NPTEL - PYTHON FOR DATA SCIENCE
ASSIGNMENT - 3

Types of questions:

- **MCQs - Multiple Choice Questions (a question has only one correct answer)**
- **MSQs –**
  - **Multiple Select Questions (a question can have two, three or four correct options)**
  - **In this case, equal weightage must be given to all options**

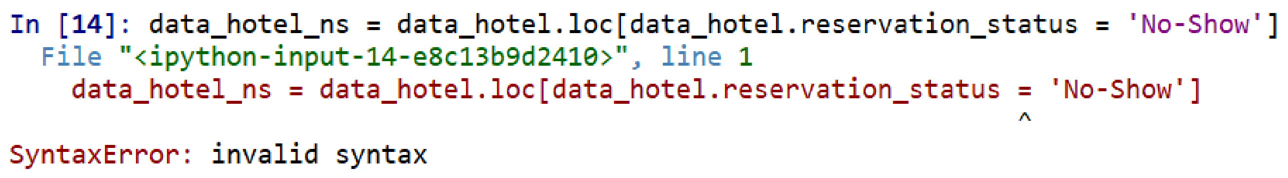
Read the comma-separated values file **hotel_bookings.csv** as a dataframe ***data_hotel*** and answer questions **1 - 3.** Please refer to ***Hotel_Bookings_Data_Description.pdf*** *for* data and variable description.

1. Choose the appropriate command(s) to filter those booking details whose ***reservation_status*** are a **No-show**?

a.
```
data_hotel_ns = data_hotel.loc[data_hotel.reservation_status = 'No-Show']
```
b.
```
data_hotel_ns = data_hotel[data_hotel.reservation_status == 'No-Show']
```
c.
```
data_hotel_ns = data_hotel.reservation_status.loc[data_hotel.isin(['No-Show'])]
```
d.
```
data_hotel_ns = data_hotel.loc[data_hotel.reservation_status.isin(['No-Show'])]
```

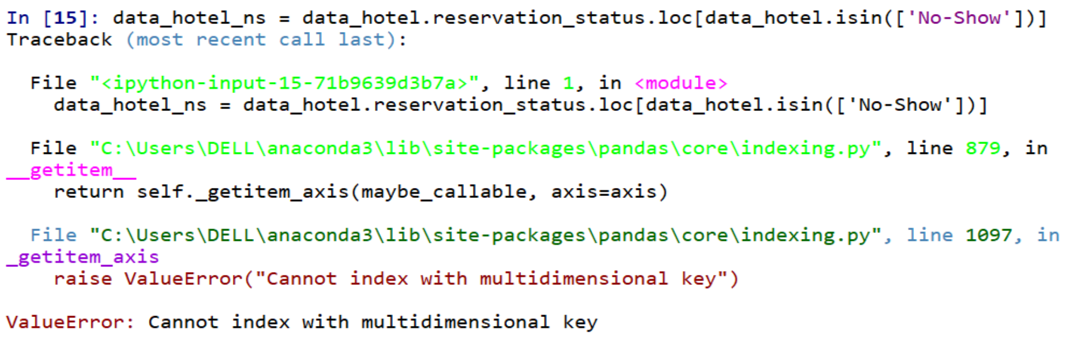
Answers: b) and d)

Option a) errors out with an invalid syntax. The equal to symbol is supposed to be ==.



```
In [14]: data_hotel_ns = data_hotel.loc[data_hotel.reservation_status = 'No-Show']
  File "<ipython-input-14-e8c13b9d2410>", line 1
    data_hotel_ns = data_hotel.loc[data_hotel.reservation_status = 'No-Show']
                                                                  ^
SyntaxError: invalid syntax
```

