Pandas Assignment

By Kanika Kanwar

Description of the Dataset: housing.csv
This dataset contains these columns: id, date, price, bedrooms, bathrooms, sqft_living, sqft_lot, floors, waterfront, view, condition, grade, sqft_above, sqft_basement, yr_built, yr_renovated, zip code, lat, long, sqft_living15, sqft_lot15.

1. How many houses have a waterfront?
A. 140
B. 340
C. 323
D. 163

2. Which zip code has the costliest house?
A. 98456
B. 98102
C. 98283
D. 89034

3. How many houses are having grade 10?
A. 1133
B. 1134
C. 1135
D. 1132

4. How many null values are there in the dataset?
A. 20
B. 19
C. 0
D. 4

5. Does this 9126100861 customer have a waterfront?
A. Yes
B. No

6. How many houses have 3 views?
A. 340
B. 550
C. 510
D. 90

7. What is the lowest price of the house?
A. 45000
B. 28000
C. 75000
D. 25000

8. Which zip code area has the cheapest house?
A. 98000
B. 98022
C. 80000
D. 94500

9. Which area has the biggest sqft_living?
A. 98053
B. 98055
C. 98035
D. None of the above

10. In which year the costliest house has been built?
A. 2000
B. 1992
C. 1911
D. 1910

11. Which of the following data types can a Pandas Series have?
A. Int
B. Str
C. Float
D. All of the above

12. Which of the following is having a one-dimensional array?
A. Series data type
B. DataFrame data type
C. Both of the above
D. None of the above

13. What value will you get on the left side after printing a series format data?
A. Data
B. Value
C. Index
D. All of the above

14. Which of the following is considered as data in pandas?
A. Dictionary
B. Boolean
C. Nd-array
D. All of the above

15. In what format the keys will get converted into when we convert the dictionary data into data frame format?
A. Rows
B. Columns
C. Indexes

D. Records

16. Write a function to perform the following operation:
Increase the grade of the house by 1 if the sqft of the house is greater than 0 and less than equal to 400. If the total sqft of the house is greater than 400, increase the grade of the house
by 2.
After creating the above function, which syntax will be using when using apply function to run the above created function in the dataset?
A. data.apply(your_function_name(arg1))
B. data.apply_fun(function_name())
C. data.app(data)
D. None of the above

17. How to check the duplicate values on id,grade and location?
A. df.duplicated(subset=['id','zipcode','grade'])
B. df.duplicate(subset=['id','zipcode','grade'])
C. df[df.duplicated(['id','zipcode','grade’])]
D. All of the above

18. Which of the following code will help to display the 3rd, 4th and 5th rows from the 6th to 9th
columns of data frame data?
A. data.loc[3:6, 6:10]
B. data.iloc[3;6,6;10]
C. data.iloc[3:6,6:10]
D. None of the above

19. Which of the following syntax will display the last two records of df?
import pandas as pd
df = pd.DataFrame({'A':[34, 78, 54], 'B':[12, 67, 43]}, index=['r1', 'r2', 'r3'])
A. df.iloc[:'r3']
B. df.loc['r2':'r3']
C. df.iloc['r2':'r3']
D. df.loc[:'r3']

20. Which of the following is/are true about loc in pandas:
A. Add new rows in the data frames
B. To change the values of a row to a particular value
C. To extract values from the particular rows
D. All of the above

21. Change the date column in the format (dd/mm/year) using the pandas to_datetime() Function.
A. pd.to_datetime(data['date'], format='%Y-%m-%d', utc=False, dayfirst=True)
B. pd.to_datetime(data['date'])
C. pd.to_datetime(data['dates'], format='%Y-%m-%d', utc=False, dayfirst=True)
D. pd.to_datetime(data['date'])

22. Create a separate data frame that satisfies the conditions below.
1. Houses built before 1980
2. Have more than 2 bedrooms
3. Have more than 2 floors.

A. data.loc[(data["yr_built"] > 1980) & (data['floors'] > 2) & (data['bedrooms'] > 2)]
B. data.loc[(data["yr_built"] < 1980) & (data['floors'] > 2) & (data['bedrooms'] > 2)]
C. data.loc[(data["yr_built"] | 1980) & (data['floors'] > 2) | (data['bedrooms'] > 2)]
D. data.loc[(data["yr_built"] < 1980) | (data['floors'] > 2) | (data['bedrooms'] < 2)]

23. For a given nested list, convert the same into a dataframe.
sample_list = [['Carl', 22],
['Martha', 25],
['Calvin', 12],
['Stuart', 15]
]
The resulting dataframe must contain the column names as ‘Name’, and ‘Age’ with the respective values from the sample_list.

