Lab 1-Week 1

Goal- To set up a lab notebook.

```
#allows you to add a not within the code cell- the program will not see this as code
#We are going to begin by defining variables
a =2
b =4
c =7


#Create a new variable from the defined list of variables
x = a + c
#print() used to denote what I want to see in the output
print(x)
```

```
9
```

```
x = a/b
print(x)
```

```
0.5
```

```
x = a + b + c
print(x)
```

```
13
```

```
x = c - a
print(x)
```

```
5
```

```
#New variables

d = 2
e = 4
f = 8



x= d+e
print(x)
```

```
6
```

```
x= f/d+a
print(x)
```

```
6.0
```

I set up my first notebook, and set up my first code! Math Commands: multiply * subtract - add + divide /

Practice defining additional variables are in the code lines above. The lab submittion includes a variable list with a new variable.

Week 2 Lab-Taking a look at Fuctions I am going to create a dataframe using a list.

```
#list1 = ['I', 'love','MSM', 'Program']
#Pandas and Numpy are two important libraries in Python - have written functions and allow the programmer to create functions without writing

import pandas as pd
import numpy as np
list1 = ['I', 'love','MSM', 'Program']
df = pd.DataFrame(data=list1)
df
```

	0
0	I
1	love
2	MSM
3	Program

```
list2 = [4,5,9,-3,0,1]
max(list2)
```

```
9
```

```
sorted(list2)
```

```
[-3, 0, 1, 4, 5, 9]
```

```
list2 = [4,5,9,-3,0,1]
list2.remove(4)
print(list2)
```

```
[5, 9, -3, 0, 1]
```

Lab Assignment Question 1 = Make a numeric list, sort by increasing value, remove 2 values. print new list

```
#Adding elements in a given list
list3 = ['R','SAS','Excel'] #basic programs
list3.insert(2,'Python')
list3.insert(4,'Prism')
list3
```

```
['R', 'SAS', 'Python', 'Excel', 'Prism']
```

```
df = pd.DataFrame(data=list3)
df
```

	0
0	R
1	SAS
2	Python
3	Excel
4	Prism

Define Function with an argument

```
def even_odd(num):
  if(num%2==0):
    return 'Even number'
  else:
      return 'Odd number'
```

```
even_odd(32)
```

```
'Even number'
```

```
even_odd(15)
```

```
'Odd number'
```

Define a function using a list

```
def unique(my_list):
  l=[]
  for i in my_list:
    if( i not in l):
      l.append(i)
  return l



list = [ 1,2,2,2,3,5,5,5,14,87,87]
unique(list)
```

```
[1, 2, 3, 5, 14, 87]
```

Double-click (or enter) to edit

LAB 3 Data Analysis

```
from ast import increment_lineno
# Library to supress warnings or deprication notes
import warnings
warnings.filterwarnings('ignore')

#Libraries to help reading and manipulating data
import numpy as np
import pandas as pd

#Libraries to help with data visualization
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns


Lab 3 Lab Assignment
import seaborn as sns
df = pd.read_csv('/content/sample_data/california_housing_train.csv')
df.head(2)
```

1 to 2 of 2 entries Filter

index	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
0	-114.31	34.19	15.0	5612.0	1283.0	1015.0	472.0	1.4936	66900.0
1	-114.47	34.4	19.0	7650.0	1901.0	1129.0	463.0	1.82	80100.0

Show 25 per page

```
df.head()
```

1 to 5 of 5 entries Filter

index	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households
0	-114.31	34.19	15.0	5612.0	1283.0	1015.0	472.0
1	-114.47	34.4	19.0	7650.0	1901.0	1129.0	463.0
2	-114.56	33.69	17.0	720.0	174.0	333.0	117.0
3	-114.57	33.64	14.0	1501.0	337.0	515.0	226.0
4	-114.57	33.57	20.0	1454.0	326.0	624.0	262.0

