Nikit Gokhe

Class: TY Comp D1

Roll No. 324022

Gr No. 21810522

Assignment 2

Problem Statement: Perform the following operations using Python on the data sets. Compute and display summary statistics for each feature available in the dataset.(eg. minimum value, maximum value, mean, range, standard deviation, variance and percentiles) ·

Data Visualization-Create a histogram for each feature in the dataset to illustrate the feature distributions, Data cleaning, Data integration, Data transformation

Objective:

1. Cleaning of Data
2. Creating histogram for each feature to understand trends
3. Creating Plots to find correlation

Theory:

Summary statistics:

Pandas **describe()** is used to view some basic statistical details like percentile, mean, std etc. of a data frame or a series of numeric values. When this method is applied to a series of string, it returns a different output which is shown in the examples below.

Check for Missing Values

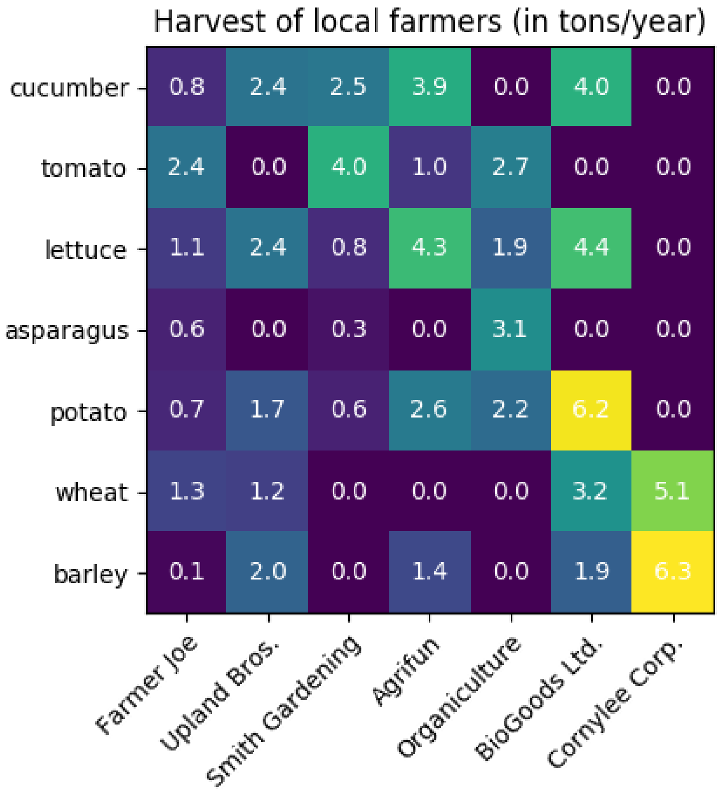
To make detecting missing values easier (and across different array dtypes), Pandas provides the **isnull()** and **notnull()** functions, which are also methods on Series and DataFrame objects.

DataFrame.astype()

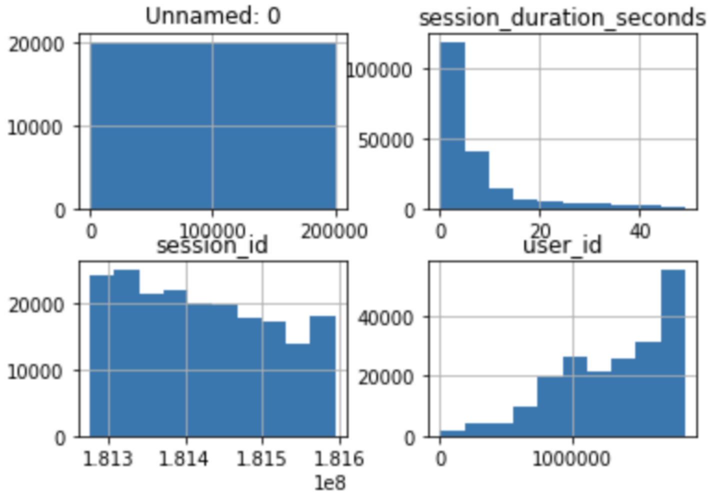
We can pass any Python, Numpy or Pandas datatype to change all columns of a dataframe to that type, or we can pass a dictionary having column names as keys and datatype as values to change type of selected columns.

Graphs :

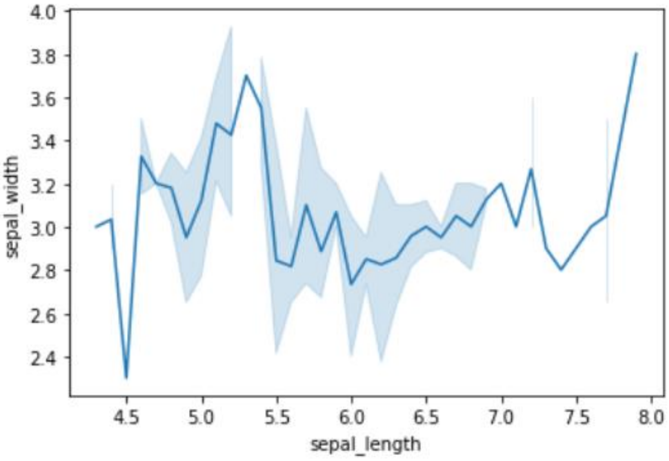
Heatmap: A heatmap contains values representing various shades of the same colour for each value to be plotted. Usually the darker shades of the chart represent higher values than the lighter shade. For a very different value a completely different colour can also be used.



