

04. Final Data Preparation



```
[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/03.a.Detail_Incident.csv", parse_dates=
['Open_Time', 'Resolved_Time', 'Close_Time'])
```

```
[3] df.dtypes
```

```
CI_Name_aff          object
CI_Type_aff          object
CI_Subtype_aff       object
Service_Component_WBS_aff  object
Incident_ID          object
Impact              int64
Urgency             int64
Priority            int64
KM_number           object
Count_Reassignments  float64
Open_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
ReopenedFlag         int64
TimeToResolve_Minutes  float64
SLAFail              int64
dtype: object
```

```
[4] df.head()
```

	CI_Name_aff	CI_Type_aff	CI_Subtype_aff	Service_Component_WBS
0	APP000005	application	Citrix	WBS000292
1	DSK000457	computer	Desktop	WBS000187
2	SBA000263	application	Server Based Application	WBS000072
3	SBA000154	application	Server Based Application	WBS000027
4	LAP000019	computer	Laptop	WBS000091

5 rows × 5 columns

```
[5] df.Priority.value_counts()
```

```
4    21120
5     7962
3     5721
2      402
1         3
Name: Priority, dtype: int64
```

```
[6] df["Priority"] = pd.cut(x=df['Priority'], bins=[-1,1,2,3,4,5],
labels=["1 Very High", "2 High", "3 Medium", "4 Low", "5 Very
Low"])
df["Impact"] = pd.cut(x=df['Impact'], bins=[-1,1,2,3,4,5],
labels=["1 Very High", "2 High", "3 Medium", "4 Low", "5 Very
Low"])
df["Urgency"] = pd.cut(x=df['Urgency'], bins=[-1,1,2,3,4,5],
labels=["1 Very High", "2 High", "3 Medium", "4 Low", "5 Very
Low"])
```

```
[7] df.Priority.value_counts()
```

```
4 Low          21120
5 Very Low     7962
3 Medium       5721
2 High         402
1 Very High      3
Name: Priority, dtype: int64
```

```
[ 8]  df['Open_Time_HourOfDay'] = df.Open_Time.dt.hour
      df['Resolved_Time_HourOfDay'] = df.Resolved_Time.dt.hour
      df['Close_Time_HourOfDay'] = df.Close_Time.dt.hour
```

```
[ 9]  df.head()
```

	CI_Name_aff	CI_Type_aff	CI_Subtype_aff	Service_Component_WBS
0	APP000005	application	Citrix	WBS000292
1	DSK000457	computer	Desktop	WBS000187
2	SBA000263	application	Server Based Application	WBS000072
3	SBA000154	application	Server Based Application	WBS000027
4	LAP000019	computer	Laptop	WBS000091

5 rows × 5 columns

```
[10]  df['Open_Time_DayOfWeek'] = df.Open_Time.dt.day_name()
      df['Resolved_Time_DayOfWeek'] = df.Resolved_Time.dt.day_name()
      df['Close_Time_DayOfWeek'] = df.Close_Time.dt.day_name()
```

```
[11]  df.head()
```

	CI_Name_aff	CI_Type_aff	CI_Subtype_aff	Service_Component_WBS
0	APP000005	application	Citrix	WBS000292
1	DSK000457	computer	Desktop	WBS000187
2	SBA000263	application	Server Based Application	WBS000072
3	SBA000154	application	Server Based Application	WBS000027
4	LAP000019	computer	Laptop	WBS000091

5 rows × 33 columns

```
[12] df['OpenShift'] = pd.cut(x=df['Open_Time_HourOfDay'], bins=[-1,
8, 16, 25], labels=['Night','Day','Evening'])
df['ResolvedShift'] = pd.cut(x=df['Resolved_Time_HourOfDay'],
bins=[-1, 8, 16, 25], labels=['Night','Day','Evening'])
df['CloseShift'] = pd.cut(x=df['Close_Time_HourOfDay'], bins=[-1,
8, 16, 25], labels=['Night','Day','Evening'])
```

```
[13] df.head()
```

	CI_Name_aff	CI_Type_aff	CI_Subtype_aff	Service_Component_WBS
0	APP000005	application	Citrix	WBS000292
1	DSK000457	computer	Desktop	WBS000187
2	SBA000263	application	Server Based Application	WBS000072
3	SBA000154	application	Server Based Application	WBS000027
4	LAP000019	computer	Laptop	WBS000091

