122.24     37.85                52.0       1467.0           190.
0
3        -122.25     37.85                52.0       1274.0           235.
0
4        -122.25     37.85                52.0       1627.0           280.
0
...          ...       ...                 ...          ...
...
20635    -121.09     39.48                25.0       1665.0           374.
0
20636    -121.21     39.49                18.0        697.0           150.
0
20637    -121.22     39.43                17.0       2254.0           485.
0
20638    -121.32     39.43                18.0       1860.0           409.
0
20639    -121.24     39.37                16.0       2785.0           616.
0

       population  households  median_income  median_house_value  \
0           322.0       126.0         8.3252            452600.0
1          2401.0      1138.0         8.3014            358500.0
2           496.0       177.0         7.2574            352100.0
3           558.0       219.0         5.6431            341300.0
4           565.0       259.0         3.8462            342200.0
...           ...         ...            ...                 ...
20635       845.0       330.0         1.5603             78100.0
20636       356.0       114.0         2.5568             77100.0
20637      1007.0       433.0         1.7000             92300.0
20638       741.0       349.0         1.8672             84700.0
20639      1387.0       530.0         2.3886             89400.0

      ocean_proximity
0            NEAR BAY
1            NEAR BAY
2            NEAR BAY
3            NEAR BAY
4            NEAR BAY
...               ...
20635          INLAND
20636          INLAND
20637          INLAND
20638          INLAND
20639          INLAND

[20640 rows x 10 columns]
```

In [6]:
```
# Identifying mean, median, std, max and min of numerical columns
housing.describe().loc[['mean','50%','std','max','min']]

# Code utilized from: Topic 9 Lab Questions Task 6
```

Out[6]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	populat
mean	-119.569704	35.631861	28.639486	2635.763081	537.870553	1425.476
50%	-118.490000	34.260000	29.000000	2127.000000	435.000000	1166.000
std	2.003532	2.135952	12.585558	2181.615252	421.385070	1132.462
max	-114.310000	41.950000	52.000000	39320.000000	6445.000000	35682.000
min	-124.350000	32.540000	1.000000	2.000000	1.000000	3.000

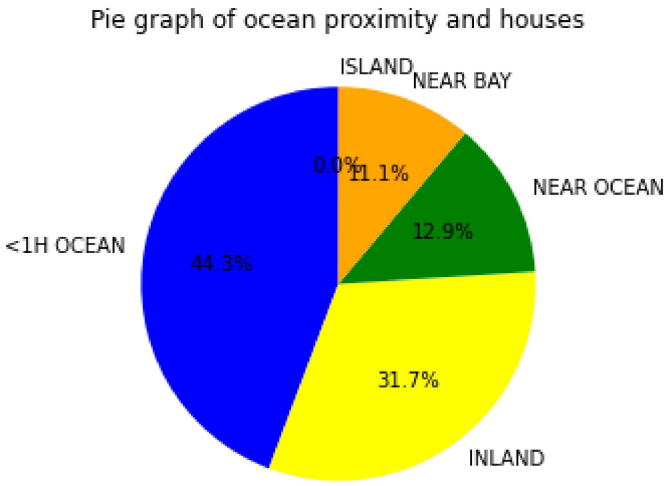
In [7]:
```
# Identifying number of rows and columns within the housing dataframe
def number_rows_columns():
    "Function identifies the total number of rows and columns in the datafr
ame housing"
    print("Total number of rows: ", housing.shape[0])
    print("Total number of columns: ", housing.shape[1])

number_rows_columns()
```

```
Total number of rows:  20640
Total number of columns:  10
```

In [8]:
```
# Creating frequency table to showcase unique values of ocean proximity and
the reoccurance of these unique values
frequency_OceanProximity = housing.ocean_proximity.value_counts()

frequency_OceanProximity
```

Out[8]:
```
<1H OCEAN     9136
INLAND        6551
NEAR OCEAN    2658
NEAR BAY      2290
ISLAND           5
Name: ocean_proximity, dtype: int64
```

In [9]:
```
# Creating pie chart to showcase breakdown of categories within the ocean proximity variable
labels = ['<1H OCEAN', 'INLAND', 'NEAR OCEAN', 'NEAR BAY', 'ISLAND']
colors =['blue', 'yellow', 'green', 'orange', 'red']
sizes = [9136, 6551, 2658, 2290, 5]
plt.pie(sizes,labels=labels, colors=colors, startangle=90, autopct='%1.1f%%')
plt.axis('equal')
plt.title('Pie graph of ocean proximity and houses', pad = 20)
plt.show()

