In [30]:

```
#this is the practical no_2

import pandas as pd
data = {"Roll-num": [10,20,30,40,50,60,70],

"Age":[12,14,13,12,14,13,15],
"NAME":['John','Camili','Rheana','Joseph','Amanti','Alexa','Siri']}
```

In [2]:

```
data = pd.read_csv('C:/Users/vikas pawar/Downloads/dataset_Facebook.csv')
```

In [3]:

```
block = pd.DataFrame(data)
```

In [4]:

```
print("Original Data frame:\n")
```

```
Original Data frame:
```

In [9]:

```
print(block)
```

```
     Page total likes    Type  Category  Post Month  Post Weekday  Post Hour  \
0              139441   Photo         2          12             4          3
1              139441  Status         2          12             3         10
2              139441   Photo         3          12             3          3
3              139441   Photo         2          12             2         10
4              139441   Photo         2          12             2          3
..                ...     ...       ...         ...           ...        ...
495             85093   Photo         3           1             7          2
496             81370   Photo         2           1             5          8
497             81370   Photo         1           1             5          2
498             81370   Photo         3           1             4         11
499             81370   Photo         2           1             4          4

     Paid  Lifetime Post Total Reach  Lifetime Post Total Impressions  \
0     0.0                      2752                             5091
1     0.0                     10460                            19057
2     0.0                      2413                             4373
3     1.0                     50128                            87991
4     0.0                      7244                            13594
..    ...                       ...                              ...
495   0.0                      4684                             7536
496   0.0                      3480                             6229
497   0.0                      3778                             7216
498   0.0                      4156                             7564
499   NaN                      4188                             7292

     Lifetime Engaged Users  Lifetime Post Consumers  \
0                       178                      109
1                      1457                     1361
2                       177                      113
3                      2211                      790
4                       671                      410
..                      ...                      ...
495                     733                      708
496                     537                      508
497                     625                      572
498                     626                      574
499                     564                      524

     Lifetime Post Consumptions  \
0                           159
1                          1674
2                           154
3                          1119
4                           580
..                          ...
495                         985
496                         687
497                         795
498                         832
499                         743

     Lifetime Post Impressions by people who have liked your Page  \
0                                                    3078
1                                                   11710
2                                                    2812
3                                                   61027
4                                                    6228
..                                                    ...
495                                                  4750
496                                                  3961
497                                                  4742
498                                                  4534
499                                                  3861
```

```
     Lifetime Post reach by people who like your Page  \
0                                                 1640
1                                                 6112
2                                                 1503
3                                                32048
4                                                 3200
..                                                 ...
495                                               2876
496                                               2104
497                                               2388
498                                               2452
499                                               2200

     Lifetime People who have liked your Page and engaged with your post  \
0                                                  119
1                                                 1108
2                                                  132
3                                                 1386
4                                                  396
..                                                 ...
495                                                392
496                                                301
497                                                363
498                                                370
499                                                316

     comment    like  share  Total Interactions
0          4    79.0   17.0                100
1          5   130.0   29.0                164
2          0    66.0   14.0                 80
3         58  1572.0  147.0               1777
4         19   325.0   49.0                393
..       ...     ...    ...                ...
495        5    53.0   26.0                 84
496        0    53.0   22.0                 75
497        4    93.0   18.0                115
498        7    91.0   38.0                136
499        0    91.0   28.0                119

[500 rows x 19 columns]
```

In [5]:

```
print(block.loc[[0,1,3]])
```

```
   Page total likes    Type  Category  Post Month  Post Weekday  Post Hour  \
0            139441   Photo         2          12             4          3
1            139441  Status         2          12             3         10
3            139441   Photo         2          12             2         10

   Paid  Lifetime Post Total Reach  Lifetime Post Total Impressions  \
0   0.0                       2752                             5091
1   0.0                      10460                            19057
3   1.0                      50128                            87991

   Lifetime Engaged Users  Lifetime Post Consumers  \
0                     178                      109
1                    1457                     1361
3                    2211                      790

   Lifetime Post Consumptions  \
0                         159
1                        1674
3                        1119

   Lifetime Post Impressions by people who have liked your Page  \
0                                               3078
1                                              11710
3                                              61027

   Lifetime Post reach by people who like your Page  \
0                                               1640
1                                               6112
3                                              32048

   Lifetime People who have liked your Page and engaged with your post  \
0                                                119
1                                               1108
3                                               1386

   comment    like  share  Total Interactions
0        4    79.0   17.0                 100
1        5   130.0   29.0                 164
3       58  1572.0  147.0                1777
```