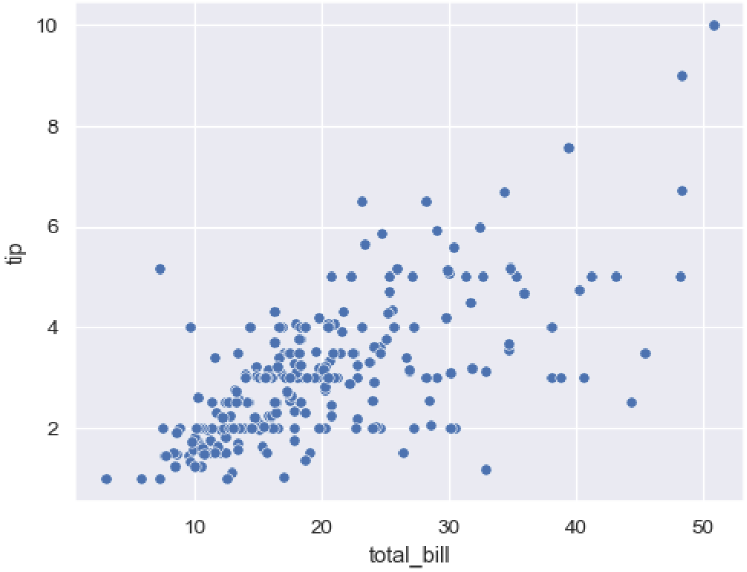
Histogram: A common way of visualizing the distribution of a single numerical variable is by using a histogram. A histogram divides the values within a numerical variable into “bins”, and counts the number of observations that fall into each bin. By visualizing these binned counts in a columnar fashion, we can obtain a very immediate and intuitive sense of the distribution of values within a variable.



Lineplot: Seaborn Line Plots depict the relationship between continuous as well as categorical values in a continuous data point format.



Scatter Plot: Scatter Plot represents the relationship between two continuous values, respectively. It depicts how one data variable gets affected by the other data variable in every fraction of the value of the data set.

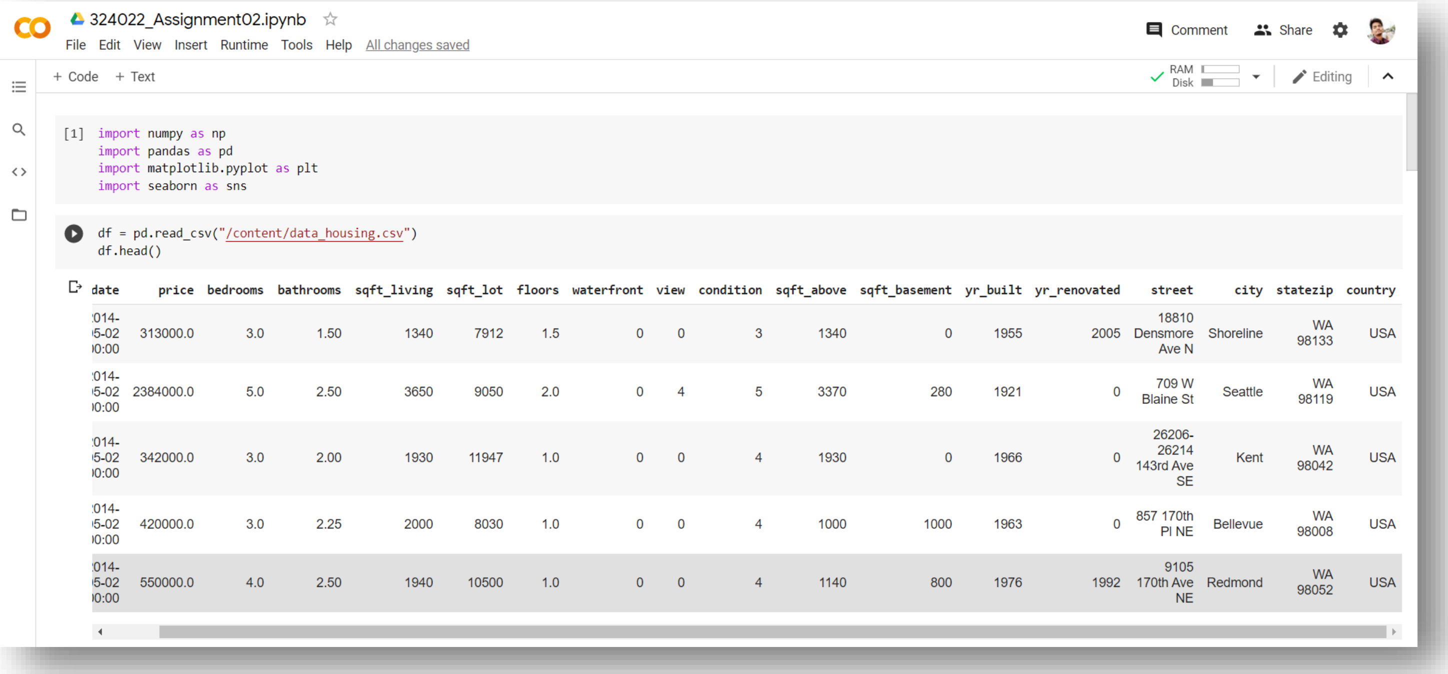


Dataset:

