Lab Assignment 26

Student Name: Chauhan Vandana Ramdayal

Student Id: AF0411629

Topic: Pandas IO and Cleaning Data

Pandas IO and Cleaning Data

In **Pandas**, the IO (Input/Output) capabilities allow you to read and write data from various file formats, while data cleaning involves transforming raw data into a usable format by handling missing values, duplicates, and incorrect data types.

Pandas IO Operations

Reading Data:

- **CSV**:
  df = pd.read_csv('file.csv')
- **Excel**:
  df = pd.read_excel('file.xlsx', sheet_name='Sheet1')
- **SQL**:
  df = pd.read_sql_query('SELECT * FROM table', conn)
- **JSON**:
  df = pd.read_json('file.json')

Writing Data:

- **To CSV**:
  df.to_csv('output.csv', index=False)
- **To Excel**:
  df.to_excel('output.xlsx', index=False)
- **To SQL**:
  df.to_sql('table_name', conn, if_exists='replace')

Data Cleaning Techniques

Handling Missing Data:

- **Check for missing values**:
  df.isnull().sum()
- **Fill missing values**:
  df['column'].fillna(value, inplace=True)
- **Drop rows with missing data**:
  df.dropna(inplace=True)

Handling Duplicates:

- **Find duplicates**:
  df.duplicated().sum()
- **Remove duplicates**:
  df.drop_duplicates(inplace=True)

Data Type Conversion:

- **Convert to numeric**:
  df['column'] = pd.to_numeric(df['column'], errors='coerce')
- **Convert to datetime**:
  df['date_column'] = pd.to_datetime(df['date_column'], errors='coerce')

String Cleaning:

- **Strip whitespace**:
  df['column'].str.strip()
- **Replace characters**:
  df['column'].str.replace('[^a-zA-Z]', '', regex=True)

Renaming Columns:

- **Rename columns**:
  df.rename(columns={'old_name': 'new_name'}, inplace=True)

Input

df = pd.DataFrame({
'ord_no':[np.nan,np.nan,70002,np.nan,np.nan,70005,np.nan,70010,70003,70012,np.nan,np.nan],
'purch_amt':[np.nan,270.65,65.26,np.nan,948.5,2400.6,5760,1983.43,2480.4,250.45, 75.29,np.nan],
'ord_date': [np.nan,'2012-09-10',np.nan,np.nan,'2012-09-10','2012-07-27','2012-09-10','2012-10-10','2012-10-10','2012-06-27','2012-08-17',np.nan],
'customer_id':[np.nan,3001,3001,np.nan,3002,3001,3001,3004,3003,3002,3001,np.nan]})
print("Original Orders DataFrame:")

Question:

Q1. Missing values in the given DataFrame - isna().sum()

Code:

```python
#1. Missing values in the given DataFrame - isna().sum()
import pandas as pd
import numpy as np
df = pd.DataFrame({
'ord_no':[70001,np.nan,70002,70004,np.nan,70005,np.nan,70010,70003,70012,np.nan,70013],
'purch_amt':[150.5,270.65,65.26,110.5,948.5,2400.6,5760,1983.43,2480.4,250.45, 75.29,3045.6],
'ord_date': ['2012-10-05','2012-09-10',np.nan,'2012-08-17','2012-09-10','2012-07-27','2012-09-10','2012-10-10','2012-10-
'customer_id':[3002,3001,3001,3003,3002,3001,3001,3004,3003,3002,3001,3001],
'salesman_id':[5002,5003,5001,np.nan,5002,5001,5001,np.nan,5003,5002,5003,np.nan]})
print("Original order datafram:")
print(df)
print("\n missing value of the given dataframe")
print(df.isna().sum())
```

