

TASK 1

- *Installing Pandas: Python Data Analysis Library*

```
[1]: !pip install pandas

Requirement already satisfied: pandas in c:\users\sanaj\appdata\local\programs\python\python312\lib\site-packages (2.2.3)
Requirement already satisfied: numpy>=1.26.0 in c:\users\sanaj\appdata\local\programs\python\python312\lib\site-packages (from pandas) (2.2.1)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\sanaj\appdata\local\programs\python\python312\lib\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\sanaj\appdata\local\programs\python\python312\lib\site-packages (from pandas) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in c:\users\sanaj\appdata\local\programs\python\python312\lib\site-packages (from pandas) (2024.2)
Requirement already satisfied: six>=1.5 in c:\users\sanaj\appdata\local\programs\python\python312\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)

[notice] A new release of pip is available: 24.0 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

- *Importing the pandas module*

Pandas provides high-performance data structures and data analysis tools. The main feature of Pandas is a fast and efficient Data Frame object for data manipulation with integrated indexing. The structure of Data Frame can be seen as a spreadsheet which offers very flexible ways of working with it. Pandas offers features like adding or removing rows and columns and reshaping it the way you want. It also provides many high- performance functions such as aggregation , merging , and joining datasets. Pandas tools for exporting and importing data from different formats file like .CSV , text files , Microsoft Excel , SQL database etc.

```
[2]: import pandas as pd
```

- *Loading and Reading the dataset*

```
[3]: cpa=pd.read_csv('C:/intern/Day 1/Customr Personality Analysis Dataset.csv',
na_values='.',
usecols=["ID","Year_Birth","Education","Marital_Status","Income","Kidhome","Teenhome","Dt_Customer","Recency"])
```

```
[4]: cpa
```

```
[4]:
```

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency
0	5524	1957	Graduation	Single	58138.0	0	0.0	2012-09-04	58.0
1	2174	1954	Graduation	Single	46344.0	1	1.0	2014-03-08	38.0
2	4141	1965	Graduation	Together	71613.0	0	0.0	2013-08-21	26.0
3	6182	1984	Graduation	Together	26646.0	1	0.0	2014-02-10	26.0
4	5324	1981	PhD	Married	58293.0	1	0.0	2014-01-19	94.0
...
2438	10870	1967	Graduation	Married	61223.0	0	1.0	2013-06-13	46.0
2439	4001	1946	PhD	Together	64014.0	2	1.0	2014-06-10	56.0
2440	7270	1981	Graduation	Divorced	56981.0	0	0.0	2014-01-25	91.0
2441	8235	1956	Master	Together	69245.0	0	1.0	2014-01-24	8.0
2442	9405	1954	PhD	Married	52869.0	1	1.0	2012-10-15	40.0

2443 rows × 9 columns

First of all, I have created a new notebook as I want. Here , I have created a notebook with the name ‘intern’ then I have downloaded the data file and stored it in the same directory to retrieve the same. The above code has to be written to read the .csv file in *pandas* is by using *read_csv* method with the file path , *na_value* parameter is used to tell pandas they consider “:” as NaN value print NaN at the place of “:” and the other method which is *usecols* are used to tell the pandas that which cols are we going to use.

The output of the execution shows that ad DataFrame size is 2443 rows x 9 columns.

- *Filtering and handling the null values*

```
[7]: cpa[cpa['Education'].isnull()]
```

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency
76	Cycle	Married	NaN	1	0.0	2012-09-14	25.0	25	3.0
1511	Cycle	Single	NaN	1	1.0	2014-01-12	49.0	5	1.0
2431	Cycle	Together	NaN	0	0.0	2012-08-12	53.0	32	2.0

```
[10]: cpa['Education'] = cpa['Education'].fillna('Graduation')
```

```
[11]: cpa[cpa['Education'].isnull()]
```

ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency
----	------------	-----------	----------------	--------	---------	----------	-------------	---------

```
[12]: cpa['Marital_Status'] = cpa['Marital_Status'].replace({
    'YOLO': 'Single',
    'Absurd': 'Single',
    'Alone': 'Single'
})
```

```
[13]: cpa['Marital_Status'] = cpa['Marital_Status'].fillna('Married')
```

```
[14]: cpa[cpa['Marital_Status'].isnull()]
```

ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency
----	------------	-----------	----------------	--------	---------	----------	-------------	---------

Internship (Day 1)

```
[17]: cpa=cpa.dropna(how='any',subset=["Dt_Customer"])
```

```
[18]: cpa
```

```
[18]:
```

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency
0	5524	1957	Graduation	Single	58138.0	0	0.0	2012-09-04	58.0
1	2174	1954	Graduation	Single	46344.0	1	1.0	2014-03-08	38.0
2	4141	1965	Graduation	Together	71613.0	0	0.0	2013-08-21	26.0
3	6182	1984	Graduation	Together	26646.0	1	0.0	2014-02-10	26.0
4	5324	1981	PhD	Married	58293.0	1	0.0	2014-01-19	94.0
...
2438	10870	1967	Graduation	Married	61223.0	0	1.0	2013-06-13	46.0
2439	4001	1946	PhD	Together	64014.0	2	1.0	2014-06-10	56.0
2440	7270	1981	Graduation	Divorced	56981.0	0	0.0	2014-01-25	91.0
2441	8235	1956	Master	Together	69245.0	0	1.0	2014-01-24	8.0
2442	9405	1954	PhD	Married	52869.0	1	1.0	2012-10-15	40.0

2240 rows × 9 columns