Name: data_housing

Link: https://www.kaggle.com/shree1992/housedata

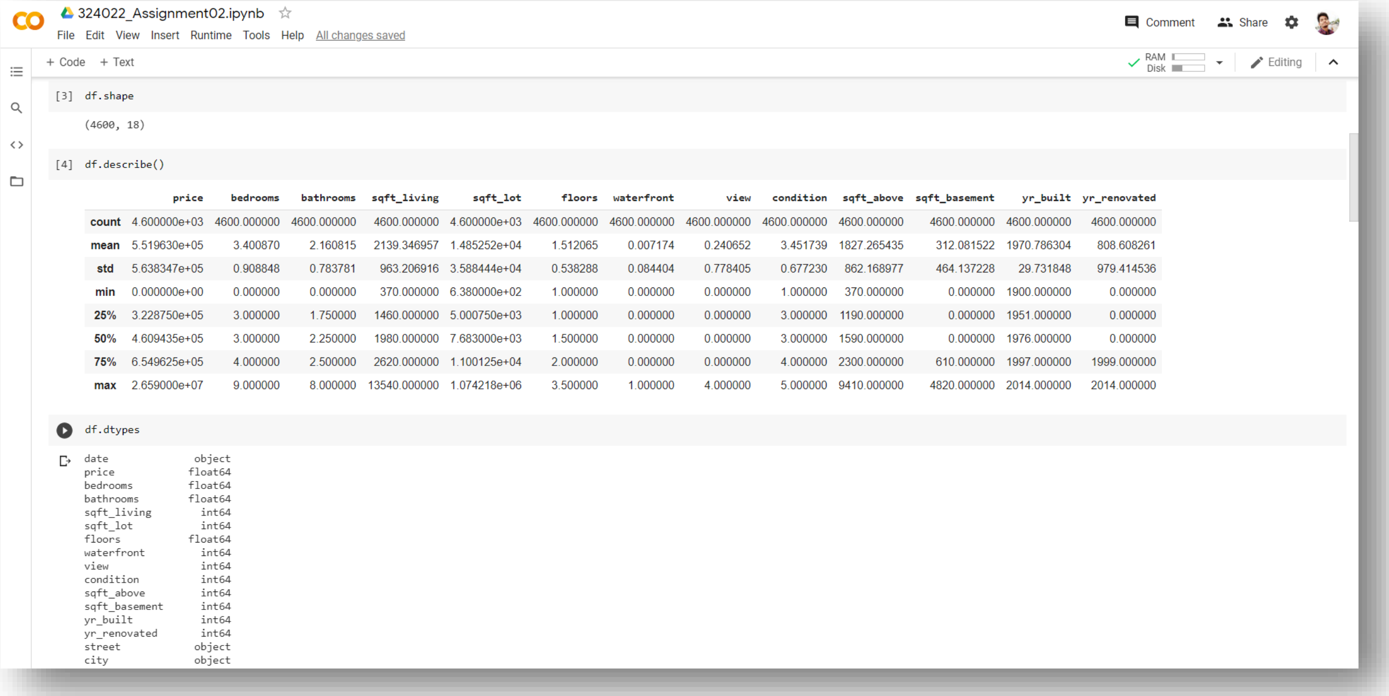
Expected Output/sample code:


324022_Assignment02.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing

```
[1] import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
```

```
df = pd.read_csv("/content/data_housing.csv")
df.head()
```

date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built	yr_renovated	street	city	statezip	country
014-5-02 0:00	313000.0	3.0	1.50	1340	7912	1.5	0	0	3	1340	0	1955	2005	18810 Densmore Ave N	Shoreline	WA 98133	USA
014-5-02 0:00	2384000.0	5.0	2.50	3650	9050	2.0	0	4	5	3370	280	1921	0	709 W Blaine St	Seattle	WA 98119	USA
014-5-02 0:00	342000.0	3.0	2.00	1930	11947	1.0	0	0	4	1930	0	1966	0	26206-26214 143rd Ave SE	Kent	WA 98042	USA
014-5-02 0:00	420000.0	3.0	2.25	2000	8030	1.0	0	0	4	1000	1000	1963	0	857 170th Pl NE	Bellevue	WA 98008	USA
014-5-02 0:00	550000.0	4.0	2.50	1940	10500	1.0	0	0	4	1140	800	1976	1992	9105 170th Ave NE	Redmond	WA 98052	USA