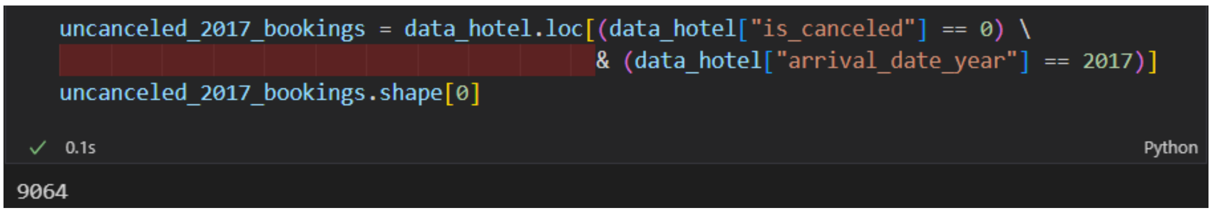
Option c) errors out as a ValueError. This is because **data_hotel.isin([‘No-Show’])** returns a DataFrame which is multidimensional, and a Series (one-dimensional data, **data_hotel.reservation_status**) is indexed based on the same. The correct statement is shown in Option d).



```
In [15]: data_hotel_ns = data_hotel.reservation_status.loc[data_hotel.isin(['No-Show'])]
Traceback (most recent call last):

  File "<ipython-input-15-71b9639d3b7a>", line 1, in <module>
    data_hotel_ns = data_hotel.reservation_status.loc[data_hotel.isin(['No-Show'])]

  File "C:\Users\DELL\anaconda3\lib\site-packages\pandas\core\indexing.py", line 879, in
__getitem__
    return self._getitem_axis(maybe_callable, axis=axis)

  File "C:\Users\DELL\anaconda3\lib\site-packages\pandas\core\indexing.py", line 1097, in
_getitem_axis
    raise ValueError("Cannot index with multidimensional key")

ValueError: Cannot index with multidimensional key
```

2. From the same data, find how many bookings were **not canceled** in the year **2017**?
   a. 9064
   b. 6231
   c. 9046
   d. None of the above

Answers: a)



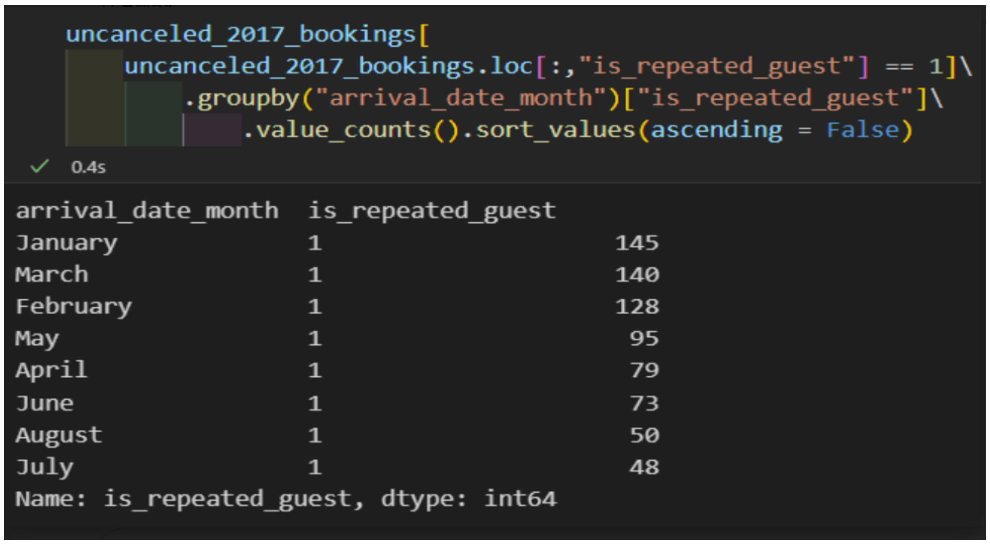
```
uncanceled_2017_bookings = data_hotel.loc[(data_hotel["is_canceled"] == 0) \
                                           & (data_hotel["arrival_date_year"] == 2017)]
uncanceled_2017_bookings.shape[0]
```

```
✓ 0.1s                                                                    Python
9064
```

Option b) is the count of records for the year 2015.

3. From the total bookings that were made in **2017** and **not canceled**, which month had the highest number of repeated guests?
   a. July
   b. February
   c. January
   d. None of the above

Answers: c)



```
uncanceled_2017_bookings[
    uncanceled_2017_bookings.loc[:,"is_repeated_guest"] == 1]\
        .groupby("arrival_date_month")["is_repeated_guest"]\
            .value_counts().sort_values(ascending = False)
```

```
✓ 0.4s
```

```
arrival_date_month  is_repeated_guest
January             1                    145
March               1                    140
February            1                    128
May                 1                     95
April               1                     79
June                1                     73
August              1                     50
July                1                     48
Name: is_repeated_guest, dtype: int64
```

Read the *'flavors_of_cocoa.csv'* file as a dataframe *'dt_cocoa'* and answer questions 4-7.