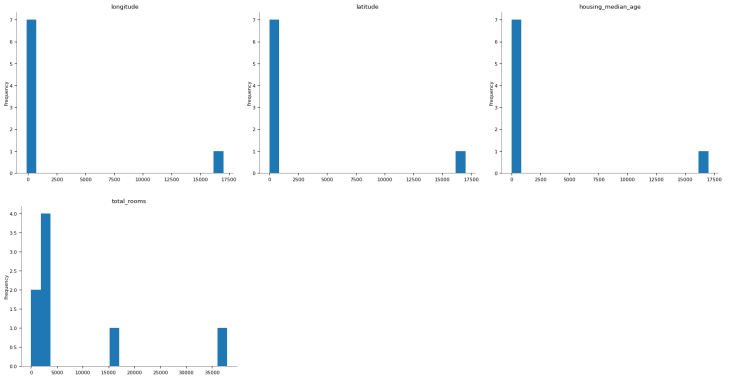
```
df.describe()
```

1 to 8 of 8 entries Filter

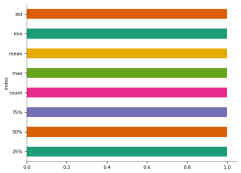
index	longitude	latitude	housing_median_age	total_rooms	total_be
count	17000.0	17000.0	17000.0	17000.0	
mean	-119.5621082352941	35.62522470588235	28.58935294117647	2643.664411764706	539.41082
std	2.0051664084261778	2.1373397946570836	12.586936981660399	2179.947071452767	421.49945
min	-124.35	32.54	1.0	2.0	
25%	-121.79	33.93	18.0	1462.0	
50%	-118.49	34.25	29.0	2127.0	
75%	-118.0	37.72	37.0	3151.25	
max	-114.31	41.95	52.0	37937.0	