A. pd.DataFrame(sample_list)
B. pd.DataFrame(sample_list, column_names=['Name', 'Age'])
C. pd.DataFrame(sample_list, columns=['Name', 'Age'])
D. pd.DataFrame(sample_list, column_name=['Name', 'Age'])

24. For a given dictionary, convert the same into a dataframe.
```
sample_dict = {'Cristiano': ['Ronaldo','Man U', 801],
          'Lionel': ['Messi','PSG', 758],
          'Luis': ['Suarez','Atletico Madrid', 509],
          'Robert': ['Lewandowski','Bayern Munich', 527],
          'Zlatan': ['Ibrahimovic','AC Milan',553]
 }
```
A.
```
df1 = pd.Dataframe(sample_dict)
df1 = df1.transpose()
df1.reset_index(inplace = True)
df1.columns = ['First Name','Last Name', 'Club', 'Goals']
```

B.
```
df1 = pd.DataFrame(sample_dict)
df1 = df1.transpose()
df1.reset_index(inplace = True)
df1.columns = ['First Name','Last Name', 'Club', 'Goals']
```

C.
```
df1 = pd.DataFrame(sample_dict)
df1.reset_index(inplace = True)
df1.columns = ['First Name','Last Name', 'Club', 'Goals']
```

D.
```
df1 = pd.DataFrame(sample_dict)
df1 = df1.transpose()
df1.columns = ['First Name','Last Name', 'Club', 'Goals']
```

25. For a given tuple, convert the same into a dataframe.
sample_tuple = ([1, 'one', 3],
[2, 'two', 3],
[3, 'Three', 5],
[4, 'Four', 4],
[5, 'Five', 4])

A. pd.Dataframe(sample_tuple, columns=['Number', 'Number_text', 'txtlen'])
B. pd.DataFrame(sample_tuple, columns=['Number', 'Number_text', 'txtlen'])
C. pd.dataframe(sample_tuple, columns=['Number', 'Number_text', 'txtlen'])
D. pd.DataFrame(sample_tup, columns=['Number', 'Number_text', 'txtlen'])

26. Create a separate dataframe that contains houses ordered in ascending or descending order of the prices of each house.
A. ascending = housing.sort_values('price', ascending=False)
descending = housing.sort_values('price', ascending=False)
B. ascending = housing.sort_values('price', ascending=True)
descending = housing.sort_values('price', ascending=False)
C. ascending = housing.sort_values('price', ascending=False)
descending = housing.sort_values('price', ascending=True)
D. ascending = housing.sort_values('price', ascending=True)
descending = housing.sort_values('price', ascending=True)

27. Calculate the mean and standard deviation of all the numerical values in the dataset. For example - the mean for the bedrooms column is 3.370 and the standard deviation is 0.930.
A. housing.describe()
B. housing.info()
C. housing.corr()
D. housing.std()

28. Perform the following operations on the pandas dataframe.
1. Get the data starting from the rows 25 to 35.
2. Get the price of the houses located at the longitude--122.045 latitude- 47.6168.
A. data[25:35], data.loc[(housing["lat"] == 47.6168) | (data["long"] == -122.045)]
B. data[25:35], data.loc[(housing["lat"] == 48.6168) & (data["long"] == -122.045)]
C. data[25:36], data.loc[(housing["lat"] == 47.6168) & (data["long"] == -122.045)]
D. data[24:36], data.loc[(housing["lat"] == 47.6168) & (data["long"] == -122.045)]

29. Create a new column with the floor area(sqft_living, sqft_lot, sqft_above, sqft_basement, all combined in one column).
A. data['Floor Area'] = data['sqft_living'], data['sqft_lot'], data['sqft_basement'], data['sqft_above']
B. df['Floor Area'] = data['sqft_living'] + data['sqft_lot'] + data['sqft_basement'] + data['sqft_above']
C. data['Floor Area'] = data['sqft_living', 'sqft_lot','sqft_basement','sqft_above']
D. data['Floor Area'] = data['sqft_liv', 'sqft_lot','sqft_base','sqft_above']

30. Perform the following operations on the dataframes given below.
A = pd.DataFrame([['Carl', 22],['Martha', 25],['Calvin', 12],['Stuart', 15]], columns=['Name', 'Age'])
B = pd.DataFrame([['Melvin', 25],['Martha', 34],['Lewis', 32],['Leo', 25]], columns=['Name', 'Age'])
1. Left Outer Join
2. Outer Join
3. Inner Join
4. Right Outer Join

A.
```
inner = pd.merge(A, B, on='Age', how='inner_join')
outer = pd.merge(A, B, on='Age', how='outer_join')
left_outer = pd.merge(A, B, on='Age', how='left_join')
right_outer = pd.merge(A, B, on='Age', how='right_join')
```

B.
```
inner = pd.merge(A, B, on='Age', how='inner')
outer = pd.merge(A, B, on='Age', how='outer')
left_outer = pd.merge(A, B, on='Age', how='left_outer')
right_outer = pd.merge(A, B, on='Age', how='right_outer')
```

C.
```
inner = pd.merge(A, B, on='Age', how='inner')
outer = pd.merge(A, B, on='Age', how='outer')
left_outer = pd.merge(A, B, on='Age', how='left')
right_outer = pd.merge(A, B, on='Age', how='right')
```

D.
```
inner = pd.merge(A, B, on='Age', how='inner')
outer = pd.merge(A, B, on='Age', how='outer')
left_outer = pd.merge(A, B, on='Age', how='outer_left')
right_outer = pd.merge(A, B, on='Age', how='outer_right')
```

31. What will be the correlation between the columns sqft_living and sqft_above?
A. 0.702035
B. 0.754665
C. 0.876597
D. 0.303093

32. If the correlation between the columns bathrooms and sqft_living is 0.754665, what all interpretations can be made about the two columns?
A. A positive correlation between the two columns
B. The columns show perfect correlation.
C. A strong negative correlation between the two columns
D. No correlation between the two columns.