```
[19]: cpa['Income'] =cpa['Income'].fillna(cpa['Income'].mean())
```

C:\Users\sanaj\AppData\Local\Temp\ipykernel_3380\1220748429.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.
cpa['Income'] =cpa['Income'].fillna(cpa['Income'].mean())

```
[20]: cpa[cpa['Income'].isnull()]
```

```
[20]:
```

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency
--	----	------------	-----------	----------------	--------	---------	----------	-------------	---------

```
[21]: cpa[cpa['Teenhome'].isnull()]
```

```
[21]:
```

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency
--	----	------------	-----------	----------------	--------	---------	----------	-------------	---------

```
[23]: cpa[cpa['Kidhome'].isnull()]
```

```
[23]:
```

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency
--	----	------------	-----------	----------------	--------	---------	----------	-------------	---------

```
[24]: cpa[cpa['Recency'].isnull()]
```

```
[24]:
```

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency
--	----	------------	-----------	----------------	--------	---------	----------	-------------	---------

```
[25]: cpa[cpa['Dt_Customer'].isnull()]
```

```
[25]:
```

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency
--	----	------------	-----------	----------------	--------	---------	----------	-------------	---------

Internship (Day 1)

- *Dropping or removing the duplicate values*

```
[26]: cpa=cpa.drop_duplicates()
```

```
[27]: cpa
```

```
[27]:
```

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency
0	5524	1957	Graduation	Single	58138.0	0	0.0	2012-09-04	58.0
1	2174	1954	Graduation	Single	46344.0	1	1.0	2014-03-08	38.0
2	4141	1965	Graduation	Together	71613.0	0	0.0	2013-08-21	26.0
3	6182	1984	Graduation	Together	26646.0	1	0.0	2014-02-10	26.0
4	5324	1981	PhD	Married	58293.0	1	0.0	2014-01-19	94.0
...
2438	10870	1967	Graduation	Married	61223.0	0	1.0	2013-06-13	46.0
2439	4001	1946	PhD	Together	64014.0	2	1.0	2014-06-10	56.0
2440	7270	1981	Graduation	Divorced	56981.0	0	0.0	2014-01-25	91.0
2441	8235	1956	Master	Together	69245.0	0	1.0	2014-01-24	8.0
2442	9405	1954	PhD	Married	52869.0	1	1.0	2012-10-15	40.0

2223 rows × 9 columns

- *Checking is there any null value remaining*

```
[28]: missing_values = cpa.isnull().sum()  
print(missing_values)
```

```
ID          0  
Year_Birth  0  
Education   0  
Marital_Status  0  
Income      0  
Kidhome     0  
Teenhome    0  
Dt_Customer  0  
Recency     0  
dtype: int64
```

Internship (Day 1)

- *Renaming column headers to be clean and uniform*

```
[31]: cpa=cpa.rename(columns={"ID":"customer_id", "Year_Birth":"year_birth", "Education":"education", "Marital_Status":"marital_status",  
                             "Income":"income", "Kidhome":"kidhome", "Teenhome":"teenhome", "Dt_Customer":"dt_customer", "Recency":"recency"})
```

```
[32]: cpa
```

```
[32]:
```

	customer_id	year_birth	education	marital_status	income	kidhome	teenhome	dt_customer	recency
0	5524	1957	Graduation	Single	58138.0	0	0.0	2012-09-04	58.0
1	2174	1954	Graduation	Single	46344.0	1	1.0	2014-03-08	38.0
2	4141	1965	Graduation	Together	71613.0	0	0.0	2013-08-21	26.0
3	6182	1984	Graduation	Together	26646.0	1	0.0	2014-02-10	26.0
4	5324	1981	PhD	Married	58293.0	1	0.0	2014-01-19	94.0
...
2438	10870	1967	Graduation	Married	61223.0	0	1.0	2013-06-13	46.0
2439	4001	1946	PhD	Together	64014.0	2	1.0	2014-06-10	56.0
2440	7270	1981	Graduation	Divorced	56981.0	0	0.0	2014-01-25	91.0
2441	8235	1956	Master	Together	69245.0	0	1.0	2014-01-24	8.0
2442	9405	1954	PhD	Married	52869.0	1	1.0	2012-10-15	40.0

2223 rows × 9 columns

- *Converting data in date format and removing invalid data after conversion*

```
[33]: cpa['dt_customer'] = pd.to_datetime(cpa['dt_customer'], errors='coerce')
```

