

Hands-on Activity 10.1 Data Analysis using Python

Submitted by: Eugene D.G. Dela Cruz
Submitted to: Engr. Roman Richard
Section: CPE22S3

Double-click (or enter) to edit

```
#importing different modules
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df = pd.read_csv('/content/data_mmda_traffic_spatial.csv')
df
```

	Date	Time	City	Location	Latitude	Longitude	High_Accuracy	Direction	Type	Lanes_Blocked
0	2018-08-20	7:55 AM	Pasig City	ORTIGAS EMERALD	14.586343	121.061481	1	EB	VEHICULAR ACCIDENT	1.0
1	2018-08-20	8:42 AM	Mandaluyong	EDSA GUADIX	14.589432	121.057243	1	NB	STALLED L300 DUE TO MECHANICAL PROBLEM	1.0
2	2018-08-20	9:13 AM	Makati City	EDSA ROCKWELL	14.559818	121.040737	1	SB	VEHICULAR ACCIDENT	1.0
3	2018-08-20	8:42 AM	Mandaluyong	EDSA GUADIX	14.589432	121.057243	1	NB	STALLED L300 DUE TO MECHANICAL PROBLEM	1.0
4	2018-08-20	10:27 AM	San Juan	ORTIGAS CLUB FILIPINO	14.601846	121.046754	1	EB	VEHICULAR ACCIDENT	1.0
...
17307	2020-12-27	12:59 PM	Manila	QUIRINO GUAZON PETRON	14.585503	120.993783	1	SB	VEHICULAR ACCIDENT	1.0

Next steps: [View recommended plots](#)

understand the data frame by providing the names of all the columns, describing the data frame, and studying the data inside the dataframe

```
df.columns
```



```
Index(['Date', 'Time', 'City', 'Location', 'Latitude', 'Longitude', 'High_Accuracy', 'Direction', 'Type', 'Lanes_Blocked', 'Involved', 'Tweet', 'Source'], dtype='object')
```

```
df.head()
```

	Date	Time	City	Location	Latitude	Longitude	High_Accuracy	Direction	Type	Lanes_Blocked	Involved
0	2018-08-20	7:55 AM	Pasig City	ORTIGAS EMERALD	14.586343	121.061481	1	EB	VEHICULAR ACCIDENT	1.0	TAX
1	2018-08-20	8:42 AM	Mandaluyong	EDSA GUADIX	14.589432	121.057243	1	NB	STALLED L300 DUE TO MECHANICAL PROBLEM	1.0	

Next steps: [View recommended plots](#)

```
df.describe()
```

	Latitude	Longitude	High_Accuracy	Lanes_Blocked	
count	17312.000000	17312.000000	17312.000000	16625.000000	
mean	14.559448	120.666794	0.955638	1.097624	
std	0.822927	6.812422	0.205905	0.302237	
min	0.000000	0.000000	0.000000	1.000000	
25%	14.577625	121.042734	1.000000	1.000000	
50%	14.603015	121.053801	1.000000	1.000000	
75%	14.632910	121.069619	1.000000	1.000000	
max	14.735495	121.119655	1.000000	6.000000	

data validation and data cleaning by dropping duplicate datas, and missing datas

```
df.drop_duplicates(inplace=True)
#this removes the duplicated rows

df.to_csv('nodupe_dataset.csv', index=False)
#save no duplicate dataset to a new csv

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 17312 entries, 0 to 17311
Data columns (total 13 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Date             17312 non-null object
1   Time             17190 non-null object
2   City             17125 non-null object
3   Location         17289 non-null object
4   Latitude         17312 non-null float64
5   Longitude        17312 non-null float64
6   High_Accuracy    17312 non-null int64
7   Direction        16455 non-null object
8   Type            17255 non-null object
9   Lanes_Blocked    16625 non-null float64
10  Involved         16880 non-null object
11  Tweet            17312 non-null object
12  Source           17312 non-null object
dtypes: float64(3), int64(1), object(9)
memory usage: 1.8+ MB
```

```
traffic_data = pd.read_csv('/content/nodupe_dataset.csv')
```

```
traffic_data = df.dropna()
#remove rows with missing values
```

traffic_data

	Date	Time	City	Location	Latitude	Longitude	High_Accuracy	Direction	Type	Lanes_Blocked
0	2018-08-20	7:55 AM	Pasig City	ORTIGAS EMERALD	14.586343	121.061481	1	EB	VEHICULAR ACCIDENT	1.0
1	2018-08-20	8:42 AM	Mandaluyong	EDSA GUADIX	14.589432	121.057243	1	NB	STALLED L300 DUE TO MECHANICAL PROBLEM	1.0
2	2018-08-20	9:13 AM	Makati City	EDSA ROCKWELL	14.559818	121.040737	1	SB	VEHICULAR ACCIDENT	1.0
3	2018-08-20	8:42 AM	Mandaluyong	EDSA GUADIX	14.589432	121.057243	1	NB	STALLED L300 DUE TO MECHANICAL PROBLEM	1.0
4	2018-08-20	10:27 AM	San Juan	ORTIGAS CLUB FILIPINO	14.601846	121.046754	1	EB	VEHICULAR ACCIDENT	1.0
...
17307	2020-12-27	12:59 PM	Manila	QUIRINO GUAZON PETRON	14.585503	120.993783	1	SB	VEHICULAR ACCIDENT	1.0