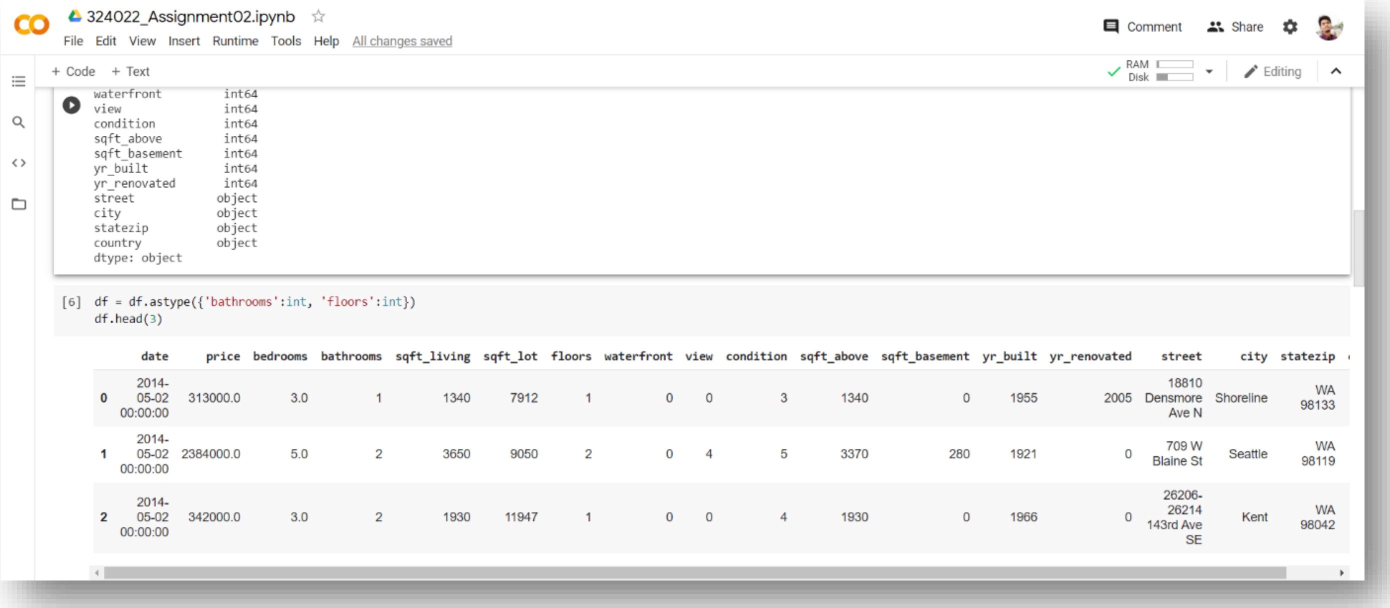

324022_Assignment02.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM
Disk
Editing

```
[3] df.shape

    (4600, 18)
```

```
[4] df.describe()
```

	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built	yr_renovated
count	4.600000e+03	4600.000000	4600.000000	4600.000000	4.600000e+03	4600.000000	4600.000000	4600.000000	4600.000000	4600.000000	4600.000000	4600.000000	4600.000000
mean	5.519630e+05	3.400870	2.160815	2139.346957	1.485252e+04	1.512065	0.007174	0.240652	3.451739	1827.265435	312.081522	1970.786304	808.608261
std	5.638347e+05	0.908848	0.783781	963.206916	3.588444e+04	0.538288	0.084404	0.778405	0.677230	862.168977	464.137228	29.731848	979.414536
min	0.000000e+00	0.000000	0.000000	370.000000	6.380000e+02	1.000000	0.000000	0.000000	1.000000	370.000000	0.000000	1900.000000	0.000000
25%	3.228750e+05	3.000000	1.750000	1460.000000	5.000750e+03	1.000000	0.000000	0.000000	3.000000	1190.000000	0.000000	1951.000000	0.000000
50%	4.609435e+05	3.000000	2.250000	1980.000000	7.683000e+03	1.500000	0.000000	0.000000	3.000000	1590.000000	0.000000	1976.000000	0.000000
75%	6.549625e+05	4.000000	2.500000	2620.000000	1.100125e+04	2.000000	0.000000	0.000000	4.000000	2300.000000	610.000000	1997.000000	1999.000000
max	2.659000e+07	9.000000	8.000000	13540.000000	1.074218e+06	3.500000	1.000000	4.000000	5.000000	9410.000000	4820.000000	2014.000000	2014.000000

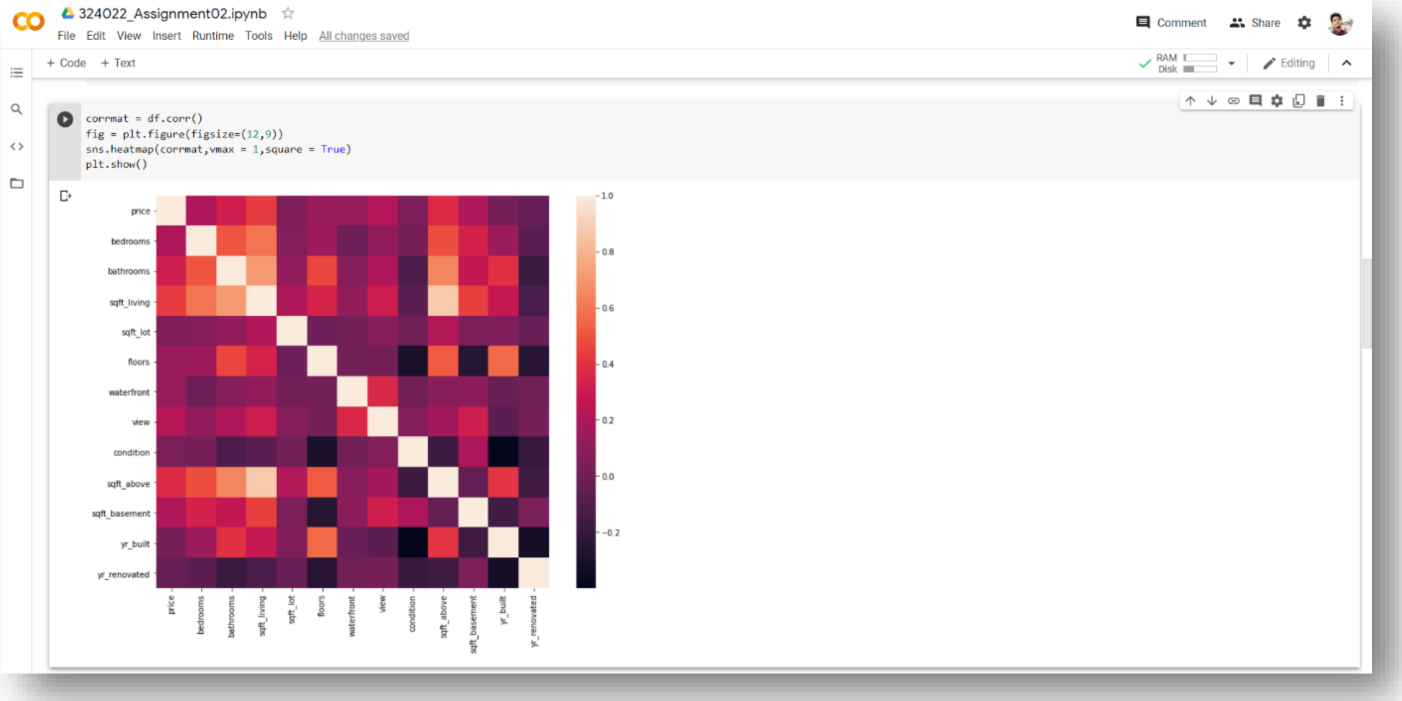
```
df.dtypes

date              object
price            float64
bedrooms         float64
bathrooms        float64
sqft_living        int64
sqft_lot           int64
floors           float64
waterfront         int64
view               int64
condition          int64
sqft_above         int64
sqft_basement      int64
yr_built           int64
yr_renovated       int64
street            object
city              object
```