Show 25 per page

Distributions

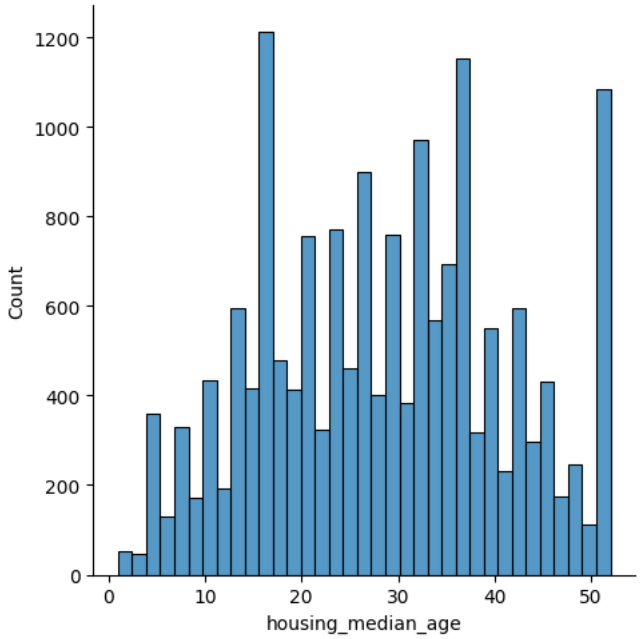


Categorical distributions

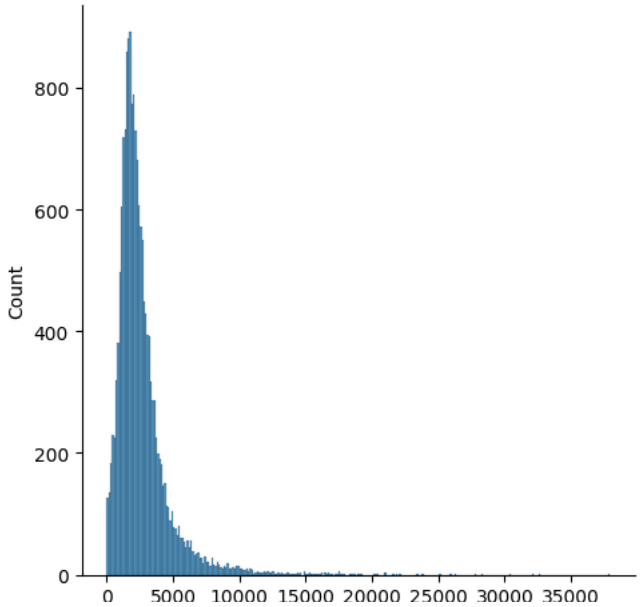


2-d distributions

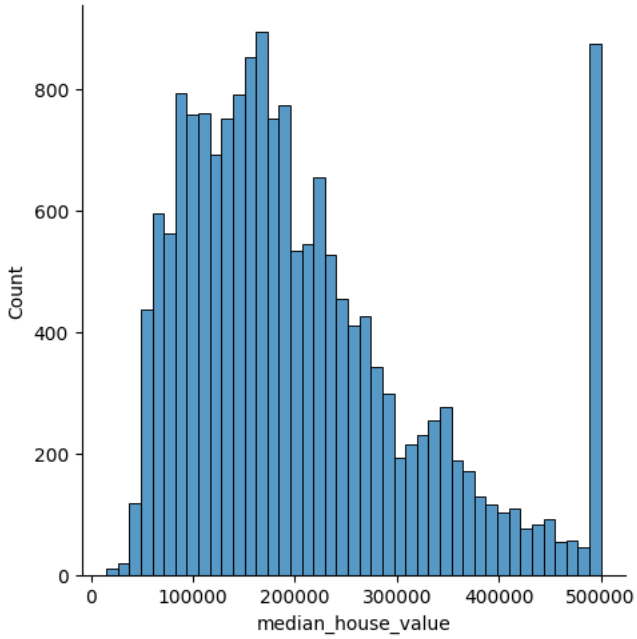
```
sns.displot(df['housing_median_age'])
plt.show()
```




```
sns.displot(df['total_rooms'])
plt.show()
```




```
sns.displot(df['median_house_value'])
plt.show()
```




```
corr_matrix = df.corr()
corr_matrix
```

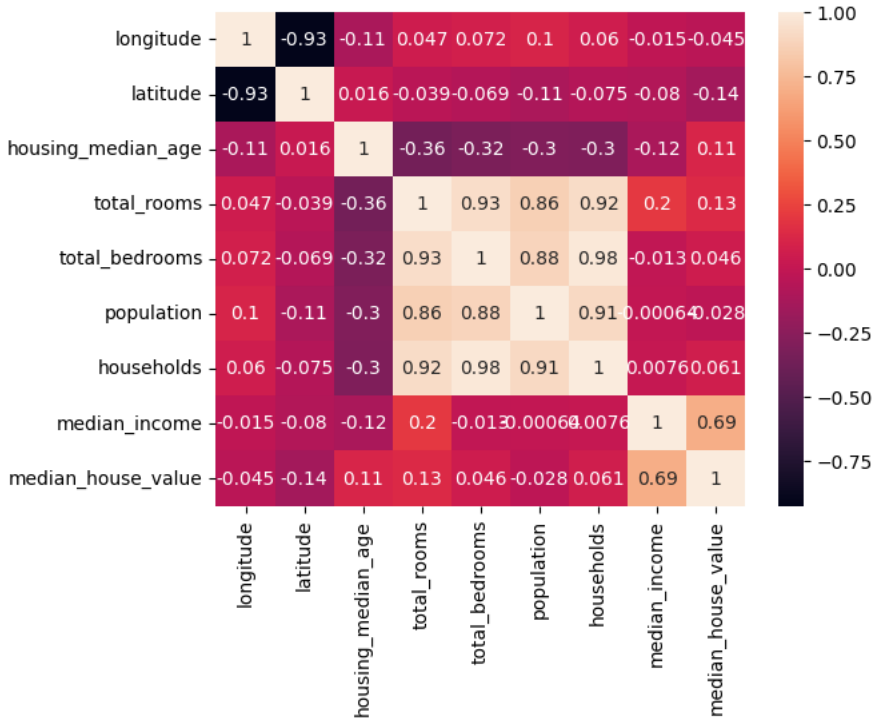
1 to 9 of 9 entries Filter

index	longitude	latitude	housing_median_age	to
longitude	1.0	-0.9252082786792101	-0.11425030616316861	0.0470
latitude	-0.9252082786792101	1.0	0.016453903095023946	-0.0387
housing_median_age	-0.11425030616316861	0.016453903095023946	1.0	-0.360
total_rooms	0.047010440328565675	-0.038772574164864966	-0.36098416572528785	
total_bedrooms	0.07180195592382516	-0.06937291517634289	-0.3204340826318241	0.92
population	0.1016742645684225	-0.11126136149822226	-0.29588980535867854	0.86
households	0.059627704209074844	-0.07490229668637566	-0.302754191175035	0.91
median_income	-0.015484961384791378	-0.08030301379233419	-0.11593162461581347	0.195
median_house_value	-0.044981696510901864	-0.1449167173376358	0.10675770707287582	0.130

Show 25 per page

```
num_cols = ['house_median_age','total_rooms','median_house_value']
```

```
sns.heatmap(corr_matrix,annot = True)
plt.show()
```




```
df.describe()
```

