We have used the dataset ‘Medical Appointment No Shows’ in Kaggle.

In this dataset, we used the Google Colab, an online platform that lets you write and run Python code in your browser.

Here we Identify and handle missing values using .isnull() in Python. We can see that there are no missing values in the dataset.

```
from google.colab import files
uploaded = files.upload()
df = pd.read_csv('medical.csv')
print(df.isnull().sum())
```

- **medical.csv**(text/csv) - 10739535 bytes, last modified: 4/21/2025 - 100% done

```
Saving medical.csv to medical.csv
PatientId         0
AppointmentID     0
Gender            0
ScheduledDay      0
AppointmentDay    0
Age               0
Neighbourhood     0
Scholarship       0
Hipertension      0
Diabetes          0
Alcoholism        0
Handcap           0
SMS_received      0
No-show           0
dtype: int64
```

Here we remove duplicate rows using drop_duplicates(). We can see that there are no duplicate rows in the dataset and all are unique.

```
df = df.drop_duplicates()
print(df.shape)
```

```
(110527, 14)
```

Here we can Standardize text values by keeping all the values from the column gender to uppercase, all the values from the column Neighbourhoood to proper case which is the first letter of the word in capital and rest in lower case and all the values in the column No-Show to upper case.

```
df['Gender'] = df['Gender'].str.upper()
df['Neighbourhood'] = df['Neighbourhood'].str.title()
df['No-show'] = df['No-show'].str.upper()
```

```
print(df[['Gender', 'Neighbourhood', 'No-show']].head())
```

```
Gender      Neighbourhood No-show
0      F    Jardim Da Penha      NO
1      M    Jardim Da Penha      NO
2      F      Mata Da Praia      NO
3      F  Pontal De Camburi      NO
4      F    Jardim Da Penha      NO
```

Here we can see that we can convert date formats to a consistent type (e.g., dd-mm-yyyy) so that data will be accurate and reliable.

```
df['ScheduledDay'] = pd.to_datetime(df['ScheduledDay'])
df['AppointmentDay'] = pd.to_datetime(df['AppointmentDay'])
df['ScheduledDay'] = df['ScheduledDay'].dt.strftime('%d-%m-%Y')
df['AppointmentDay'] = df['AppointmentDay'].dt.strftime('%d-%m-%Y')
print(df[['ScheduledDay', 'AppointmentDay','Gender',
'Neighbourhood']].head())
```

```
ScheduledDay AppointmentDay Gender      Neighbourhood
0   29-04-2016     29-04-2016      F    Jardim Da Penha
1   29-04-2016     29-04-2016      M    Jardim Da Penha
2   29-04-2016     29-04-2016      F      Mata Da Praia
3   29-04-2016     29-04-2016      F  Pontal De Camburi
4   29-04-2016     29-04-2016      F    Jardim Da Penha
```

Here we can rename column headers to be clean and uniform by making all headers lowercase and replace spaces or dashes with underscores for easy access so that it is easy to access and evaluate and can be accurate.

```
df.columns = [col.strip().lower().replace(' ', '_') for col in
df.columns]
print(df.columns)
```

```
Index(['patientid', 'appointmentid', 'gender', 'scheduledday',
       'appointmentday', 'age', 'neighbourhood', 'scholarship',
'hipertension',
       'diabetes', 'alcoholism', 'handcap', 'sms_received', 'no-show'],
      dtype='object')
```

We can see hyphen in the column header of no-show, to remove the hyphen we can use

```
df.columns = [col.replace('-', '') for col in df.columns]
print(df.columns)
```

```
Index(['patientid', 'appointmentid', 'gender', 'scheduledday',
```

```
       'appointmentday', 'age', 'neighbourhood', 'scholarship',
'hipertension',
       'diabetes', 'alcoholism', 'handcap', 'sms_received', 'noshow'],
      dtype='object')
```

Here we can see that to check and fix data types (e.g., age should be int, date as datetime), we can see that scheduledday and appointmentday is in object data type we need to convert it into datetime data type and PatientId is in object data type we need to convert it into float data type.

```
PatientId          object
AppointmentID       int64
Gender             object
ScheduledDay       object
AppointmentDay     object
Age                 int64
Neighbourhood      object
Scholarship         int64
Hipertension        int64
Diabetes            int64
Alcoholism          int64
Handcap             int64
SMS_received        int64
No-show            object
dtype: object

We have changed into respective datetime and float data type.
```

```
df['ScheduledDay'] = pd.to_datetime(df['ScheduledDay'],
errors='coerce')
df['AppointmentDay'] = pd.to_datetime(df['AppointmentDay'],
errors='coerce')
df['PatientId'] = df['PatientId'].astype(str).str.strip()
print(df.dtypes)
```

```
patientid                 float64
appointmentid               int64
gender                     object
scheduledday       datetime64[ns]
appointmentday     datetime64[ns]
age                         int64
neighbourhood              object
scholarship                 int64
hipertension                int64
diabetes                    int64
alcoholism                  int64
handcap                     int64
sms_received                int64
noshow                     object
dtype: object
```