Variable	**Description**
Id	Serial no.
Company	Name of a manufacturing company
Bean Origin	Place of origin of cocoa bean
Review Data	Year in which chocolates were rated
Cocoa Percent	Percentage of cocoa in chocolate
Company Location	Location of a manufacturing company
Rating	Ratings of the chocolates

4. Which of the following commands can be used to create a variable **Flag**, and set the values as **Premium** when the *rating* is **equal to or greater than 3.25**, and otherwise as **Regular**?

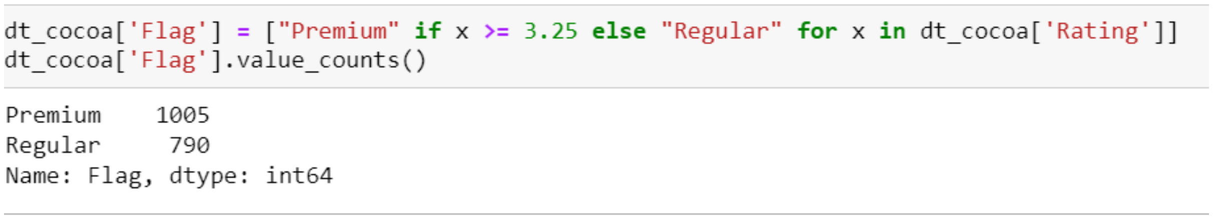
a. `dt_cocoa['Flag'] = ["Premium" if x > 3.25 else "Regular" for x in dt_cocoa['Rating']]`

b. `dt_cocoa['Flag'] = ["Premium" if x >= 3.25 else "Regular" for x in dt_cocoa['Rating']]`

c. `dt_cocoa["Flag"] = np.where(dt_cocoa["Rating"] < 3.25, "Regular", "Premium")`

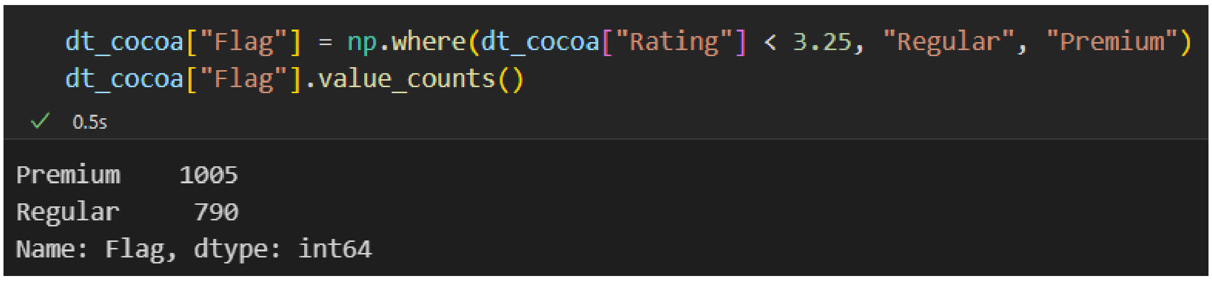
d. None of the above

Answers: b) and c)



```
dt_cocoa['Flag'] = ["Premium" if x >= 3.25 else "Regular" for x in dt_cocoa['Rating']]
dt_cocoa['Flag'].value_counts()
```

```
Premium    1005
Regular     790
Name: Flag, dtype: int64
```



```
dt_cocoa["Flag"] = np.where(dt_cocoa["Rating"] < 3.25, "Regular", "Premium")
dt_cocoa["Flag"].value_counts()
```
✓ 0.5s

```
Premium    1005
Regular     790
Name: Flag, dtype: int64
```

Option a) provides the following output without including chocolates that were given a rating of 3.25

```
dt_cocoa['Flag'] = ["Premium" if x > 3.25 else "Regular" for x in dt_cocoa['Rating']]
dt_cocoa['Flag'].value_counts()
```

```
Regular    1093
Premium     702
Name: Flag, dtype: int64
```

5. Which instruction can be used to impute the missing values in the column **Review Data** from the dataframe ***dt_cocoa*** by grouping the records company – wise?

a. `dt_cocoa['Review Date'] = dt_cocoa.groupby(['Company'])['Review Date'].apply(lambda x: x.fillna(x.mode().iloc[0]))`

b. `dt_cocoa['Review Date'] = dt_cocoa.groupby(['Company'])['Review Date'].apply(lambda x: x.fillna(x.mean()))`

c. `dt_cocoa['Review Date'] = dt_cocoa.groupby(['Company'])['Review Date'].apply(lambda x: x.fillna(x.mode()))`

d. None of the above

Answer a) imputes the missing value correctly.