324022_Assignment02.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM
Disk
Editing

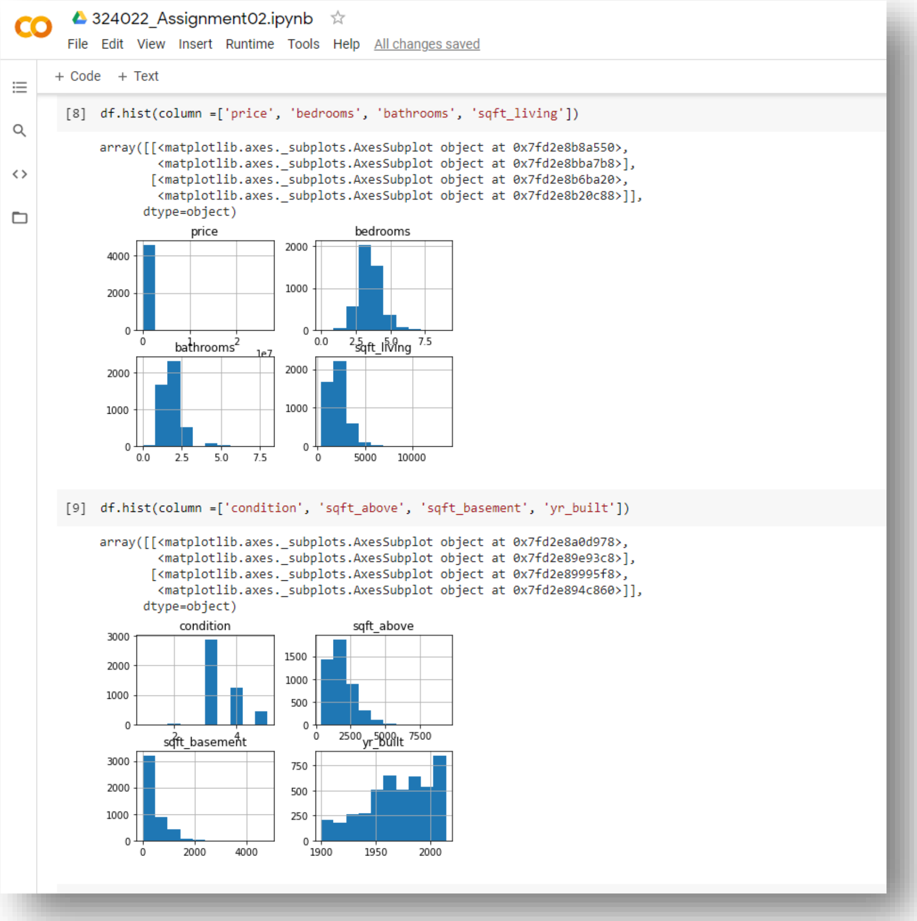
```
waterfront         int64
view               int64
condition          int64
sqft_above         int64
sqft_basement      int64
yr_built           int64
yr_renovated       int64
street            object
city              object
statezip          object
country           object
dtype: object
```

```
[6] df = df.astype({'bathrooms':int, 'floors':int})
    df.head(3)
```

	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built	yr_renovated	street	city	statezip
0	2014-05-02 00:00:00	313000.0	3.0	1	1340	7912	1	0	0	3	1340	0	1955	2005	18810 Densmore Ave N	Shoreline	WA 98133
1	2014-05-02 00:00:00	2384000.0	5.0	2	3650	9050	2	0	4	5	3370	280	1921	0	709 W Blaine St	Seattle	WA 98119
2	2014-05-02 00:00:00	342000.0	3.0	2	1930	11947	1	0	0	4	1930	0	1966	0	26206-26214 143rd Ave SE	Kent	WA 98042