In [11]:

```
print(block.loc[0:3])
```

```
   Page total likes    Type  Category  Post Month  Post Weekday  Post Hour  \
0            139441   Photo         2          12             4          3
1            139441  Status         2          12             3         10
2            139441   Photo         3          12             3          3
3            139441   Photo         2          12             2         10

   Paid  Lifetime Post Total Reach  Lifetime Post Total Impressions  \
0   0.0                      2752                             5091
1   0.0                     10460                            19057
2   0.0                      2413                             4373
3   1.0                     50128                            87991

   Lifetime Engaged Users  Lifetime Post Consumers  \
0                     178                      109
1                    1457                     1361
2                     177                      113
3                    2211                      790

   Lifetime Post Consumptions  \
0                         159
1                        1674
2                         154
3                        1119

   Lifetime Post Impressions by people who have liked your Page  \
0                                               3078
1                                              11710
2                                               2812
3                                              61027

   Lifetime Post reach by people who like your Page  \
0                                              1640
1                                              6112
2                                              1503
3                                             32048

   Lifetime People who have liked your Page and engaged with your post  \
0                                                119
1                                               1108
2                                                132
3                                               1386

   comment    like  share  Total Interactions
0        4    79.0   17.0                 100
1        5   130.0   29.0                 164
2        0    66.0   14.0                  80
3       58  1572.0  147.0                1777
```

In [6]:

```
print(block.loc[0:2,['Type','like']])
```

```
     Type   like
0   Photo   79.0
1  Status  130.0
2   Photo   66.0
```

In [7]:

```
print(block.iloc[[0,1,3,6],[0,2]])
```

```
   Page total likes  Category
0            139441         2
1            139441         2
3            139441         2
6            139441         3
```

In [18]:

```
d1 = {'Name': ['Pankaj', 'Meghna', 'Lisa'], 'Country': ['India', 'India', 'USA'], 'Role': ['CEO',
'CTO', 'CTO']}
```

In [19]:

```
df1 = pd.DataFrame(d1)
```

In [20]:

```
print('DataFrame 1:\n', df1)
```

```
DataFrame 1:
      Name Country Role
0  Pankaj   India  CEO
1  Meghna   India  CTO
2    Lisa     USA  CTO
```

In [34]:

```
df2 = pd.DataFrame({'ID': [1, 2, 3], 'Name': ['Pankaj', 'Anupam', 'Amit']})
```

In [35]:

```
print('DataFrame 2:\n', df2)
```

```
DataFrame 2:
    ID    Name
0   1  Pankaj
1   2  Anupam
2   3    Amit
```

In [23]:

```
df_merged = df1.merge(df2)
```

In [24]:

```
print('Result Inner Join:\n', df_merged)
```

```
Result Inner Join:
      Name Country Role  ID
0  Pankaj   India  CEO   1
```

In [25]:

```
print('Result Left Join:\n', df1.merge(df2, how='left'))
```

```
Result Left Join:
      Name Country Role   ID
0  Pankaj   India  CEO  1.0
1  Meghna   India  CTO  NaN
2    Lisa     USA  CTO  NaN
```

In [26]:

```
print('Result Right Join:\n', df1.merge(df2, how='right'))
```

```
Result Right Join:
      Name Country Role  ID
0  Pankaj   India  CEO   1
1  Anupam     NaN  NaN   2
2    Amit     NaN  NaN   3
```

In [27]:

```
print('Result Outer Join:\n', df1.merge(df2, how='outer'))
```

```
Result Outer Join:
      Name Country Role   ID
0  Pankaj   India  CEO  1.0
1  Meghna   India  CTO  NaN
2    Lisa     USA  CTO  NaN
3  Anupam     NaN  NaN  2.0
4    Amit     NaN  NaN  3.0
```