Output:

```
Original order datafram:
    ord_no  purch_amt    ord_date  customer_id  salesman_id
0  70001.0     150.50  2012-10-05         3002       5002.0
1      NaN     270.65  2012-09-10         3001       5003.0
2  70002.0      65.26         NaN         3001       5001.0
3  70004.0     110.50  2012-08-17         3003          NaN
4      NaN     948.50  2012-09-10         3002       5002.0
5  70005.0    2400.60  2012-07-27         3001       5001.0
6      NaN    5760.00  2012-09-10         3001       5001.0
7  70010.0    1983.43  2012-10-10         3004          NaN
8  70003.0    2480.40  2012-10-10         3003       5003.0
9  70012.0     250.45  2012-06-27         3002       5002.0
10     NaN      75.29  2012-08-17         3001       5003.0
11 70013.0    3045.60  2012-04-25         3001          NaN

 missing value of the given dataframe
ord_no         4
purch_amt      0
ord_date       1
customer_id    0
salesman_id    3
dtype: int64
```

2. Drop rows in the DateFrame - dropna()

Code:

```python
#Q2. Drop rows in the DateFrame - dropna()
import pandas as pd
import numpy as np
df = pd.DataFrame({
'ord_no':[70001,np.nan,70002,70004,np.nan,70005,np.nan,70010,70003,70012,np.nan,70013],
'purch_amt':[150.5,270.65,65.26,110.5,948.5,2400.6,5760,1983.43,2480.4,250.45, 75.29,3045.6],
'ord_date': ['2012-10-05','2012-09-10',np.nan,'2012-08-17','2012-09-10','2012-07-27','2012-09-10','2012-10-10','2012-10-
'customer_id':[3002,3001,3001,3003,3002,3001,3001,3004,3003,3002,3001,3001],
'salesman_id':[5002,5003,5001,np.nan,5002,5001,5001,np.nan,5003,5002,5003,np.nan]})
print("Original order datafram:")
print(df)
print("\ndrop the row where atleast one element is missing:")
result=df.dropna()
print(result)
```

Output:

```
Original order datafram:
     ord_no  purch_amt    ord_date  customer_id  salesman_id
0   70001.0     150.50  2012-10-05         3002       5002.0
1       NaN     270.65  2012-09-10         3001       5003.0
2   70002.0      65.26         NaN         3001       5001.0
3   70004.0     110.50  2012-08-17         3003          NaN
4       NaN     948.50  2012-09-10         3002       5002.0
5   70005.0    2400.60  2012-07-27         3001       5001.0
6       NaN    5760.00  2012-09-10         3001       5001.0
7   70010.0    1983.43  2012-10-10         3004          NaN
8   70003.0    2480.40  2012-10-10         3003       5003.0
9   70012.0     250.45  2012-06-27         3002       5002.0
10      NaN      75.29  2012-08-17         3001       5003.0
11  70013.0    3045.60  2012-04-25         3001          NaN

drop the row where atleast one element is missing:
    ord_no  purch_amt    ord_date  customer_id  salesman_id
0  70001.0     150.50  2012-10-05         3002       5002.0
5  70005.0    2400.60  2012-07-27         3001       5001.0
8  70003.0    2480.40  2012-10-10         3003       5003.0
9  70012.0     250.45  2012-06-27         3002       5002.0
```

3 Drop Entire Rows in the DataFrame - dropna(how='all')

Code:

```
#Q3 Drop Entire Rows in the DataFrame - dropna(how='all')
import pandas as pd
import numpy as np
df = pd.DataFrame({
'ord_no':[70001,np.nan,70002,70004,np.nan,70005,np.nan,70010,70003,70012,np.nan,70013],
'purch_amt':[150.5,270.65,65.26,110.5,948.5,2400.6,5760,1983.43,2480.4,250.45, 75.29,3045.6],
'ord_date': ['2012-10-05','2012-09-10',np.nan,'2012-08-17','2012-09-10','2012-07-27','2012-09-10','2012-10-10','2012-10-
'customer_id':[3002,3001,3001,3003,3002,3001,3001,3004,3003,3002,3001,3001],
'salesman_id':[5002,5003,5001,np.nan,5002,5001,5001,np.nan,5003,5002,5003,np.nan]})
print("Original order datafram:")
print(df)
print("\ndrop the row where all element is missing:")
result=df.dropna(how='all')
print(result)
```