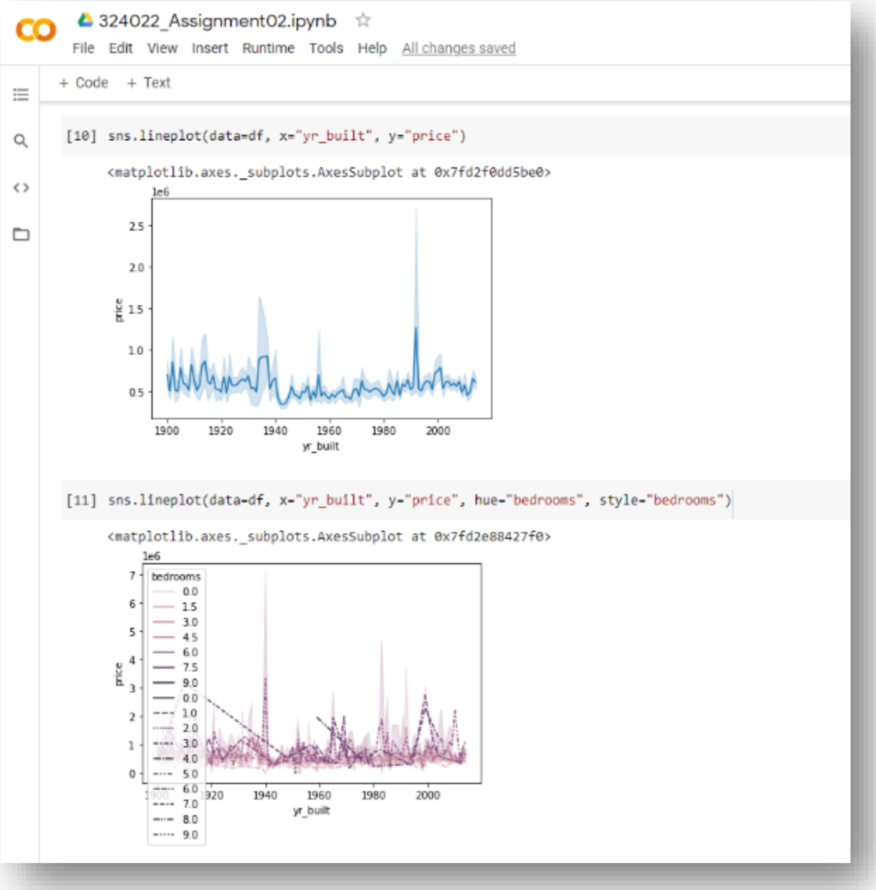


```
corrmat = df.corr()
fig = plt.figure(figsize=(12,9))
sns.heatmap(corrmat,vmax = 1,square = True)
plt.show()
```



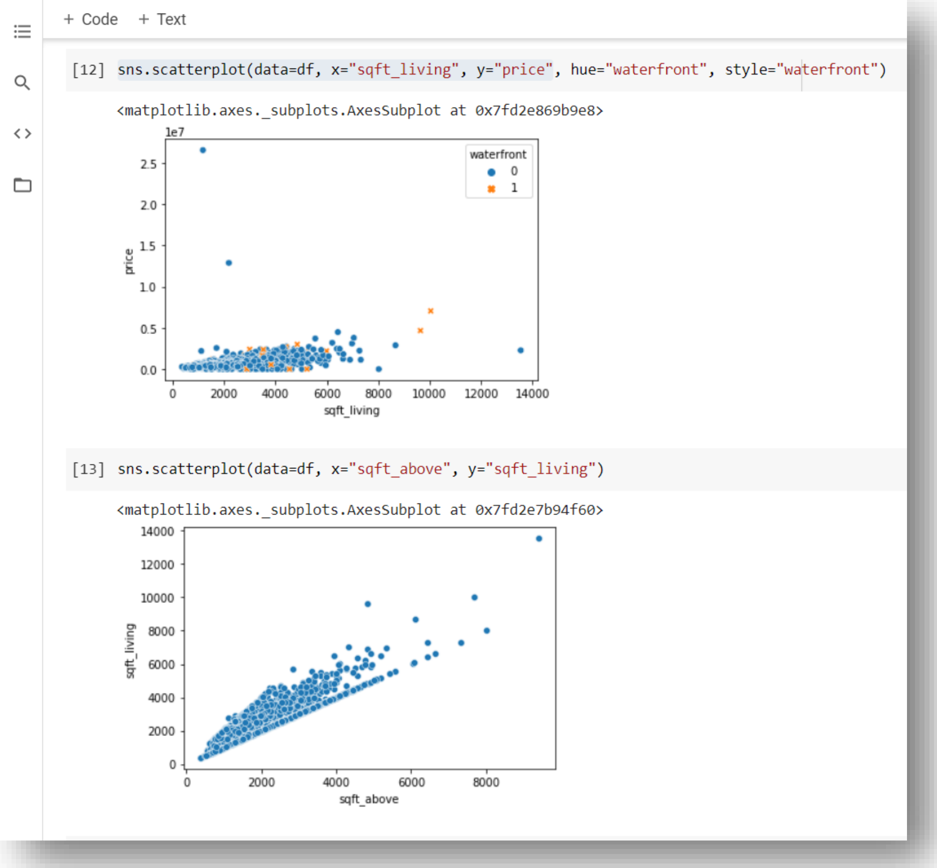
```
[8]  df.hist(column =['price', 'bedrooms', 'bathrooms', 'sqft_living'])

     array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7fd2e8b8a550>,
             <matplotlib.axes._subplots.AxesSubplot object at 0x7fd2e8bba7b8>],
            [<matplotlib.axes._subplots.AxesSubplot object at 0x7fd2e8b6ba20>,
             <matplotlib.axes._subplots.AxesSubplot object at 0x7fd2e8b20c88>]],
           dtype=object)
```

```
[9]  df.hist(column =['condition', 'sqft_above', 'sqft_basement', 'yr_built'])

     array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7fd2e8a0d978>,
             <matplotlib.axes._subplots.AxesSubplot object at 0x7fd2e89e93c8>],
            [<matplotlib.axes._subplots.AxesSubplot object at 0x7fd2e89995f8>,
             <matplotlib.axes._subplots.AxesSubplot object at 0x7fd2e894c860>]],
           dtype=object)
```