In [28]:

```
print("\nBefore sorting:")
```

```
Before sorting:
```

In [29]:

```
print(data)
```

```
     Page total likes    Type  Category  Post Month  Post Weekday  Post Hour  \
0              139441   Photo         2          12             4          3
1              139441  Status         2          12             3         10
2              139441   Photo         3          12             3          3
3              139441   Photo         2          12             2         10
4              139441   Photo         2          12             2          3
..                ...     ...       ...         ...           ...        ...
495             85093   Photo         3           1             7          2
496             81370   Photo         2           1             5          8
497             81370   Photo         1           1             5          2
498             81370   Photo         3           1             4         11
499             81370   Photo         2           1             4          4

     Paid  Lifetime Post Total Reach  Lifetime Post Total Impressions  \
0     0.0                      2752                             5091
1     0.0                     10460                            19057
2     0.0                      2413                             4373
3     1.0                     50128                            87991
4     0.0                      7244                            13594
..    ...                       ...                              ...
495   0.0                      4684                             7536
496   0.0                      3480                             6229
497   0.0                      3778                             7216
498   0.0                      4156                             7564
499   NaN                      4188                             7292

     Lifetime Engaged Users  Lifetime Post Consumers  \
0                       178                      109
1                      1457                     1361
2                       177                      113
3                      2211                      790
4                       671                      410
..                      ...                      ...
495                     733                      708
496                     537                      508
497                     625                      572
498                     626                      574
499                     564                      524

     Lifetime Post Consumptions  \
0                           159
1                          1674
2                           154
3                          1119
4                           580
..                          ...
495                         985
496                         687
497                         795
498                         832
499                         743

     Lifetime Post Impressions by people who have liked your Page  \
0                                                    3078
1                                                   11710
2                                                    2812
3                                                   61027
4                                                    6228
..                                                    ...
495                                                  4750
496                                                  3961
497                                                  4742
498                                                  4534
499                                                  3861
```

```
     Lifetime Post reach by people who like your Page  \
0                                                1640
1                                                6112
2                                                1503
3                                               32048
4                                                3200
..                                                ...
495                                              2876
496                                              2104
497                                              2388
498                                              2452
499                                              2200

     Lifetime People who have liked your Page and engaged with your post  \
0                                                  119
1                                                 1108
2                                                  132
3                                                 1386
4                                                  396
..                                                 ...
495                                                392
496                                                301
497                                                363
498                                                370
499                                                316

     comment    like  share  Total Interactions
0          4    79.0   17.0                 100
1          5   130.0   29.0                 164
2          0    66.0   14.0                  80
3         58  1572.0  147.0                1777
4         19   325.0   49.0                 393
..       ...     ...    ...                 ...
495        5    53.0   26.0                  84
496        0    53.0   22.0                  75
497        4    93.0   18.0                 115
498        7    91.0   38.0                 136
499        0    91.0   28.0                 119

[500 rows x 19 columns]
```

In [39]:

```
#this is the practical no_2

import numpy as np
csvData = pd.read_csv("C:/Users/vikas pawar/Downloads/records.csv")
```

In [40]:

```
# displaying unsorted data frame
print("\nBefore sorting:")
print(csvData)
```

```
Before sorting:
   ID    Name    Role  Salary
0   1  Pankaj  Editor   10000
1   2    Lisa  Editor    8000
2   3   David  Author    6000
3   4     Ram  Author    4000
4   5  Anupam  Author    5000
```

In [47]:

```
# sort data frame
csvData.sort_values(["Salary"], axis=0, ascending=[False], inplace=True)
csvData.sort_values(["Salary"],axis=0, ascending=[True], inplace=True)
```

In [42]:

```
# displaying sorted data frame
print("\nAfter sorting:")
print(csvData)
```

```
After sorting:
   ID    Name    Role  Salary
3   4     Ram  Author    4000
4   5  Anupam  Author    5000
2   3   David  Author    6000
1   2    Lisa  Editor    8000
0   1  Pankaj  Editor   10000
```

In [43]:

```
#Transpose
print(csvData)
print(csvData.T)
print(csvData.shape)
print(csvData.size)
```

```
   ID    Name    Role  Salary
3   4     Ram  Author    4000
4   5  Anupam  Author    5000
2   3   David  Author    6000
1   2    Lisa  Editor    8000
0   1  Pankaj  Editor   10000
             3       4       2       1       0
ID           4       5       3       2       1
Name       Ram  Anupam   David    Lisa  Pankaj
Role    Author  Author  Author  Editor  Editor
Salary    4000    5000    6000    8000   10000
(5, 4)
20
```