Option c) doesn't impute the null values which can be checked by the below option:

```
dt_cocoa[np.isnan(dt_cocoa['Review Date'])]
```

	Id	Company	Bean Origin	Review Date	Cocoa Percent	Company Location	Rating
5	6	A. Morin	Carenero	NaN	70%	France	2.75
16	17	A. Morin	Papua New Guinea	NaN	70%	France	3.25
170	171	Bellflower	Kakao Kamili, Kilombero Valley	NaN	70%	U.S.A.	3.50
178	179	Benoit Nihant	Cuyagua Village	NaN	74%	Belgium	3.50

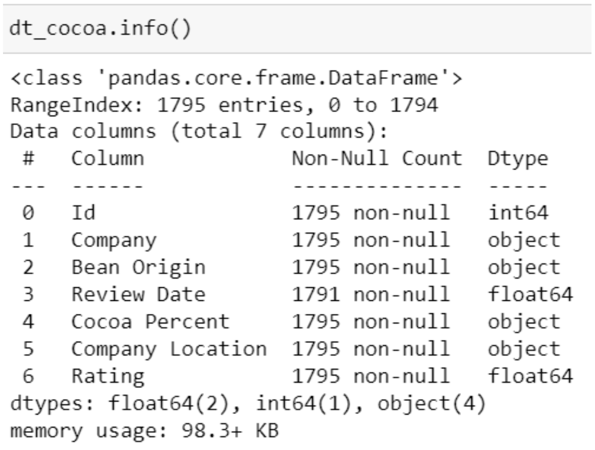
Option b) wrongly imputes the mean value of the column that has years as values.

178	179	Benoit Nihant	Cuyagua Village	2012.2	74%	Belgium	3.50

6. After checking the data summary, which feature requires a data conversion considering the data values held?
   a. Rating
   b. Review Date

c. Company
d. None of the above

Answer b) The type float64 is incorrect for the Review Date which holds years as values.



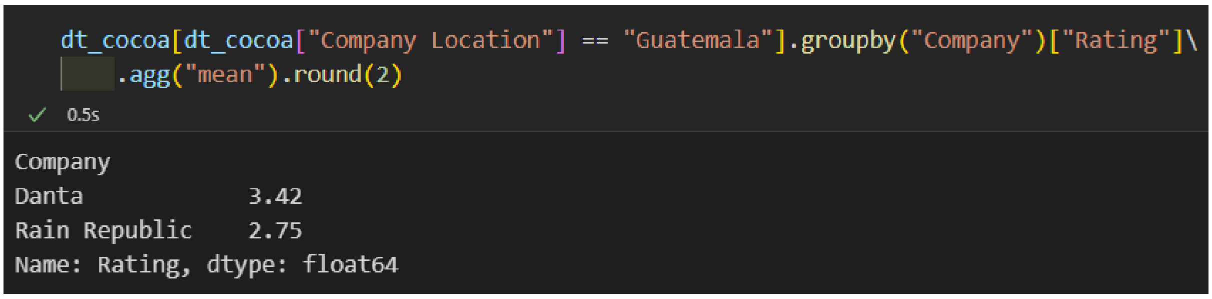
```
dt_cocoa.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1795 entries, 0 to 1794
Data columns (total 7 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Id                1795 non-null   int64  
 1   Company           1795 non-null   object 
 2   Bean Origin       1795 non-null   object 
 3   Review Date       1791 non-null   float64
 4   Cocoa Percent     1795 non-null   object 
 5   Company Location  1795 non-null   object 
 6   Rating            1795 non-null   float64
dtypes: float64(2), int64(1), object(4)
memory usage: 98.3+ KB
```

7. What is the maximum average rating for the cocoa companies based out of **Guatemala**?
   a. 4
   b. 3.5
   c. 3.42
   d. None of the above

Answer c)

The average of the cocoa rating across companies based out of Guatemala can be found by applying aggregation after group by on Company. The answer reveals which specific company produces better grade chocolate.



```
dt_cocoa[dt_cocoa["Company Location"] == "Guatemala"].groupby("Company")["Rating"]\
    .agg("mean").round(2)
```

✓ 0.5s

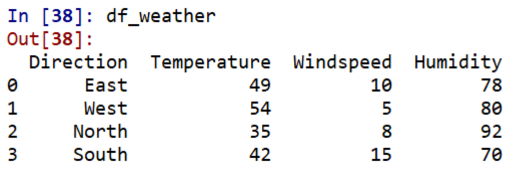
```
Company
Danta            3.42
Rain Republic    2.75
Name: Rating, dtype: float64
```

8. Which pandas function is used to stack the dataframes vertically?
   a. pd.merge()
   b. pd.concat()
   c. join()
   d. None of the above

Answer: b)

Using pd.merge(), dataframes can be combined on the basis of common columns. Using pd.concat(), 2 dataframes can be simply stacked either horizontally or vertically.