1 to 8 of 8 entries Filter

index	longitude	latitude	housing_median_age	total_rooms	total_be
count	17000.0	17000.0	17000.0	17000.0	
mean	-119.5621082352941	35.62522470588235	28.58935294117647	2643.664411764706	539.41082
std	2.0051664084261778	2.1373397946570836	12.586936981660399	2179.947071452767	421.49945
min	-124.35	32.54	1.0	2.0	
25%	-121.79	33.93	18.0	1462.0	
50%	-118.49	34.25	29.0	2127.0	
75%	-118.0	37.72	37.0	3151.25	
max	-114.31	41.95	52.0	37937.0	

Show 25 per page

Lab 3 Example

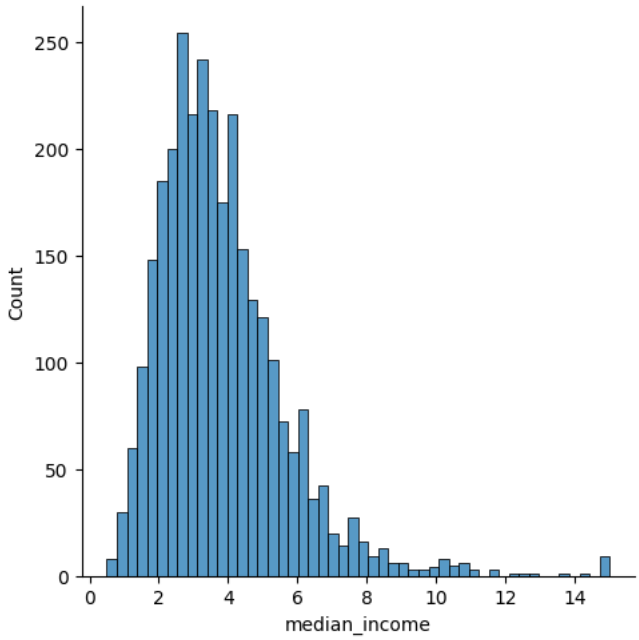
```
#read in the data
df = pd.read_csv('/content/sample_data/california_housing_test.csv')
df.head(2)
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	hou
0	-122.05	37.37	27.0	3885.0	661.0	1537.0	
1	-118.30	34.26	43.0	1510.0	310.0	809.0	

```
df.head()
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	hou
0	-122.05	37.37	27.0	3885.0	661.0	1537.0	
1	-118.30	34.26	43.0	1510.0	310.0	809.0	
2	-117.81	33.78	27.0	3589.0	507.0	1484.0	
3	-118.36	33.82	28.0	67.0	15.0	49.0	
4	-119.67	36.33	19.0	1241.0	244.0	850.0	

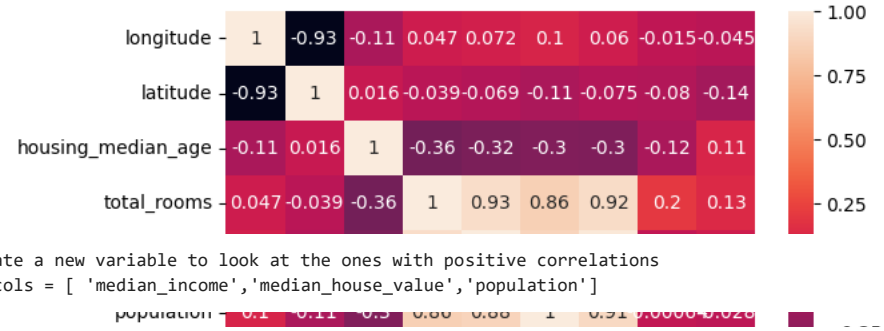
```
sns.displot(df['median_income'])
plt.show()
```




```
#looking at correlations - relationships within the data
corr_matrix = df.corr()
corr_matrix
```

	longitude	latitude	housing_median_age	total_rooms	total_bedroom
longitude	1.000000	-0.925208	-0.114250	0.047010	0.07180
latitude	-0.925208	1.000000	0.016454	-0.038773	-0.06937
housing_median_age	-0.114250	0.016454	1.000000	-0.360984	-0.32043
total_rooms	0.047010	-0.038773	-0.360984	1.000000	0.92840
total_bedrooms	0.071802	-0.069373	-0.320434	0.928403	1.00000
population	0.101674	-0.111261	-0.295890	0.860170	0.88116
households	0.059628	-0.074902	-0.302754	0.919018	0.98092
median_income	-0.015485	-0.080303	-0.115932	0.195383	-0.01349
median_house_value	-0.044982	-0.144917	0.106758	0.130991	0.04578

```
#look at the relationship of variables using a heatmap
sns.heatmap(corr_matrix,annot = True)
plt.show()
```




```
#create a new variable to look at the ones with positive correlations
num_cols = [ 'median_income','median_house_value','population']
```