Output:

```
Original order datafram:
     ord_no  purch_amt    ord_date  customer_id  salesman_id
0   70001.0     150.50  2012-10-05         3002       5002.0
1       NaN     270.65  2012-09-10         3001       5003.0
2   70002.0      65.26         NaN         3001       5001.0
3   70004.0     110.50  2012-08-17         3003          NaN
4       NaN     948.50  2012-09-10         3002       5002.0
5   70005.0    2400.60  2012-07-27         3001       5001.0
6       NaN    5760.00  2012-09-10         3001       5001.0
7   70010.0    1983.43  2012-10-10         3004          NaN
8   70003.0    2480.40  2012-10-10         3003       5003.0
9   70012.0     250.45  2012-06-27         3002       5002.0
10      NaN      75.29  2012-08-17         3001       5003.0
11  70013.0    3045.60  2012-04-25         3001          NaN

drop the row where all element is missing:
     ord_no  purch_amt    ord_date  customer_id  salesman_id
0   70001.0     150.50  2012-10-05         3002       5002.0
1       NaN     270.65  2012-09-10         3001       5003.0
2   70002.0      65.26         NaN         3001       5001.0
3   70004.0     110.50  2012-08-17         3003          NaN
4       NaN     948.50  2012-09-10         3002       5002.0
5   70005.0    2400.60  2012-07-27         3001       5001.0
6       NaN    5760.00  2012-09-10         3001       5001.0
7   70010.0    1983.43  2012-10-10         3004          NaN
8   70003.0    2480.40  2012-10-10         3003       5003.0
9   70012.0     250.45  2012-06-27         3002       5002.0
10      NaN      75.29  2012-08-17         3001       5003.0
11  70013.0    3045.60  2012-04-25         3001          NaN
```

4. Drop Specific columns - NaN

Code:

```
#Q4.Drop Specific columns-NaN.
import pandas as pd
import numpy as np
df = pd.DataFrame({
'ord_no':[70001,np.nan,70002,70004,np.nan,70005,np.nan,70010,70003,70012,np.nan,70013],
'purch_amt':[150.5,270.65,65.26,110.5,948.5,2400.6,5760,1983.43,2480.4,250.45, 75.29,3045.6],
'ord_date': ['2012-10-05','2012-09-10',np.nan,'2012-08-17','2012-09-10','2012-07-27','2012-09-10','2012-10-10','2012-10
'customer_id':[3002,3001,3001,3003,3002,3001,3001,3004,3003,3002,3001,3001],
'salesman_id':[5002,5003,5001,np.nan,5002,5001,5001,np.nan,5003,5002,5003,np.nan]})
print("Original order datafram:")
print(df)
result=df.dropna(subset=['ord_no','customer_id'])
print(result)
```

Output:

```
Original order datafram:
    ord_no  purch_amt    ord_date  customer_id  salesman_id
0  70001.0     150.50  2012-10-05         3002       5002.0
1      NaN     270.65  2012-09-10         3001       5003.0
2  70002.0      65.26         NaN         3001       5001.0
3  70004.0     110.50  2012-08-17         3003          NaN
4      NaN     948.50  2012-09-10         3002       5002.0
5  70005.0    2400.60  2012-07-27         3001       5001.0
6      NaN    5760.00  2012-09-10         3001       5001.0
7  70010.0    1983.43  2012-10-10         3004          NaN
8  70003.0    2480.40  2012-10-10         3003       5003.0
9  70012.0     250.45  2012-06-27         3002       5002.0
10     NaN      75.29  2012-08-17         3001       5003.0
11 70013.0    3045.60  2012-04-25         3001          NaN
    ord_no  purch_amt    ord_date  customer_id  salesman_id
0  70001.0     150.50  2012-10-05         3002       5002.0
2  70002.0      65.26         NaN         3001       5001.0
3  70004.0     110.50  2012-08-17         3003          NaN
5  70005.0    2400.60  2012-07-27         3001       5001.0
7  70010.0    1983.43  2012-10-10         3004          NaN
8  70003.0    2480.40  2012-10-10         3003       5003.0
9  70012.0     250.45  2012-06-27         3002       5002.0
11 70013.0    3045.60  2012-04-25         3001          NaN
```