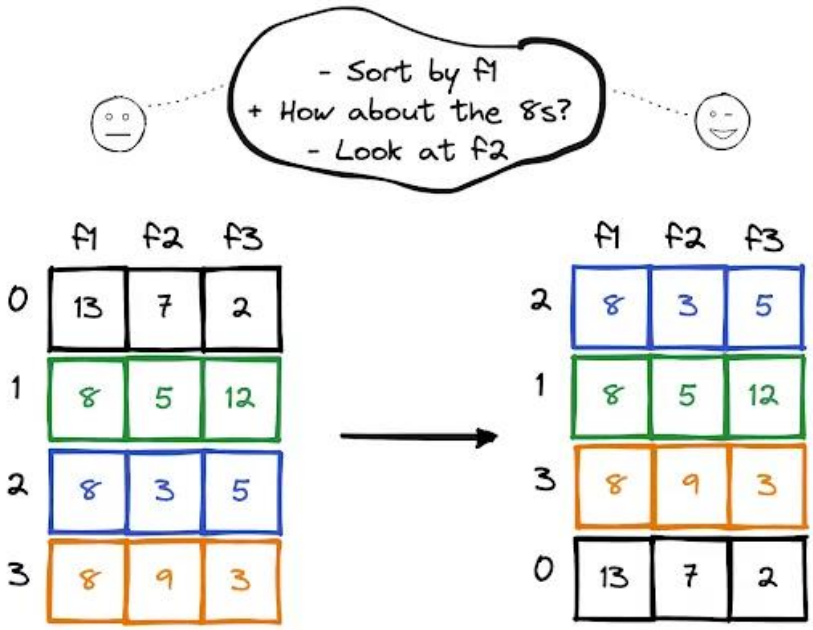

Pandas functions & missing values - Home Exercises

Exercise 1:

1. Use the following dataframe:
   df = pd.DataFrame({
   'CustomerID': np.arange(1, 11),
   'FirstName': ['John', 'Steve', 'Anna', 'Mike', 'Emily', 'Jake', 'Laura', 'Nick', 'Olivia', 'David'],

   'LastName': ['Smith', 'Johnson', 'Williams', 'Jones', 'Brown', 'Davis', 'Miller', 'Wilson', 'Moore', 'Taylor'],

   'Age': np.random.randint(25, 65, size=10),
   'TotalPurchaseAmount': np.random.randint(100, 1000, size=10)
   })

2. Create a new column could ‘classification’ and classify the customers based on their `TotalPurchaseAmount`. If the purchase amount is less than 500 it's 'Low', otherwise it's 'High'.

3. Using the column with the previous exercise create a new column could “classification_by_age’ and put values according to the following logic:
   If the classification ‘Low’ and the customer age is below 20 put ‘Low_Young’
   If the classification ‘HIGH’’ and the customer age is above 20 put ‘HIGH_Old’
   If the classification ‘Low’ and the customer age is above 20 put ‘Low_Old’
   If the classification ‘High’ and the customer age is below 20 put ‘High_Young’

Exercise 2:

1. Use the df from the previous exercise
2. Sort the DataFrame by the customer name from A-Z order
3. Sort the DataFrame first by `TotalPurchaseAmount` in ascending order, and then by `Age` in descending order for rows having the same `TotalPurchaseAmount`.

Exercise 3:

1. Run the following code to create the df:
   ```
   np.random.seed(100)
   df = pd.DataFrame(np.random.randint(0,100,size=(20, 10)),
   columns=list('ABCDEFGHIJ'))
   n_rows, n_cols = df.shape
   row_idx = np.random.randint(0, n_rows)
   col_idxs = np.random.choice(n_cols, size=n_cols//2, replace=False)
   df.iloc[row_idx, col_idxs] = np.nan
   n_nan_remaining = 15 - len(col_idxs)
   row_idxs = np.random.randint(0, n_rows, size=n_nan_remaining)
   col_idxs = np.random.randint(0, n_cols, size=n_nan_remaining)
   df.iloc[row_idxs, col_idxs] = np.nan
   ```
2. In any exercise create a copy from the original df
3. Drop all the rows containing any NaN values
4. Fill the NaN values using a random value between 0-100
5. Fill the NaN values in the C and D columns with the mean value of the columns
6. Drop the NaN values for all columns with at least 2 NaN's in all columns beside the 'D' column

את שיעורי הבית יש לשלוח ל- pythonai170624+HW28@gmail.com

יש לעלות את קובץ ה- ipynb ל- Github