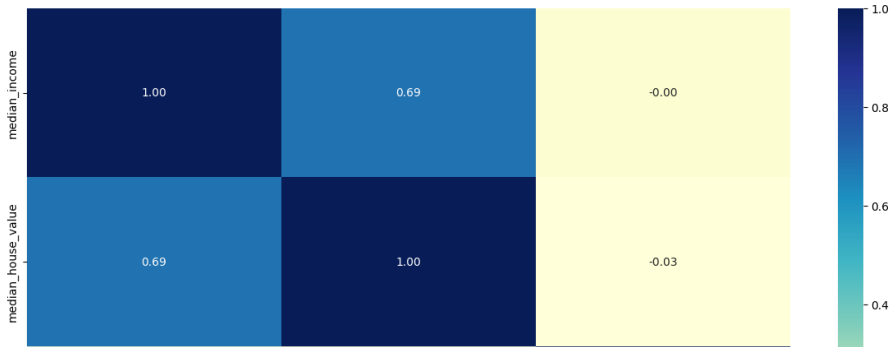
```
df[num_cols].describe().T
```

	count	mean	std	min	25%	
median_income	17000.0	3.883578	1.908157	0.4999	2.566375	
median_house_value	17000.0	207300.912353	115983.764387	14999.0000	119400.000000	180
population	17000.0	1429.573941	1147.852959	3.0000	790.000000	1

```
#look at your data visually
df[num_cols].hist(figsize = (14,14))
plt.show
```

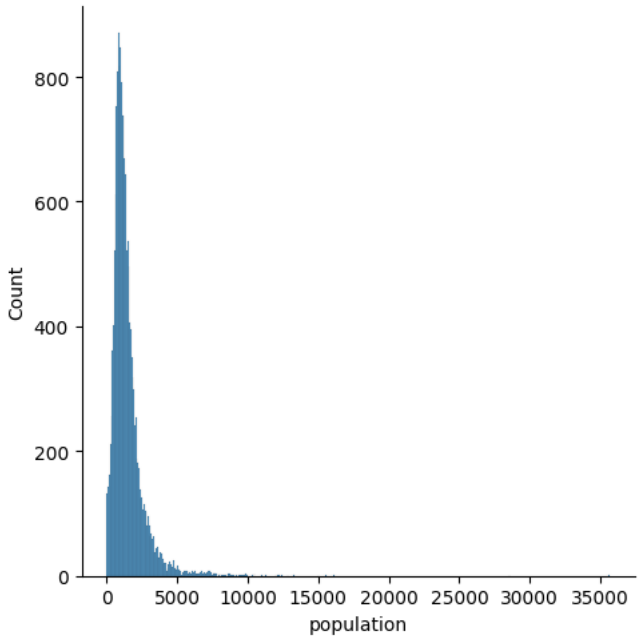
```
#look at the correlation between the variables you are interested in
plt.figure(figsize = (15, 8))
sns.heatmap(df[num_cols].corr(), annot = True, fmt = '0.2f', cmap = 'YlGnBu')
```

```
<Axes: >
```




```
#data distribution
sns.displot(df['population'])
plt.show()
```

```
#data distribution
sns.displot(df['population'])
plt.show()
```




```
#Look at basic statistics
df.describe()
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	popula
count	3000.000000	3000.00000	3000.000000	3000.000000	3000.000000	3000.0(
mean	-119.589200	35.63539	28.845333	2599.578667	529.950667	1402.7(
std	1.994936	2.12967	12.555396	2155.593332	415.654368	1030.5(
min	-124.180000	32.56000	1.000000	6.000000	2.000000	5.0(
25%	-121.810000	33.93000	18.000000	1401.000000	291.000000	780.0(
50%	-118.485000	34.27000	29.000000	2106.000000	437.000000	1155.0(
75%	-118.020000	37.69000	37.000000	3129.000000	636.000000	1742.7(
max	-114.490000	41.92000	52.000000	30450.000000	5419.000000	11935.0(