Next steps: [View recommended plots](#)

Descriptive statistics of the cleaned dataset

```
print(traffic_data.describe())
```

	Latitude	Longitude	High_Accuracy	Lanes_Blocked
count	15314.000000	15314.000000	15314.000000	15314.000000
mean	14.604863	121.051962	0.959841	1.098668
std	0.039302	0.022452	0.196339	0.301709
min	14.499519	120.959766	0.000000	1.000000
25%	14.577625	121.044944	1.000000	1.000000

50%	14.601442	121.053801	1.000000	1.000000
75%	14.632047	121.069619	1.000000	1.000000
max	14.735495	121.104893	1.000000	4.000000

traffic_data.info()

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 15314 entries, 0 to 17311
Data columns (total 13 columns):
#   Column          Non-Null Count  Dtype
---  ---
0    Date            15314 non-null  object
1    Time            15314 non-null  object
2    City            15314 non-null  object
3    Location        15314 non-null  object
4    Latitude        15314 non-null  float64
5    Longitude       15314 non-null  float64
6    High_Accuracy   15314 non-null  int64
7    Direction       15314 non-null  object
8    Type            15314 non-null  object
9    Lanes_Blocked   15314 non-null  float64
10   Involved        15314 non-null  object
11   Tweet          15314 non-null  object
12   Source         15314 non-null  object
dtypes: float64(3), int64(1), object(9)
memory usage: 2.1+ MB
```

traffic_data['Lanes_Blocked']

```
0      1.0
1      1.0
2      1.0
3      1.0
4      1.0
...
17307   1.0
17308   1.0
17309   1.0
17310   1.0
17311   1.0
Name: Lanes_Blocked, Length: 15314, dtype: float64
```

Correlation Analysis

```
laneblockedbyinv = df.loc[df['Lanes_Blocked'] == 4.0]
columnshow = ['Involved', 'Lanes_Blocked', 'Type']
laneblockedbyinv[columnshow]
#this correlates the vehicle involved with lanes blocked and accident type
```

	Involved	Lanes_Blocked	Type
9006	TOW TRUCK AND BUS	4.0	VEHICULAR ACCIDENT
16710	TRAILER TRUCK	4.0	VEHICULAR ACCIDENT

```
laneblockedbyinv = df.loc[df['Lanes_Blocked'] == 3.0]
columnshow = ['Involved', 'Lanes_Blocked', 'Type']
laneblockedbyinv[columnshow]
#this correlates the vehicle involved with lanes blocked and accident type
```

	Involved	Lanes_Blocked	Type
898	BUS AND TAXI	3.0	VEHICULAR ACCIDENT
1078	TAXI AND VAN	3.0	VEHICULAR ACCIDENT
3918	BUS AND PICK UP	3.0	VEHICULAR ACCIDENT
5414	BUSVAN AND 2 AUV	3.0	MULTIPLE COLLISION
8036	BUS AND L300	3.0	VEHICULAR ACCIDENT
9680	NaN	3.0	ONGOING DPWH CONCRETE RE BLOCKING
14968	2-TAXI AND 4-PUJ	3.0	MULTIPLE COLLISION
16398	BUS AND TAXI	3.0	VEHICULAR ACCIDENT
16830	2 CARS, AND SUV	3.0	MULTIPLE COLLISION
17103	2 CARS AND MOTORCYCLE	3.0	MULTIPLE COLLISION
17107	2 CARS AND VAN	3.0	MULTIPLE COLLISION

```
laneblockedbyinv = df.loc[df['Lanes_Blocked'] == 2.0]
columnshow = ['Involved', 'Lanes_Blocked', 'Type']
laneblockedbyinv[columnshow]
#this correlates the vehicle involved with lanes blocked and accident type
```


<ipython-input-74-7acde7bc7e21>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns from the input DataFrame to silence this warning.
x = traffic_data.corr()

Next steps: [View full interactive steps](#) [High Accuracy](#) [Lanes Blocked](#)

plt.figure(figsize=(15,15))
sns.heatmap(x, vmin = -1.0, vmax = 1.0, annot=True, cmap = 'RdBu_r')

<Axes: >