# Code utilized from: http://www.learningaboutelectronics.com/Articles/How-to-create-a-pie-chart-in-matplotlib-with-Python.php
```



The pie graph above clearly indicates that most people live 1 hour away from the ocean with 44.3% of the chart. While the 31.7% of people live inland. This might be the result that most people would rather live near the ocean than the city.

Task 2

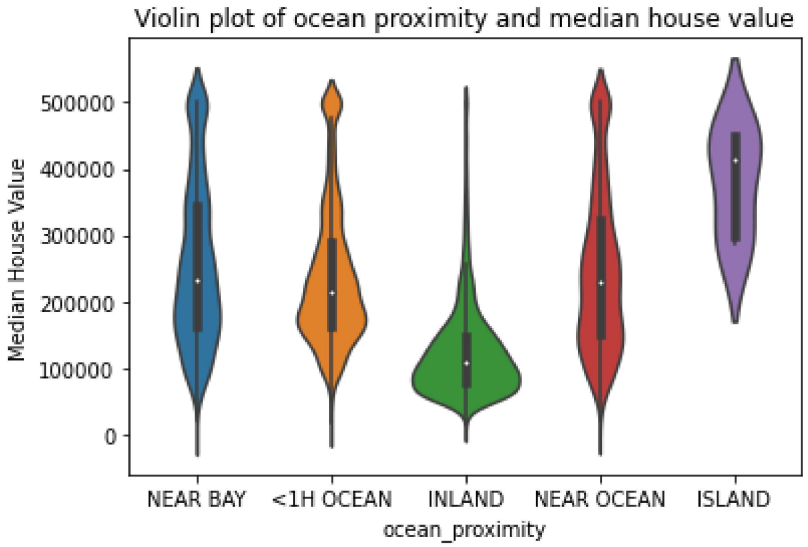
In [11]:
```
# Total number of missing values in total bedrooms
TB = housing['total_bedrooms']

def NA_total_bedrooms():
    "Function returns the total number of NAN values in total_bedrooms"
    print("Total number of missing values in total_bedrooms: ", TB.isnull().sum())

NA_total_bedrooms()
```

```
Total number of missing values in total_bedrooms:  207
```

In [12]:
```
# First 5 rows of missing values for total bedrooms
def first_5rows_NA():
    "Places all NAN values in total_bedrooms in a dataframe so the first 5 rows can be accessed"
    isNA = housing[housing['total_bedrooms'].isna()]
    print(isNA.head(5))

first_5rows_NA()
```

```
     longitude  latitude  housing_median_age  total_rooms  total_bedrooms
\
290    -122.16     37.77                47.0       1256.0             NaN
341    -122.17     37.75                38.0        992.0             NaN
538    -122.28     37.78                29.0       5154.0             NaN
563    -122.24     37.75                45.0        891.0             NaN
696    -122.10     37.69                41.0        746.0             NaN

     population  households  median_income  median_house_value ocean_proxi
mity
290       570.0       218.0         4.3750            161900.0        NEAR
BAY
341       732.0       259.0         1.6196             85100.0        NEAR
BAY
538      3741.0      1273.0         2.5762            173400.0        NEAR
BAY
563       384.0       146.0         4.9489            247100.0        NEAR
BAY
696       387.0       161.0         3.9063            178400.0        NEAR
BAY
```

In [13]:
```
# Creating a violin plot to showcase the median house value with respect to the ocean
sns.violinplot(x = "ocean_proximity", y = "median_house_value", data = housing)
plt.title("Violin plot of ocean proximity and median house value")
plt.ylabel("Median House Value")
plt.show()

# Code utilized from: Topic 10 lab questions task 4
```



The violon plot above showcases the relationship between the median house value of houses and their proximity towards to the ocean. It is evident that a small yet wealthy portion of houses live on islands, while most of the houses are inland.

Task 3

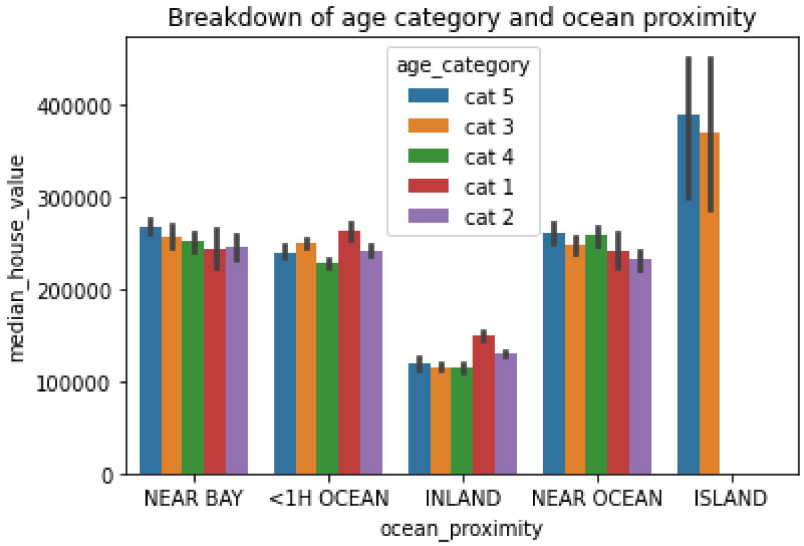
In [14]:
```
# Adding the column "age_category" to the housing dataframe based on 5 cate
gories

# Identifying conditions
conditions = [
    (housing['housing_median_age'] < 11 ),
    (housing['housing_median_age'] < 21),
    (housing['housing_median_age'] < 31),
    (housing['housing_median_age'] < 41),
    (housing['housing_median_age'] > 40),
    ]

# List of values for each condition
values = ['cat 1', 'cat 2', 'cat 3', 'cat 4', 'cat 5']

# Adding the new column to the housing dataframe
housing['age_category'] = np.select(conditions, values)
```

In [15]:
```
# Printing out the first 10 rows of median house age and age category
housing.iloc[0:10, [2,10]]
```

Out[15]:

	housing_median_age	age_category
0	41.0	cat 5
1	21.0	cat 3
2	52.0	cat 5
3	52.0	cat 5
4	52.0	cat 5
5	52.0	cat 5
6	52.0	cat 5
7	52.0	cat 5
8	42.0	cat 5
9	52.0	cat 5

In [16]:
```
# Creating a grouped bar chart for age_category and ocean_proximity
sns.barplot(x = "ocean_proximity", y = "median_house_value", hue = "age_cat
egory", data = housing)
plt.title('Breakdown of age category and ocean proximity')
plt.show()

# Utilized code from: Topic 9 lab questions task 7
```

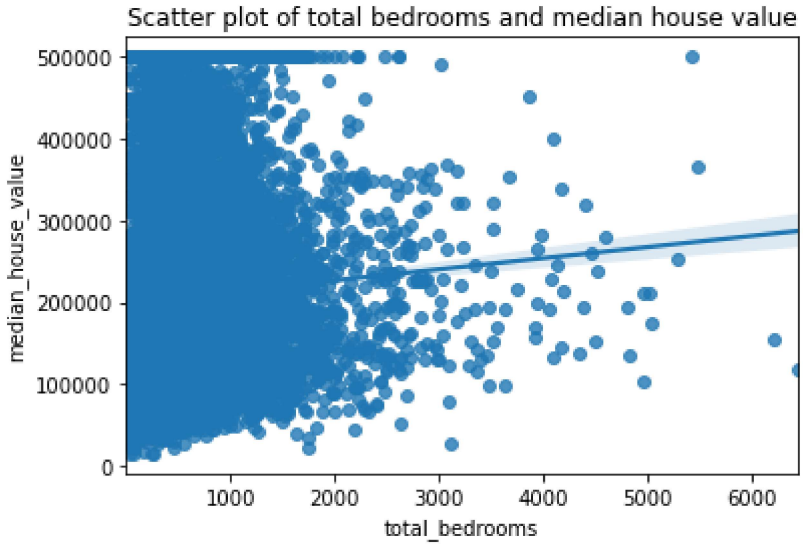


The graph above cleary identifies that there are no houses from age categories 1,2,4 in the island, while the least houses are inland.

Task 4

In [17]:
```
# Creating a scatter plot with a linear regression line
sns.regplot(x = "total_bedrooms", y = "median_house_value", data = housin
g);
plt.title('Scatter plot of total bedrooms and median house value')
plt.show

# Code utilized from: https://datavizpyr.com/how-to-make-scatter-plot-with-
regression-line-using-seaborn-in-python/
```

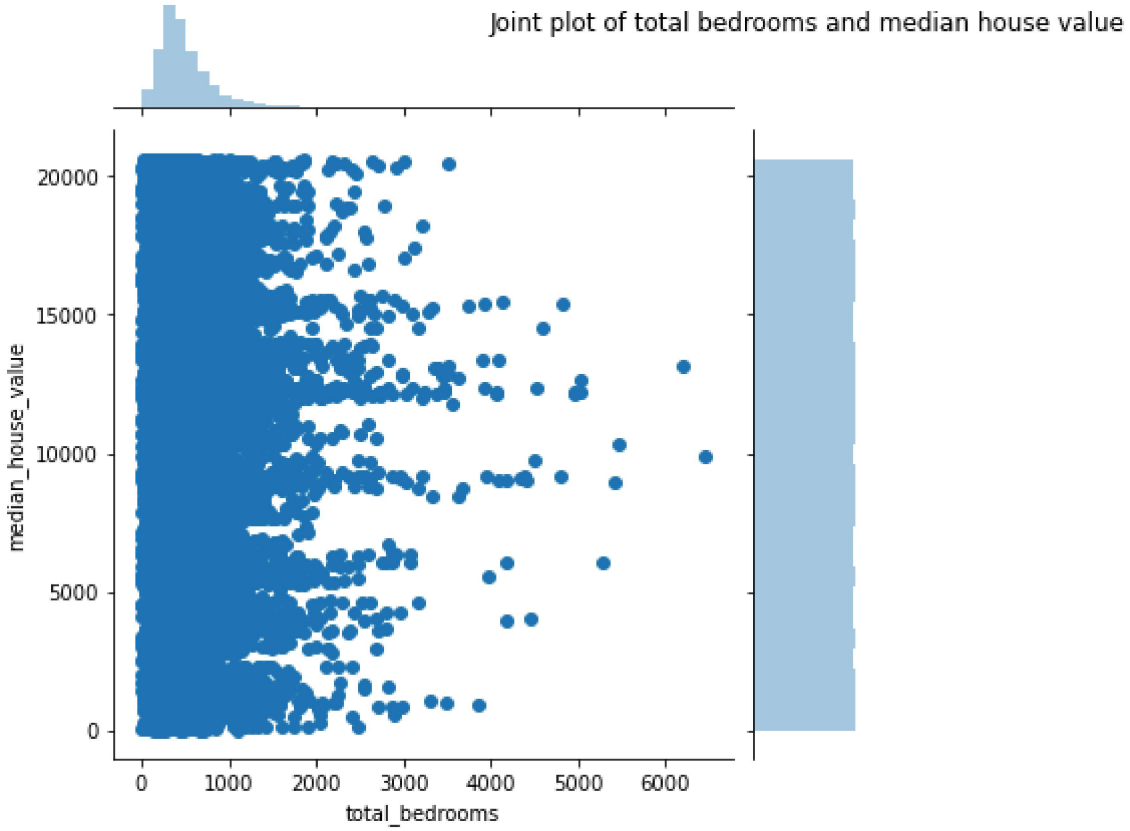
Out[17]:
```
<function matplotlib.pyplot.show(*args, **kw)>
```



The graph above showcases that the total bedrooms under 1000 are within all the house values. Furthermore, the graph identifies that not many houses have above 4000 bedrooms.

In [18]:
```
# Creating a joint plot
housing['median_house_value'] = housing.index

sns.jointplot(x = "total_bedrooms", y = "median_house_value", kind = "scatter", data = housing)
plt.title('Joint plot of total bedrooms and median house value', pad = 50)
plt.show

# Code utilized from: https://seaborn.pydata.org/generated/seaborn.jointplot.html
```

Out[18]:
```
<function matplotlib.pyplot.show(*args, **kw)>
```



The joint plot above is a represntation of the total bedrooms and the median house value. It is evident from the graph above that not many houses have more than 5000 bedrooms, regardless of the median house value. Furthermore, it is evident that most of the houses have less than 1000 rooms, as the histogram on the top indicates so.

In [19]:
```
# Filtering housing data for median income above $60,000 and median house age under 5 years
filtered_housing = housing.loc[(housing['median_income'] >= 6) &
                               (housing['housing_median_age'] < 5),
                               ['median_income', 'housing_median_age']]
```

In [20]:
```
# First 5 rows of the filtered_housing dataframe
filtered_housing.head(5)
```

Out[20]:

	median_income	housing_median_age
871	6.0824	3.0
981	6.8132	4.0
1362	6.0424	4.0
1560	6.3942	4.0
1566	15.0001	2.0

In [21]:
```
# Total number of census blocks in the filtered_housing dataframe
print('Total number of census blocks in the dataframe filtered_housing i
s:', filtered_housing.shape[0])
```

```
Total number of census blocks in the dataframe filtered_housing is: 84
```

In [22]:
```
# Random 500 sample from the housing data
random_housing = housing.sample(500)
```

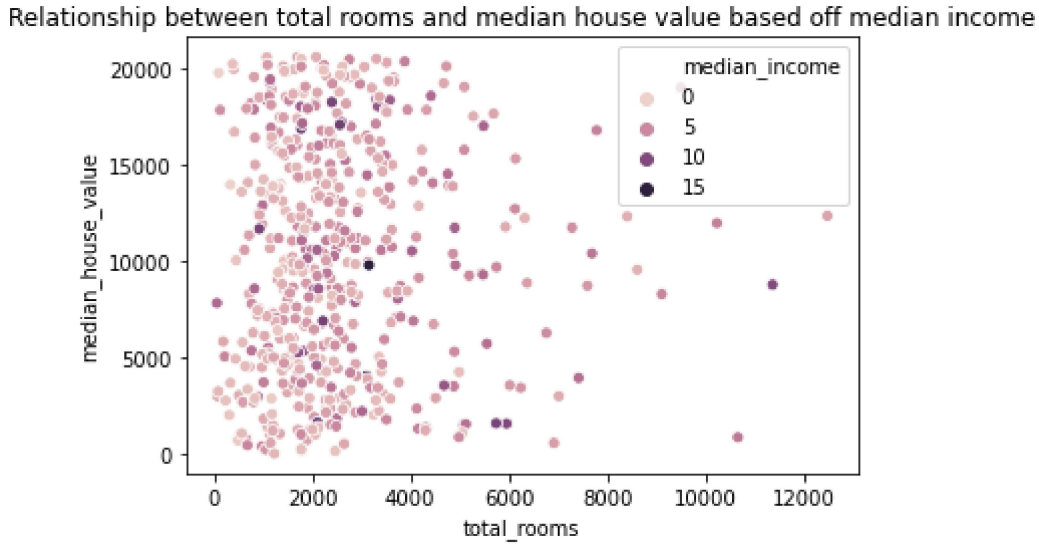
In [23]:
```
# First 10 rows of the random 500 sample
random_housing.head(10)
```

Out[23]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	hou
2923	-119.03	35.36	41.0	2551.0	594.0	1342.0	
5534	-118.43	33.96	38.0	1104.0	216.0	415.0	
15795	-122.42	37.77	52.0	4226.0	1315.0	2619.0	
13631	-117.33	34.07	32.0	2086.0	458.0	1355.0	
3008	-119.01	35.28	10.0	7011.0	1453.0	4163.0	
10019	-121.18	39.19	16.0	1528.0	351.0	729.0	
15321	-117.34	33.21	23.0	2062.0	376.0	1302.0	
6788	-118.15	34.09	52.0	2203.0	430.0	1238.0	
20344	-119.06	34.23	29.0	3511.0	632.0	2591.0	
19555	-121.01	37.61	5.0	3655.0	696.0	2316.0	

In [24]:
```
# Creating a scatter plot between 'total_rooms' and 'median_house_value' wi
th colored points based off of the 'median_income'
sns.scatterplot(x = 'total_rooms', y = 'median_house_value', hue = 'median_
income', data = random_housing);
plt.title('Relationship between total rooms and median house value based of
f median income')

# Code utilized from: Topic 9 lab question task 7
```

Out[24]:
```
Text(0.5, 1.0, 'Relationship between total rooms and median house value ba
sed off median income')
```



The scatter plot above is a representation of the total rooms and median house value based off of the median income for the randomly selected 500 census blocks. The graph above indicates that most houses have between 0 to 5000 rooms in their houses, while the most condensed area of the graph is based on houses with a median value between 300,000 and below. The graph also indicates that most of the people have an income of 50,000 or less.

Task 5

In [25]:
```
# Calculating number of people within each household
housing['number_people'] = housing['population'] / housing['households']
```

In [26]:
```
# Rounding the 'number_people' column
housing['number_people'] = housing['number_people'].round(decimals = 0)
print(housing['number_people'])
```

```
0        3.0
1        2.0
2        3.0
3        3.0
4        2.0
        ...
20635    3.0
20636    3.0
20637    2.0
20638    2.0
20639    3.0
Name: number_people, Length: 20640, dtype: float64
```

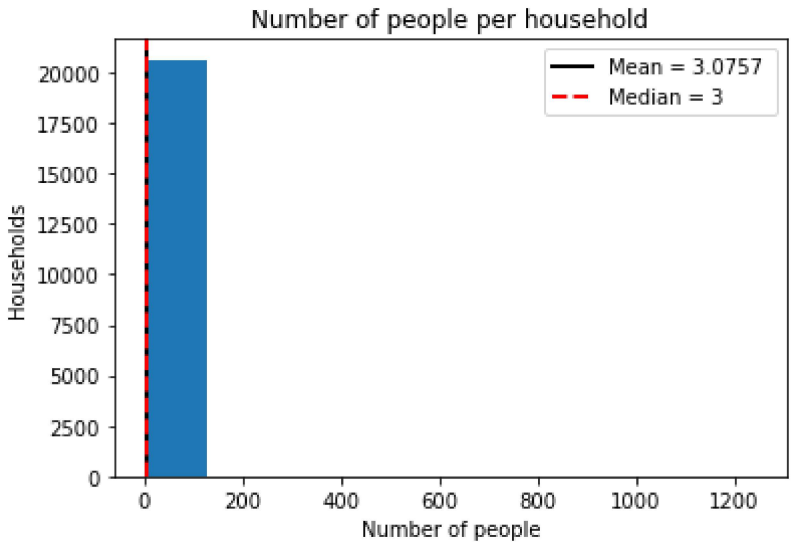
In [27]:
```
# Top 10 Largest number of people within a household
top10_people = housing.nlargest(10, 'number_people')
print(top10_people)
```

```
       longitude  latitude  housing_median_age  total_rooms  total_bedroom
s  \
19006    -121.98     38.32                45.0         19.0             5.
0
3364     -120.51     40.41                36.0         36.0             8.
0
16669    -120.70     35.32                46.0        118.0            17.
0
13034    -121.15     38.69                52.0        240.0            44.
0
9172     -118.59     34.47                 5.0        538.0            98.
0
12104    -117.33     33.97                 8.0        152.0            19.
0
16420    -121.29     37.89                26.0        161.0            27.
0
8874     -118.45     34.06                52.0        204.0            34.
0
13366    -117.63     33.94                36.0        447.0            95.
0
5986     -117.71     34.10                52.0        567.0           152.
0

       population  households  median_income  median_house_value  \
19006      7460.0         6.0        10.2264                19006
3364       4198.0         7.0         5.5179                 3364
16669      6532.0        13.0         4.2639                16669
13034      6675.0        29.0         6.1359                13034
9172       8733.0       105.0         4.2391                 9172
12104      1275.0        20.0         1.6250                12104
16420      1542.0        30.0         5.7485                16420
8874       1154.0        28.0         9.3370                 8874
13366      2886.0        85.0         4.2578                13366
5986       2688.0       126.0         1.8750                 5986

      ocean_proximity age_category  number_people
19006          INLAND        cat 5         1243.0
3364           INLAND        cat 4          600.0
16669      NEAR OCEAN        cat 5          502.0
13034          INLAND        cat 5          230.0
9172           INLAND        cat 1           83.0
12104          INLAND        cat 1           64.0
16420          INLAND        cat 3           51.0
8874         <1H OCEAN        cat 5           41.0
13366          INLAND        cat 4           34.0
5986           INLAND        cat 5           21.0
```

It is evident that the highest number of people living in a household is 1243 people, that live inland, while the second highest number of people living in a household is almost half of the top at 600 people in a household. Furthermore, it is evident that 8 out of 10 of these houses are located inland with one household near the ocean and the other 1 hour away from the ocean.

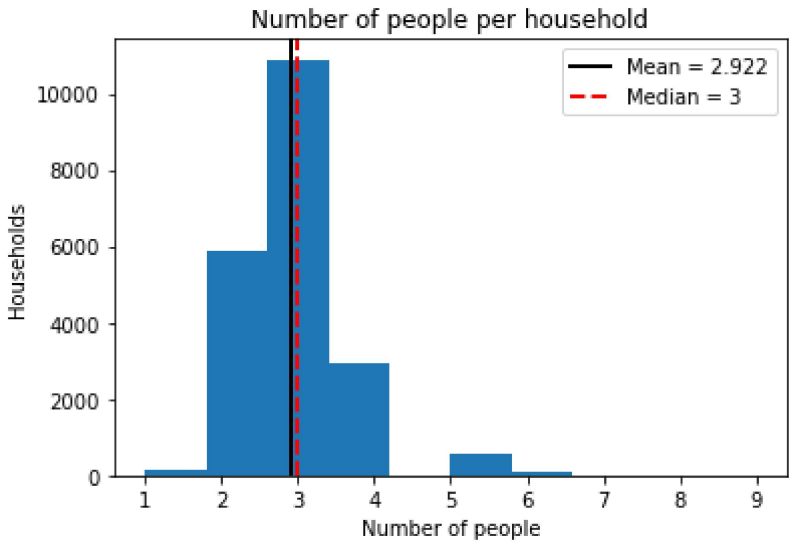
In [31]:
```
# Plotting a histogram of people per household with mean and median
number_people = housing['number_people']

number_people.mean()
number_people.median()

plt.hist(number_people)
plt.axvline(number_people.mean(), color='k', linewidth=2, label='Mean = 3.0
757 ')
plt.axvline(number_people.median(), color='r', linestyle='dashed', linewidt
h=2, label='Median = 3')
plt.title('Number of people per household')
plt.xlabel('Number of people')
plt.ylabel('Households')
plt.legend()

# Code utilized from: https://stackoverflow.com/questions/16180946/drawing-
average-line-in-histogram-matplotlib
```

Out[31]: <matplotlib.legend.Legend at 0x7fd243918100>



In [32]:
```
# Plotting a histogram for people per household under 10
less10 = housing[(housing['number_people'] < 10)]
less10_people = less10['number_people']

less10_people.mean()
less10_people.median()

plt.hist(less10_people)
plt.axvline(less10_people.mean(), color='k', linewidth=2, label='Mean = 2.9
22')
plt.axvline(less10_people.median(), color='r', linestyle='dashed', linewidt
h=2, label='Median = 3')
plt.title('Number of people per household')
plt.xlabel('Number of people')
plt.ylabel('Households')
plt.legend()
```

Out[32]:
```
<matplotlib.legend.Legend at 0x7fd242ca7bb0>
```



The plot with less than 10 people in a household is easier to read. This is due to the fact that the graph above, that includes everyone showcases households that have over 500 individuals which renders the graph as what seems like a single bar due to the x axis being so long.

In [34]:
```
# Identifying lowest 10 of median income
median_lowest10 = housing.nsmallest(10, 'median_income')
```

In [35]:
```
# Identifying highest 10 median income
median_highest10 = housing.nlargest(10, 'median_income')
```

Comparing top 10 people in a household with lowest 10 of median income

In [36]:
```
# Top 10 people in a household summary statistics
print(top10_people.describe().loc[['mean','50%','std','max','min']])
```

```
      longitude  latitude  housing_median_age  total_rooms  total_bedrooms
\
mean -119.53400  36.11700           35.800000   248.200000       49.900000
50%  -119.55000  34.89500           40.500000   182.500000       30.500000
std     1.75961   2.45047           17.618172   199.591694       48.754373
max  -117.33000  40.41000           52.000000   567.000000      152.000000
min  -121.98000  33.94000            5.000000    19.000000        5.000000

      population  households  median_income  median_house_value  number_pe
ople
mean  4314.30000   44.900000       5.322650        11799.500000      286.9
0000
50%   3542.00000   28.500000       4.890900        12569.000000       73.5
0000
std   2818.99001   43.646178       2.789473         4960.757945      394.9
2628
max   8733.00000  126.000000      10.226400        19006.000000     1243.0
0000
min   1154.00000    6.000000       1.625000         3364.000000       21.0
0000
```

In [37]:
```
# Top 10 people in a household house location
top10_people.ocean_proximity.value_counts()
```

Out[37]:
```
INLAND        8
NEAR OCEAN    1
<1H OCEAN     1
Name: ocean_proximity, dtype: int64
```

In [38]:
```
# Lowest 10 median income summary statistics
print(median_lowest10.describe().loc[['mean','50%','std','max','min']])
```

```
       longitude   latitude  housing_median_age  total_rooms  total_bedroo
ms  \
mean -119.351000  35.572000            34.40000   221.800000       69.9000
00
50%  -118.280000  34.420000            32.50000   208.000000       55.5000
00
std     2.068861   2.038212            15.63614   174.126901       68.9838
15
max  -116.900000  39.420000            52.00000   515.000000      229.0000
00
min  -122.890000  33.900000            10.00000    12.000000        4.0000
00

       population  households  median_income  median_house_value  \
mean   346.700000   50.800000   4.999000e-01         7033.200000
50%     91.000000   38.000000   4.999000e-01         5778.000000
std    825.756495   61.769824   1.170278e-16         5678.122828
max   2690.000000  217.000000   4.999000e-01        19523.000000
min     15.000000    6.000000   4.999000e-01           73.000000

      number_people
mean       3.600000
50%        2.500000
std        3.169297
max       12.000000
min        1.000000
```

In [39]:
```
#  Lowest 10 median income house location
median_lowest10.ocean_proximity.value_counts()
```

Out[39]:
```
INLAND       6
<1H OCEAN    3
NEAR BAY     1
Name: ocean_proximity, dtype: int64
```

It is evident that most of the highest 10 people live inland as 8 of the houses are located inland, with an average house value of 215,540.1 dollars and an average house age of 35.8 years, with a maximum number of 152 total bedrooms and a minimum median income of 16,250 dollars. Furthermore, the top 10 people have a maximum of 126 households and a maximum total rooms of 567.

While mostt of the lowest 10 median income people live inland with 6 houses, then 1 hour away from the ocean with 3 houses and 1 house near the bay. The average house value is 174,410 dollars and an average house age of 34.4 years, with a maximum number of 229 total bedrooms and a minimum median income of 49,999 dollars. Furthermore, the lowest 10 median income individuals have a maximum of 217 households and a maximum total rooms of 515.

This indicates that the top 10 number of people in a house and the lowest 10 people with a median income are living fairly similary lives. Although the top 10 people in a house have a higher average house value than those in the lowest 10 median income, the lowest 10 median income people have a higher average median income. Furthermore, the top 10 people in a house live with a maximum of 567 total rooms while the lowest 10 median income people live in a house with 515 total rooms.

Comparing top 10 people in a household with highest 10 of median income

In [40]:
```
# Highest 10 median income summary statistics
print(median_highest10.describe().loc[['mean','50%','std','max','min']])
```

```
       longitude   latitude  housing_median_age  total_rooms  total_bedroo
ms  \
mean -118.725000  34.446000            41.30000  1833.900000       228.300
00
50%  -118.355000  34.080000            49.00000  1601.500000       183.000
00
std     1.137514   1.157509            15.88885  1584.534313       214.046
75
max  -118.320000  37.740000            52.00000  5578.000000       753.000
00
min  -121.960000  34.060000             2.00000   200.000000        20.000
00

       population  households  median_income  median_house_value  \
mean   597.400000  214.000000        15.0001         4477.400000
50%    609.500000  178.500000        15.0001         4616.000000
std    447.500019  196.307582         0.0000         1073.616236
max   1567.000000  697.000000        15.0001         5248.000000
min     25.000000    9.000000        15.0001         1566.000000

      number_people
mean       3.000000
50%        3.000000
std        0.666667
max        4.000000
min        2.000000
```

In [41]:
```
#  Highest 10 median income house location
median_highest10.ocean_proximity.value_counts()
```

Out[41]:
```
<1H OCEAN    10
Name: ocean_proximity, dtype: int64
```

It is evident that most of the highest 10 people live inland as 8 of the houses are located inland, with an average house value of 215,540.1 dollars and an average house age of 35.8 years, with a maximum number of 152 total bedrooms and a minimum median income of 16,250 dollars. Furthermore, the top 10 people have a maximum of 126 households and a maximum total rooms of 567.

Meanwhile, the highest 10 median income indicates that all of the residents live an hour away from the ocean, with an average house value of 485,001 dollars and an average house age of 41.3 years. The highest 10 median income also have a maximum number of 753 total bedrooms and a minimum median income of 150,001 dollars. Furthermore, the highest 10 median income have a maximum of 687 households and a maximum total rooms of 5578.

Through comparing and contrasting between the top 10 people in a household and the highest 10 median income, it is evident that the highest 10 median income have a higher median income and a average house value almost double than the top 10 people in a household. Furthermore, the highest 10 median income have a extremely higher minimum median income than those in the top 10 people and have larger houses that can accomodate more people. This suggests that the people in the highest 10 median income can spend more on their houses due to their high income and live in more spacious houses.