5 rows × 36 columns

```
[14] df.columns
```

```
Index(['CI_Name_aff', 'CI_Type_aff', 'CI_Subtype_aff',
      'Service_Component_WBS_aff', 'Incident_ID', 'Impact', 'Urgency',
      'Priority', 'KM_number', 'Count_Reassignments', 'Open_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',
      'ReopenedFlag', 'TimeToResolve_Minutes', 'SLAFail',
      'Open_Time_HourOfDay', 'Resolved_Time_HourOfDay',
      'Close_Time_HourOfDay', 'Open_Time_DayOfWeek',
      'Resolved_Time_DayOfWeek', 'Close_Time_DayOfWeek', 'OpenShift',
      'ResolvedShift', 'CloseShift'],
      dtype='object')
```

```
[15] df['CI_TypeSubType_aff'] = df.CI_Type_aff + "-" +
df.CI_Subtype_aff
df['CI_TypeSubType_CBy'] = df.CI_Type_CBy + "-" +
df.CI_Subtype_CBy
```

```
[16] df.head()
```

	CI_Name_aff	CI_Type_aff	CI_Subtype_aff	Service_Component_WBS
0	APP000005	application	Citrix	WBS000292
1	DSK000457	computer	Desktop	WBS000187
2	SBA000263	application	Server Based Application	WBS000072
3	SBA000154	application	Server Based Application	WBS000027
4	LAP000019	computer	Laptop	WBS000091

5 rows × 38 columns

```
[17] df = df.drop([ "CI_Type_aff", "CI_Subtype_aff", "CI_Type_CBy",
"CI_Subtype_CBy"], axis='columns')
```

```
[18] df = df.drop(['Incident_ID', "Related_Interaction",
"Related_Change"], axis='columns')
```

```
[19] df.columns
```

```
Index(['CI_Name_aff', 'Service_Component_WBS_aff', 'Impact', 'Urgency',
'Priority', 'KM_number', 'Count_Reassignments', 'Open_Time',
'Resolved_Time', 'Close_Time', 'Handle_Time_Hours',
'Closure_Code',
'Count_Related_Interactions', 'Count_Related_Incidents',
'Count_Related_Changes', 'CI_Name_CBy', 'ServiceComp_WBS_CBy',
'ReopenedFlag', 'TimeToResolve_Minutes', 'SLAFail',
'Open_Time_HourOfDay', 'Resolved_Time_HourOfDay',
'Close_Time_HourOfDay', 'Open_Time_DayOfWeek',
'Resolved_Time_DayOfWeek', 'Close_Time_DayOfWeek', 'OpenShift',
```

```
    'ResolvedShift', 'CloseShift', 'CI_TypeSubType_aff',  
    'CI_TypeSubType_CBy'],  
    dtype='object')
```

```
[20] dfAtOpen = df[['CI_Name_aff', 'Service_Component_WBS_aff',  
    'Impact', 'Urgency',  
    'KM_number', 'Count_Related_Interactions',  
    'Count_Related_Incidents',  
    'Count_Related_Changes', 'SLAFail',  
    'Open_Time_HourOfDay', 'Open_Time_DayOfWeek',  
    'CI_TypeSubType_aff']]
```

```
[21] dfAtOpen.columns
```

```
Index(['CI_Name_aff', 'Service_Component_WBS_aff', 'Impact', 'Urgency',  
    'KM_number', 'Count_Related_Interactions',  
    'Count_Related_Incidents',  
    'Count_Related_Changes', 'SLAFail', 'Open_Time_HourOfDay',  
    'Open_Time_DayOfWeek', 'CI_TypeSubType_aff'],  
    dtype='object')
```

```
[22] dfAtOpen.shape
```

```
(35208, 12)
```

END and OUTPUT

```
[23] with open("data/04.a.Detail_Incident.csv", 'w') as f:  
    df.to_csv(f, index=False)
```

```
[24] df.reset_index(drop=True, inplace=True)  
    profile = ProfileReport(df, title="Profile of Final BPIC 2014  
    Detail Incident Data", html={'style': {'full_width': True}})
```

```
[25] profile.to_file(Path(str("reports/04.b.Detail_Incident_Profile.ht  
    ml")))
```

```
[26] with open("data/04.a.Detail_Incident_AtOpen.csv", 'w') as f:
```

```
dfAtOpen.to_csv(f, index=False)
```

```
[27] dfAtOpen.reset_index(drop=True, inplace=True)
      profile = ProfileReport(dfAtOpen, title="Profile of Final BPIC
      2014 Detail Incident At Open Data", html={'style': {'full_width':
      True}})
```

```
[28] profile.to_file(Path(str("reports/04.b.Detail_Incident_AtOpen_Pro
      file.html")))
```

```
[ ]
```