```
[35]: cpa = cpa[cpa['dt_customer'].notnull()]
```

```
[36]: cpa
```

```
[36]:
```

	customer_id	year_birth	education	marital_status	income	kidhome	teenhome	dt_customer	recency
0	5524	1957	Graduation	Single	58138.0	0	0.0	2012-09-04	58.0
1	2174	1954	Graduation	Single	46344.0	1	1.0	2014-03-08	38.0
2	4141	1965	Graduation	Together	71613.0	0	0.0	2013-08-21	26.0
3	6182	1984	Graduation	Together	26646.0	1	0.0	2014-02-10	26.0
4	5324	1981	PhD	Married	58293.0	1	0.0	2014-01-19	94.0
...
2438	10870	1967	Graduation	Married	61223.0	0	1.0	2013-06-13	46.0
2439	4001	1946	PhD	Together	64014.0	2	1.0	2014-06-10	56.0
2440	7270	1981	Graduation	Divorced	56981.0	0	0.0	2014-01-25	91.0
2441	8235	1956	Master	Together	69245.0	0	1.0	2014-01-24	8.0
2442	9405	1954	PhD	Married	52869.0	1	1.0	2012-10-15	40.0

2037 rows × 9 columns

- *Ensuring correct datatypes*

```
[38]: cpa['customer_id'] = cpa['customer_id'].astype(int)  
cpa['year_birth'] = cpa['year_birth'].astype(int)  
cpa['education'] = cpa['education'].astype(str)  
cpa['marital_status'] = cpa['marital_status'].astype(str)  
cpa['income'] = cpa['income'].astype(float)  
cpa['recency'] = cpa['recency'].astype(int)  
cpa['kidhome'] = cpa['kidhome'].astype(int)  
cpa['teenhome'] = cpa['teenhome'].astype(int)
```

Internship (Day 1)

[39]:

cpa

[39]:

	customer_id	year_birth	education	marital_status	income	kidhome	teenhome	dt_customer	recency
0	5524	1957	Graduation	Single	58138.0	0	0	2012-09-04	58
1	2174	1954	Graduation	Single	46344.0	1	1	2014-03-08	38
2	4141	1965	Graduation	Together	71613.0	0	0	2013-08-21	26
3	6182	1984	Graduation	Together	26646.0	1	0	2014-02-10	26
4	5324	1981	PhD	Married	58293.0	1	0	2014-01-19	94
...
2438	10870	1967	Graduation	Married	61223.0	0	1	2013-06-13	46
2439	4001	1946	PhD	Together	64014.0	2	1	2014-06-10	56
2440	7270	1981	Graduation	Divorced	56981.0	0	0	2014-01-25	91
2441	8235	1956	Master	Together	69245.0	0	1	2014-01-24	8
2442	9405	1954	PhD	Married	52869.0	1	1	2012-10-15	40

2037 rows × 9 columns

- *Convert 'dt_customer' to consistent date format (dd-mm-yyyy)*

[40]:

```
cpa['dt_customer'] = cpa['dt_customer'].dt.strftime('%d-%m-%Y')
|
```

[41]:

cpa

[41]:

	customer_id	year_birth	education	marital_status	income	kidhome	teenhome	dt_customer	recency
0	5524	1957	Graduation	Single	58138.0	0	0	04-09-2012	58
1	2174	1954	Graduation	Single	46344.0	1	1	08-03-2014	38
2	4141	1965	Graduation	Together	71613.0	0	0	21-08-2013	26
3	6182	1984	Graduation	Together	26646.0	1	0	10-02-2014	26
4	5324	1981	PhD	Married	58293.0	1	0	19-01-2014	94
...
2438	10870	1967	Graduation	Married	61223.0	0	1	13-06-2013	46
2439	4001	1946	PhD	Together	64014.0	2	1	10-06-2014	56
2440	7270	1981	Graduation	Divorced	56981.0	0	0	25-01-2014	91
2441	8235	1956	Master	Together	69245.0	0	1	24-01-2014	8
2442	9405	1954	PhD	Married	52869.0	1	1	15-10-2012	40

2037 rows × 9 columns

- *Standardize the text value*

[42]:

```
cpa['education'] = cpa['education'].astype(str).str.lower()
cpa['marital_status'] = cpa['marital_status'].astype(str).str.lower() |
```

Internship (Day 1)

[43]: cpa

[43]:

	customer_id	year_birth	education	marital_status	income	kidhome	teenhome	dt_customer	recency
0	5524	1957	graduation	single	58138.0	0	0	04-09-2012	58
1	2174	1954	graduation	single	46344.0	1	1	08-03-2014	38
2	4141	1965	graduation	together	71613.0	0	0	21-08-2013	26
3	6182	1984	graduation	together	26646.0	1	0	10-02-2014	26
4	5324	1981	phd	married	58293.0	1	0	19-01-2014	94
...
2438	10870	1967	graduation	married	61223.0	0	1	13-06-2013	46
2439	4001	1946	phd	together	64014.0	2	1	10-06-2014	56
2440	7270	1981	graduation	divorced	56981.0	0	0	25-01-2014	91
2441	8235	1956	master	together	69245.0	0	1	24-01-2014	8
2442	9405	1954	phd	married	52869.0	1	1	15-10-2012	40

2037 rows × 9 columns

- *Saved clean data*

[44]: cpa.to_csv("cleaned_customer_personality_analysis_data.csv", index=False)