Assume a pandas dataframe ***df_weather*** which when printed is as shown below:



```
In [38]: df_weather
Out[38]:
  Direction  Temperature  Windspeed  Humidity
0      East           49         10        78
1      West           54          5        80
2     North           35          8        92
3     South           42         15        70
```

9. Of the following set of statements, which of them can be used to extract the column **Direction** as a separate dataframe?

   a. `df_weather[['Direction']]`

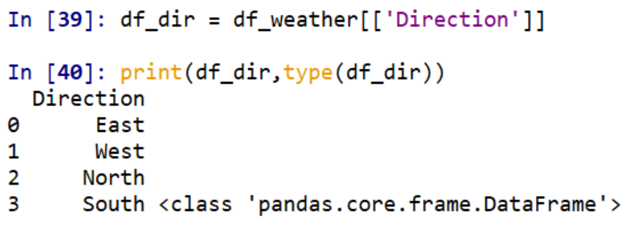
   b. `df_weather.iloc[:,0]`

   c. `df_weather.loc[:,['Direction']]`

   d. None of the above

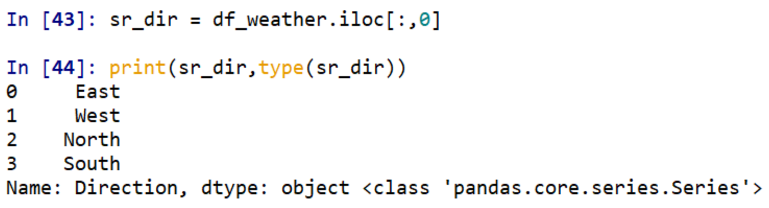
Answer: a and c.

Option a. ->



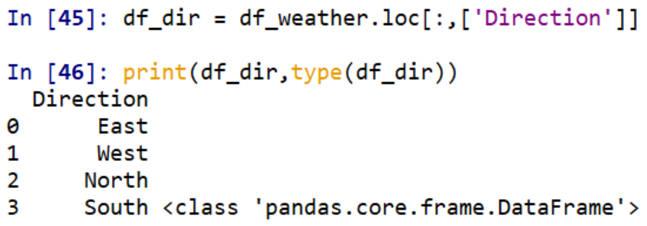
```
In [39]: df_dir = df_weather[['Direction']]

In [40]: print(df_dir,type(df_dir))
  Direction
0      East
1      West
2     North
3     South <class 'pandas.core.frame.DataFrame'>
```

Option b ->



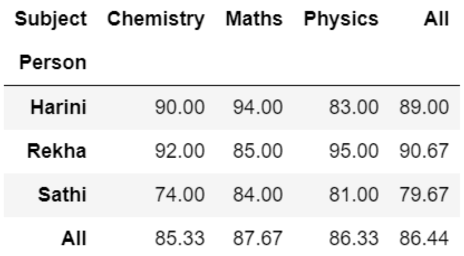
```
In [43]: sr_dir = df_weather.iloc[:,0]

In [44]: print(sr_dir,type(sr_dir))
0     East
1     West
2    North
3    South
Name: Direction, dtype: object <class 'pandas.core.series.Series'>
```

Option c ->



```
In [45]: df_dir = df_weather.loc[:,['Direction']]

In [46]: print(df_dir,type(df_dir))
  Direction
0      East
1      West
2     North
3     South <class 'pandas.core.frame.DataFrame'>
```

10. A file **“Students.csv”** contains the attendance and scores of three separate students. This dataset is loaded into a dataframe ***df_study*** and a cross table is obtained from the same which results in the following output



Subject	Chemistry	Maths	Physics	All
Person				
Harini	90.00	94.00	83.00	89.00
Rekha	92.00	85.00	95.00	90.67
Sathi	74.00	84.00	81.00	79.67
All	85.33	87.67	86.33	86.44

Which one of these students’ average score across all subjects was the lowest? Which subject has the highest average score across students?

a. Harini, Maths
b. Sathi, Maths
c. Harini, Physics
d. Rekha, Maths

Answer: b) Sathi, Maths