01. Exploratory Data Analysis and Preliminary Cleaning

[1]
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib
from matplotlib import pyplot as plt
from pandas_profiling import ProfileReport
from pathlib import Path
```

[2]
```
df = pd.read_csv("data/Detail_Incident.csv", sep=";",
decimal=',')
```

[3]
```
df.columns
```

```
Index(['CI Name (aff)', 'CI Type (aff)', 'CI Subtype (aff)',
       'Service Component WBS (aff)', 'Incident ID', 'Status', 'Impact',
       'Urgency', 'Priority', 'Category', 'KM number', 'Alert Status',
       '# Reassignments', 'Open Time', 'Reopen Time', 'Resolved Time',
       'Close Time', 'Handle Time (Hours)', 'Closure Code',
       '# Related Interactions', 'Related Interaction', '# Related
Incidents',
       '# Related Changes', 'Related Change', 'CI Name (CBy)', 'CI Type
(CBy)',
       'CI Subtype (CBy)', 'ServiceComp WBS (CBy)', 'Unnamed: 28',
       'Unnamed: 29', 'Unnamed: 30', 'Unnamed: 31', 'Unnamed: 32',
       'Unnamed: 33', 'Unnamed: 34', 'Unnamed: 35', 'Unnamed: 36',
       'Unnamed: 37', 'Unnamed: 38', 'Unnamed: 39', 'Unnamed: 40',
       'Unnamed: 41', 'Unnamed: 42', 'Unnamed: 43', 'Unnamed: 44',
       'Unnamed: 45', 'Unnamed: 46', 'Unnamed: 47', 'Unnamed: 48',
       'Unnamed: 49', 'Unnamed: 50', 'Unnamed: 51', 'Unnamed: 52',
       'Unnamed: 53', 'Unnamed: 54', 'Unnamed: 55', 'Unnamed: 56',
       'Unnamed: 57', 'Unnamed: 58', 'Unnamed: 59', 'Unnamed: 60',
       'Unnamed: 61', 'Unnamed: 62', 'Unnamed: 63', 'Unnamed: 64',
       'Unnamed: 65', 'Unnamed: 66', 'Unnamed: 67', 'Unnamed: 68',
       'Unnamed: 69', 'Unnamed: 70', 'Unnamed: 71', 'Unnamed: 72',
       'Unnamed: 73', 'Unnamed: 74', 'Unnamed: 75', 'Unnamed: 76',
       'Unnamed: 77'],
      dtype='object')
```

Remove empty rows and columns

[4]
```
df.dropna(axis='columns', how='all', inplace=True)
```

[5]
```
df.dropna(axis='rows', how='all', inplace=True)
```

[6]
```
df.shape
```

```
(46606, 28)
```

Adjust column names for easier reference

[7]
```
df.columns = df.columns.str.replace(' ', '_')
df.columns = df.columns.str.replace('(', '')
df.columns = df.columns.str.replace(')', '')
df.columns = df.columns.str.replace('#', 'Count')
```

[8]
```
df.columns
```

```
Index(['CI_Name_aff', 'CI_Type_aff', 'CI_Subtype_aff',
       'Service_Component_WBS_aff', 'Incident_ID', 'Status', 'Impact',
       'Urgency', 'Priority', 'Category', 'KM_number', 'Alert_Status',
       'Count_Reassignments', 'Open_Time', 'Reopen_Time',
'Resolved_Time',
       'Close_Time', 'Handle_Time_Hours', 'Closure_Code',
       'Count_Related_Interactions', 'Related_Interaction',
       'Count_Related_Incidents', 'Count_Related_Changes',
'Related_Change',
       'CI_Name_CBy', 'CI_Type_CBy', 'CI_Subtype_CBy',
'ServiceComp_WBS_CBy'],
      dtype='object')
```

Convert date columns to datetime

[9]
```
colsDatetime = ['Open_Time', 'Reopen_Time', 'Resolved_Time',
'Close_Time']
```

[10]
```
for i in colsDatetime:
    df[i] = pd.to_datetime(df[i], format='%d/%m/%Y %H:%M:%S',
errors='coerce' )
```

[11]
```
df.dtypes
```

```
CI_Name_aff                          object
CI_Type_aff                          object
CI_Subtype_aff                       object
Service_Component_WBS_aff            object
Incident_ID                          object
Status                               object
Impact                              float64
Urgency                              object
Priority                            float64
Category                             object
KM_number                            object
Alert_Status                         object
Count_Reassignments                 float64
Open_Time                    datetime64[ns]
Reopen_Time                  datetime64[ns]
Resolved_Time                datetime64[ns]
Close_Time                   datetime64[ns]
Handle_Time_Hours                   float64
Closure_Code                         object
Count_Related_Interactions          float64
Related_Interaction                  object
Count_Related_Incidents             float64
Count_Related_Changes               float64
Related_Change                       object
CI_Name_CBy                          object
CI_Type_CBy                          object
CI_Subtype_CBy                       object
ServiceComp_WBS_CBy                  object
dtype: object
```

Investigate `Urgency` as an object

[12]
```
df.Urgency.value_counts()
```

```
4               18349
5               14094
3                5362
4                4239
5                2685
3                1174
2                 607
2                  89
1                   4
1                   2
5 - Very Low        1
Name: Urgency, dtype: int64
```

Fix `Urgency`, convert it along with `Impact` and `Priority` to string

[13]
```
df.Impact = df.Impact.astype(str).str[:1]
df.Priority = df.Priority.astype(str).str[:1]
df.Urgency = df.Urgency.astype(str).str[:1]
```

[14]
```
df.Urgency.value_counts()
```

```
4    22588
5    16780
3     6536
2      696
1        6
Name: Urgency, dtype: int64
```

[15]
```
df.dtypes
```

```
CI_Name_aff                        object
CI_Type_aff                        object
CI_Subtype_aff                     object
Service_Component_WBS_aff          object
Incident_ID                        object
Status                             object
Impact                             object
Urgency                            object
Priority                           object
Category                           object
KM_number                          object
Alert_Status                       object
Count_Reassignments               float64
Open_Time                  datetime64[ns]
Reopen_Time                datetime64[ns]
Resolved_Time              datetime64[ns]
Close_Time                 datetime64[ns]
Handle_Time_Hours                 float64
Closure_Code                       object
Count_Related_Interactions        float64
Related_Interaction                object
Count_Related_Incidents           float64
Count_Related_Changes             float64
Related_Change                     object
CI_Name_CBy                        object
CI_Type_CBy                        object
CI_Subtype_CBy                     object
ServiceComp_WBS_CBy                object
dtype: object
```

Output file and create profile report

[16]
```
with open("data/01.a.Detail_Incident.csv",'w') as f:
    df.to_csv(f, index=False)
```

[17]
```
profile = ProfileReport(df, title="Profile of BPIC 2014
Detail_Incident Data after Initial Cleaning", html={'style':
{'full_width': True}})
```

[18]
```
profile.to_file(Path(str("reports/01.b.Detail_Incident_Profile.ht
ml")))
```

[ ]