In [44]:

```
print(csvData.size)
```

```
20
```

In [45]:

```
print(csvData.T)
```

```
             3       4       2       1       0
ID           4       5       3       2       1
Name       Ram  Anupam   David    Lisa  Pankaj
Role    Author  Author  Author  Editor  Editor
Salary    4000    5000    6000    8000   10000
```

In [46]:

```
print(csvData.shape)
```

```
(5, 4)
```

In [48]:

```
#Shap


print(csvData)
print(csvData.shape)
print(csvData.size)
```

```
   ID    Name    Role  Salary
3   4     Ram  Author    4000
4   5  Anupam  Author    5000
2   3   David  Author    6000
1   2    Lisa  Editor    8000
0   1  Pankaj  Editor   10000
(5, 4)
20
```

In [49]:

```
#this is the practical no_3


## Data Cleaning
# Data cleaning 1
punctuations = '''!()-{}[];:"'\,<>./?@#$%^&*_~`'''
my_str = "Hello!!!, he said ...and went."
no_punct = ""
for char in my_str:
 if(char not in punctuations):
  no_punct = no_punct + char


print(no_punct)
```

```
Hello he said and went
```

In [50]:

```
# Data cleaning 2
import re
s = "Hello!!!, he said ...and went."
s = re.sub(r'[^\w\s]','',s)
# not of word and space character
print(s)
```

```
Hello he said and went
```

In [51]:

```
# Tokanization
import nltk
nltk.download('punkt')
from nltk.tokenize import sent_tokenize, word_tokenize
example_text = "Hello Mr. Pravin, how are you doing today? The weather in lonavala is rainy. The

print(sent_tokenize(example_text))
print(word_tokenize(example_text))
```

```
['Hello Mr. Pravin, how are you doing today?', 'The weather in lonavala is rain
y.', 'The sky is full of cloud.']
['Hello', 'Mr.', 'Pravin', ',', 'how', 'are', 'you', 'doing', 'today', '?', 'The',
'weather', 'in', 'lonavala', 'is', 'rainy', '.', 'The', 'sky', 'is', 'full', 'of',
'cloud', '.']

[nltk_data] Downloading package punkt to C:\Users\vikas
[nltk_data]     pawar\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

In [52]:

```
# Stopwords
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
stop_words = stopwords.words('english')
print(stop_words)
```

```
[nltk_data] Downloading package stopwords to C:\Users\vikas
[nltk_data]     pawar\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!

['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "yo
u've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him',
'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its',
'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who',
'whom', 'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'we
re', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did',
'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'wh
ile', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'thr
ough', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down',
'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'he
re', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few',
'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'sam
e', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't",
'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren',
"aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "had
n't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "might
n't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "should
n't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]
```

In [53]:

```
# Steming
import nltk
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize
stemmer = PorterStemmer()
input_str = "There are several types of stemming algorithms."
input_str = nltk.word_tokenize(input_str)
print (input_str)
for word in input_str:
 print(stemmer.stem(word))
```

```
['There', 'are', 'several', 'types', 'of', 'stemming', 'algorithms', '.']
there
are
sever
type
of
stem
algorithm
.
```

In [54]:

```
# Lamitization
import nltk
nltk.download('wordnet')
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize
lemmatizer = WordNetLemmatizer()
input_str = "There are several cities with mice."
input_str = nltk.word_tokenize(input_str)
print (input_str)
for word in input_str:
 print(lemmatizer.lemmatize(word))
```

```
[nltk_data] Downloading package wordnet to C:\Users\vikas
[nltk_data]     pawar\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!

['There', 'are', 'several', 'cities', 'with', 'mice', '.']

---------------------------------------------------------------------------
LookupError                               Traceback (most recent call last)
File ~\anaconda3\lib\site-packages\nltk\corpus\util.py:84, in LazyCorpusLoader._
_load(self)
     83 try:
---> 84     root = nltk.data.find(f"{self.subdir}/{zip_name}")
     85 except LookupError:

File ~\anaconda3\lib\site-packages\nltk\data.py:583, in find(resource_name, path
s)
    582 resource_not_found = f"\n{sep}\n{msg}\n{sep}\n"
--> 583 raise LookupError(resource_not_found)

LookupError:
```

In [56]:

```
#this is the practical no_12


csvData.head()
print(csvData)
```

```
   ID    Name    Role  Salary
3   4     Ram  Author    4000
4   5  Anupam  Author    5000
2   3   David  Author    6000
1   2    Lisa  Editor    8000
0   1  Pankaj  Editor   10000
```

In [ ]:

```
csvData.info()
```

In [55]:

```
#1. Deleting the column with missing data
updated_df = csvData.dropna(axis=1)
print(updated_df)
updated_df.info()
```

```
   ID    Name    Role  Salary
3   4     Ram  Author    4000
4   5  Anupam  Author    5000
2   3   David  Author    6000
1   2    Lisa  Editor    8000
0   1  Pankaj  Editor   10000
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5 entries, 3 to 0
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   ID      5 non-null      int64 
 1   Name    5 non-null      object
 2   Role    5 non-null      object
 3   Salary  5 non-null      int64 
dtypes: int64(2), object(2)
memory usage: 200.0+ bytes
```

In [57]:

```
#2. Deleting the row with missing data
updated_df = csvData.dropna(axis=0)
print(updated_df)
updated_df.info()
```

```
   ID    Name    Role  Salary
3   4     Ram  Author    4000
4   5  Anupam  Author    5000
2   3   David  Author    6000
1   2    Lisa  Editor    8000
0   1  Pankaj  Editor   10000
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5 entries, 3 to 0
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   ID      5 non-null      int64
 1   Name    5 non-null      object
 2   Role    5 non-null      object
 3   Salary  5 non-null      int64
dtypes: int64(2), object(2)
memory usage: 200.0+ bytes
```

In [58]:

```
#3. Filling the Missing Values – Imputation
updated_df = csvData
updated_df['Salary']=updated_df['Salary'].fillna(updated_df['Salary'].mean())
print(updated_df)
```

```
   ID    Name    Role  Salary
3   4     Ram  Author    4000
4   5  Anupam  Author    5000
2   3   David  Author    6000
1   2    Lisa  Editor    8000
0   1  Pankaj  Editor   10000
```

In [59]:

```
#4. Imputation with an additional column
updated_df = csvData
updated_df['Salaryismissing'] = updated_df['Salary'].isnull()
print(updated_df)
```

```
   ID    Name    Role  Salary  Salaryismissing
3   4     Ram  Author    4000            False
4   5  Anupam  Author    5000            False
2   3   David  Author    6000            False
1   2    Lisa  Editor    8000            False
0   1  Pankaj  Editor   10000            False
```

In [60]:

```
#5. Filling with a Regression Model
testdf = csvData[csvData['Salary'].isnull()==True]
traindf = csvData[csvData['Salary'].isnull()==False]
traindf.drop("Salary",axis=1,inplace=True)
testdf.drop("Salary",axis=1,inplace=True)
print(traindf)
```

```
   ID    Name    Role  Salaryismissing
3   4     Ram  Author            False
4   5  Anupam  Author            False
2   3   David  Author            False
1   2    Lisa  Editor            False
0   1  Pankaj  Editor            False

C:\Users\vikas pawar\AppData\Local\Temp\ipykernel_15048\2528031715.py:5: SettingWi
thCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.o
rg/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  testdf.drop("Salary",axis=1,inplace=True)
```

In [62]:

```
# this is practical no_8

## Data Model
## House Prize Prediction
#Data model_prediction_house_prize
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
data=pd.read_csv('C:/Users/vikas pawar/Downloads/house.csv')
```

In [63]:

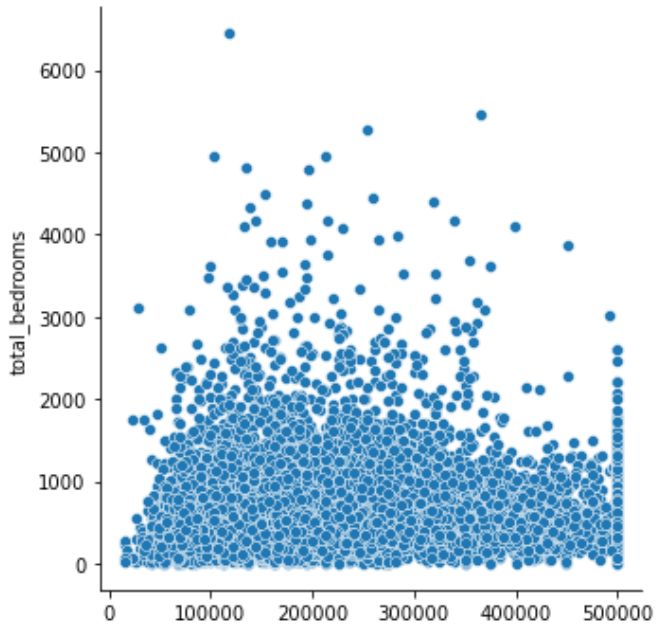
```
print(data.columns) # This will show all the column names
print(data.head(10)) # Show first 10 records of dataframe
print(data.describe()) #You can look at summary of numerical fields by using describe()function