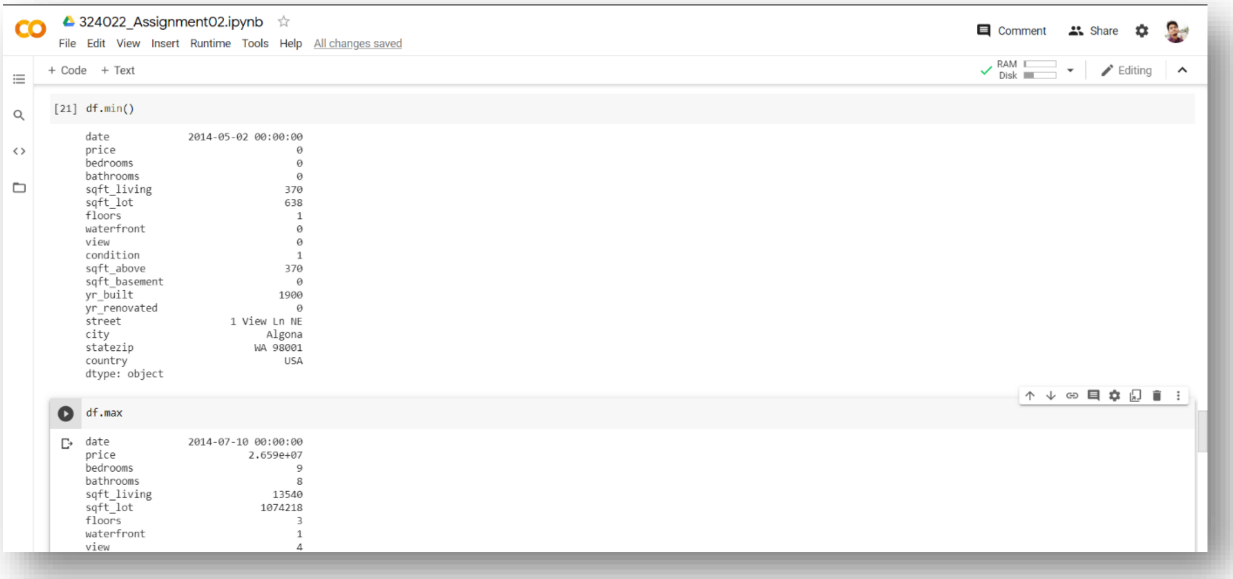
```
[10] sns.lineplot(data=df, x="yr_built", y="price")

<matplotlib.axes._subplots.AxesSubplot at 0x7fd2f0dd5be0>
```

```
[11] sns.lineplot(data=df, x="yr_built", y="price", hue="bedrooms", style="bedrooms")

<matplotlib.axes._subplots.AxesSubplot at 0x7fd2e88427f0>
```



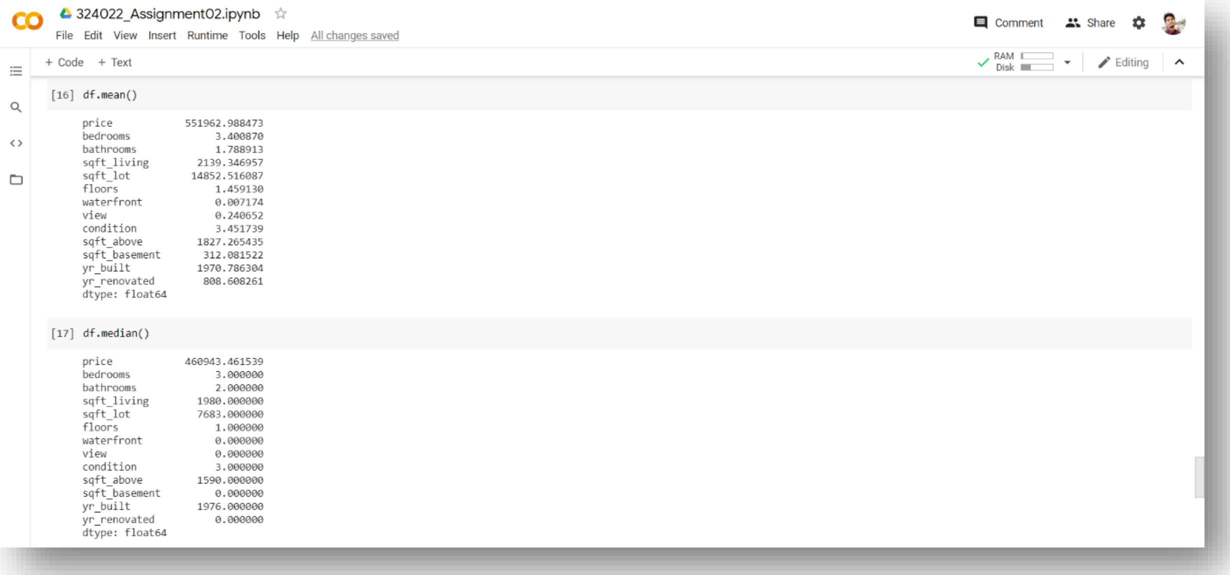
```
[12] sns.scatterplot(data=df, x="sqft_living", y="price", hue="waterfront", style="waterfront")

<matplotlib.axes._subplots.AxesSubplot at 0x7fd2e869b9e8>
```

```
[13] sns.scatterplot(data=df, x="sqft_above", y="sqft_living")

<matplotlib.axes._subplots.AxesSubplot at 0x7fd2e7b94f60>
```


324022_Assignment02.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing

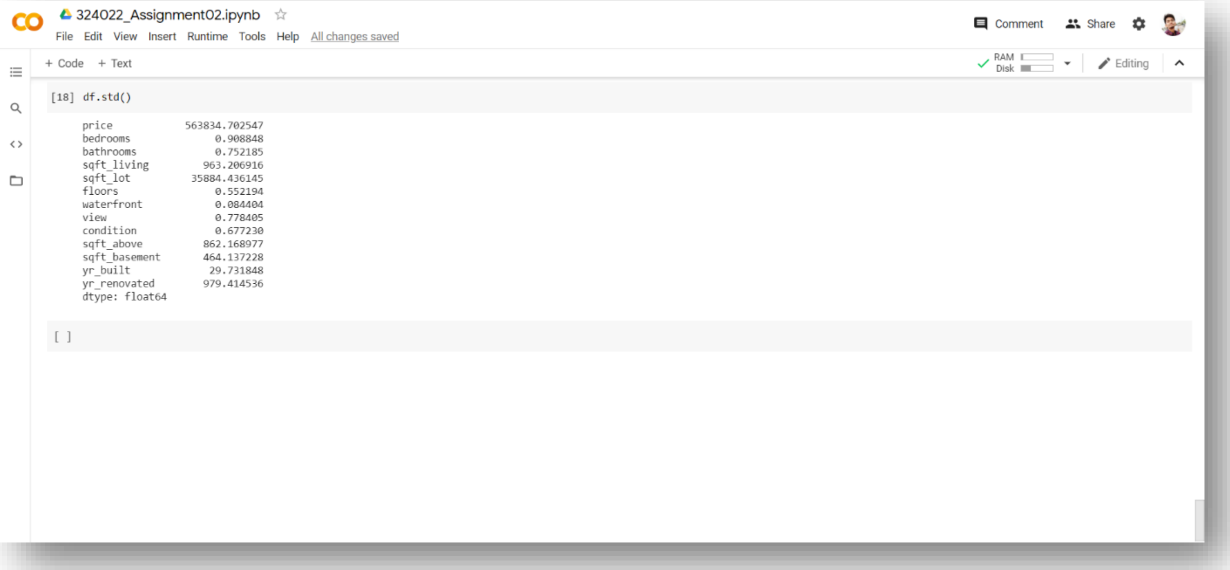
```
[21] df.min()

     date              2014-05-02 00:00:00
     price                               0
     bedrooms                            0
     bathrooms                           0
     sqft_living                       370
     sqft_lot                          638
     floors                              1
     waterfront                          0
     view                                0
     condition                           1
     sqft_above                        370
     sqft_basement                       0
     yr_built                         1900
     yr_renovated                        0
     street                     1 View Ln NE
     city                           Algona
     statezip                     WA 98001
     country                           USA
     dtype: object
```

```
df.max

     date              2014-07-10 00:00:00
     price                       2.659e+07
     bedrooms                            9
     bathrooms                           8
     sqft_living                     13540
     sqft_lot                      1074218
     floors                              3
     waterfront                          1
     view                                4
```


324022_Assignment02.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing

```
[16] df.mean()

     price            551962.988473
     bedrooms              3.400870
     bathrooms             1.788913
     sqft_living        2139.346957
     sqft_lot          14852.516087
     floors                1.459130
     waterfront            0.007174
     view                  0.240652
     condition             3.451739
     sqft_above         1827.265435
     sqft_basement       312.081522
     yr_built           1970.786304
     yr_renovated        808.608261
     dtype: float64
```

```
[17] df.median()

     price            460943.461539
     bedrooms              3.000000
     bathrooms             2.000000
     sqft_living        1980.000000
     sqft_lot           7683.000000
     floors                1.000000
     waterfront            0.000000
     view                  0.000000
     condition             3.000000
     sqft_above         1590.000000
     sqft_basement         0.000000
     yr_built           1976.000000
     yr_renovated          0.000000
     dtype: float64
```



324022_Assignment02.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code
+ Text
RAM
Disk
Editing

```
[18] df.std()

price            563834.702547
bedrooms              0.908848
bathrooms             0.752185
sqft_living         963.206916
sqft_lot          35884.436145
floors                0.552194
waterfront            0.084404
view                  0.778405
condition             0.677230
sqft_above          862.168977
sqft_basement       464.137228
yr_built             29.731848
yr_renovated        979.414536
dtype: float64
```

Inference:

1.Understood the statistical summary of the data for each numerical column.

2.Changed datatype of floors and bathrooms.

3.Used heatmap to find correlation between columns. Hence found highest correlation between sqrt_living and sqrt_above.

4.Most people have 3 bedrooms and 2 bathrooms.

5.People prefer having single floor.

6.Average value of house increases as number of bedroom increases.

7.Having waterfront does not significantly affect the house prices.