print(data.shape)
print(data.isnull().sum())
```

```
Index(['longitude', 'latitude', 'housing_median_age', 'total_rooms',
       'total_bedrooms', 'population', 'households', 'median_income',
       'median_house_value'],
      dtype='object')
   longitude  latitude  housing_median_age  total_rooms  total_bedrooms  \
0    -114.31     34.19                  15         5612            1283
1    -114.47     34.40                  19         7650            1901
2    -114.56     33.69                  17          720             174
3    -114.57     33.64                  14         1501             337
4    -114.57     33.57                  20         1454             326
5    -114.58     33.63                  29         1387             236
6    -114.58     33.61                  25         2907             680
7    -114.59     34.83                  41          812             168
8    -114.59     33.61                  34         4789            1175
9    -114.60     34.83                  46         1497             309

   population  households  median_income  median_house_value
0        1015         472         1.4936               66900
1        1129         463         1.8200               80100
2         333         117         1.6509               85700
3         515         226         3.1917               73400
4         624         262         1.9250               65500
5         671         239         3.3438               74000
6        1841         633         2.6768               82400
7         375         158         1.7083               48500
8        3134        1056         2.1782               58400
9         787         271         2.1908               48100
          longitude      latitude  housing_median_age   total_rooms  \
count  17000.000000  17000.000000        17000.000000  17000.000000
mean    -119.562108     35.625225           28.589353   2643.664412
std        2.005166      2.137340           12.586937   2179.947071
min     -124.350000     32.540000            1.000000      2.000000
25%     -121.790000     33.930000           18.000000   1462.000000
50%     -118.490000     34.250000           29.000000   2127.000000
75%     -118.000000     37.720000           37.000000   3151.250000
max     -114.310000     41.950000           52.000000  37937.000000

       total_bedrooms    population    households  median_income  \
count    17000.000000  17000.000000  17000.000000   17000.000000
mean       539.410824   1429.573941    501.221941       3.883578
std        421.499452   1147.852959    384.520841       1.908157
min          1.000000      3.000000      1.000000       0.499900
25%        297.000000    790.000000    282.000000       2.566375
50%        434.000000   1167.000000    409.000000       3.544600
75%        648.250000   1721.000000    605.250000       4.767000
max       6445.000000  35682.000000   6082.000000      15.000100

       median_house_value
count        17000.000000
mean        207300.912353
std         115983.764387
min          14999.000000
25%         119400.000000
50%         180400.000000
75%         265000.000000
max         500001.000000
(17000, 9)
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
dtype: int64
```

In [64]:

```
sns.relplot(x='median_house_value', y= 'total_bedrooms', data=data)
sns.relplot(x='median_house_value', y= 'total_rooms', data=data)
sns.relplot(x='median_house_value', y= 'population', data=data)
sns.relplot(x='median_house_value', y= 'median_income', data=data)
sns.relplot(x='median_house_value', y= 'households', data=data)
sns.relplot(x='median_house_value', y= 'median_income', hue= 'total_rooms', data=data)
plt.show()
```



In [65]:

```
#Model
train =data.drop(['median_house_value','longitude','latitude','housing_median_age'],
axis=1)
test =data['median_house_value']
x_train, x_test, y_train, y_test = train_test_split(train, test, test_size=0.3,
random_state=2)

regr= LinearRegression()
regr.fit(x_train, y_train)
pred = regr.predict(x_test)
print(pred)
print(regr.score(x_test, y_test))
```

```
[153579.02594907 221485.91195112  91641.96769418 ... 184433.07649541
 224004.49656511 185189.89991136]
0.541520003241859
```

In [15]:

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

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [15], in <cell line: 1>()
----> 1 data(['longitude', 'latitude', 'housing_median_age', 'total_rooms',
      2 'total_bedrooms', 'population', 'households', 'median_income',
      3 'median_house_value'],
      4  dtype='object')

TypeError: 'DataFrame' object is not callable
```

In [ ]: