DATA CLEANING PROCESS

|  |  |
| --- | --- |
| DATE | 08-10-2023 |
| TEAM ID | PROJ-212172\_TEAM\_I |
| PROJECT NAME | Earthquake Prediction Model using PYTHON |
| MAXIMUM MARK |  |

**DATA CLEANING STEPS:**

To clean the dataset we need to find a missing values and remove null values , find the outliers fill the correct values by using fillna() method.

|  |
| --- |
| import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  %matplotlib inline |

* We need to import the pandas library to work with the spreadsheet-like data enabling fast loading , aligning, manipulating, and merging , in addition to other key functions.
* We need to import the numpy library to working with numerical values as it makes it easy to apply mathematical functions.
* We need to import the matplotlib.pyplot to use the pyplot functions. Matplotlib.pyplot is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

|  |
| --- |
| **df=pd.read\_csv('D:\IBM PROJECT\database.csv')** |

* The .read\_csv() function takes a path to a CSV file and reads the data into a Pandas DataFrame object.

|  |
| --- |
| **df.head(5)** |

* The head() method returns a specified number of rows, string from the top. The head() method returns the first 5 rows if a number is not specified. Note: The column names will also be returned, in addition to the specified rows.

|  |
| --- |
| **df.isnull().sum()** |

* The function dataframe. isnull(). sum(). sum() returns the number of null values in the dataset.

**OUTPUT:**

|  |
| --- |
| Date 0  Time 0  Latitude 0  Longitude 0  Type 0  Depth 0  Depth Error 18951  Depth Seismic Stations 16315  Magnitude 0  Magnitude Type 3  Magnitude Error 23085  Magnitude Seismic Stations 20848  Azimuthal Gap 16113  Horizontal Distance 21808  Horizontal Error 22256  Root Mean Square 6060  ID 0  Source 0  Location Source 0  Magnitude Source 0  Status 0  dtype: int64 |

|  |
| --- |
| **df['Depth Error'].fillna(df['Depth Error'].mode()[0],inplace=True)** |

Now we remove the null values in the depth error column and fill the correct values in it by using the above syntax

* The fillna() method replaces the NULL values with a specified value.
* The specified value find by using the mode() method.
* The fillna() method returns a new DataFrame object unless the inplace parameter is set to True , in that case the fillna() method does the replacing in the original DataFrame instead.

|  |
| --- |
| **df.info()** |

* The info() method prints information about the DataFrame.
* The information contains the number of columns, column labels, column data types, memory usage, range index, and the number of cells in each column (non-null values).

**OUTPUT:**

|  |
| --- |
| <class 'pandas.core.frame.DataFrame'>  RangeIndex: 23412 entries, 0 to 23411  Data columns (total 21 columns):  # Column Non-Null Count Dtype  --- ------ -------------- -----  0 Date 23412 non-null object  1 Time 23412 non-null object  2 Latitude 23412 non-null float64  3 Longitude 23412 non-null float64  4 Type 23412 non-null object  5 Depth 23412 non-null float64  6 Depth Error 23412 non-null float64  7 Depth Seismic Stations 7097 non-null float64  8 Magnitude 23412 non-null float64  9 Magnitude Type 23409 non-null object  10 Magnitude Error 327 non-null float64  11 Magnitude Seismic Stations 2564 non-null float64  12 Azimuthal Gap 7299 non-null float64  13 Horizontal Distance 1604 non-null float64  14 Horizontal Error 1156 non-null float64  15 Root Mean Square 17352 non-null float64  16 ID 23412 non-null object  17 Source 23412 non-null object  18 Location Source 23412 non-null object  19 Magnitude Source 23412 non-null object  20 Status 23412 non-null object  dtypes: float64(12), object(9)  memory usage: 3.8+ MB |

|  |
| --- |
| **df['Depth Seismic Stations'].fillna(df['Depth Seismic Stations'].mode()[0],inplace=True)** |

Now we remove the null values in the depth seismic stations column and fill the correct values in it by using the above syntax

* The fillna() method replaces the NULL values with a specified value.
* The specified value find by using the mode() method.
* The fillna() method returns a new DataFrame object unless the inplace parameter is set to True , in that case the fillna() method does the replacing in the original DataFrame instead.

|  |
| --- |
| **df['Magnitude Type'].fillna(df['Magnitude Type'].mode()[0],inplace=True)** |

Now we remove the null values in the magnitude column and fill the correct values in it by using the above syntax

* The fillna() method replaces the NULL values with a specified value.
* The specified value find by using the mode() method.
* The fillna() method returns a new DataFrame object unless the inplace parameter is set to True , in that case the fillna() method does the replacing in the original DataFrame instead.

|  |
| --- |
| **df['Magnitude Error'].fillna(df['Magnitude Error'].mode()[0],inplace=True)** |

Now we remove the null values in the magnitude error column and fill the correct values in it by using the above syntax

* The fillna() method replaces the NULL values with a specified value.
* The specified value find by using the mode() method.
* The fillna() method returns a new DataFrame object unless the inplace parameter is set to True , in that case the fillna() method does the replacing in the original DataFrame instead.

|  |
| --- |
| **df['Magnitude Seismic Stations'].fillna(df['Magnitude Seismic Stations'].mode()[0],inplace=True)** |

Now we remove the null values in the magnitude seismic column and fill the correct values in it by using the above syntax

* The fillna() method replaces the NULL values with a specified value.
* The specified value find by using the mode() method.
* The fillna() method returns a new DataFrame object unless the inplace parameter is set to True , in that case the fillna() method does the replacing in the original DataFrame instead.

|  |
| --- |
| **df['Azimuthal Gap'].fillna(df['Azimuthal Gap'].mode()[0],inplace=True)** |

Now we remove the null values in the Azimuthal gap column and fill the correct values in it by using the above syntax

* The fillna() method replaces the NULL values with a specified value.
* The specified value find by using the mode() method.
* The fillna() method returns a new DataFrame object unless the inplace parameter is set to True , in that case the fillna() method does the replacing in the original DataFrame instead.

|  |
| --- |
| **df['Horizontal Distance'].fillna(df['Horizontal Distance'].mode()[0],inplace=True)** |

Now we remove the null values in the Horizontal distance column and fill the correct values in it by using the above syntax

* The fillna() method replaces the NULL values with a specified value.
* The specified value find by using the mode() method.
* The fillna() method returns a new DataFrame object unless the inplace parameter is set to True , in that case the fillna() method does the replacing in the original DataFrame instead.

|  |
| --- |
| **df['Horizontal Error'].fillna(df['Horizontal Error'].mode()[0],inplace=True**) |

Now we remove the null values in the Horizontal error column and fill the correct values in it by using the above syntax

* The fillna() method replaces the NULL values with a specified value.
* The specified value find by using the mode() method.
* The fillna() method returns a new DataFrame object unless the inplace parameter is set to True , in that case the fillna() method does the replacing in the original DataFrame instead.

|  |
| --- |
| **df['Root Mean Square'].fillna(df['Root Mean Square'].mode()[0],inplace=True)** |

Now we remove the null values in the Root mean square column and fill the correct values in it by using the above syntax

* The fillna() method replaces the NULL values with a specified value
* The specified value find by using the mode() method.
* The fillna() method returns a new DataFrame object unless the inplace parameter is set to True , in that case the fillna() method does the replacing in the original DataFrame instead.

|  |
| --- |
| **df.isnull().sum()** |

* The function dataframe. isnull(). sum(). sum() returns the number of missing values in the dataset.
* Now all the null values are removed and replaced in data set are shown in the output

**OUTPUT:**

|  |
| --- |
| Date 0  Time 0  Latitude 0  Longitude 0  Type 0  Depth 0  Depth Error 0  Depth Seismic Stations 0  Magnitude 0  Magnitude Type 0  Magnitude Error 0  Magnitude Seismic Stations 0  Azimuthal Gap 0  Horizontal Distance 0  Horizontal Error 0  Root Mean Square 0  ID 0  Source 0  Location Source 0  Magnitude Source 0  Status 0  dtype: int64 |

|  |
| --- |
| **df.Date.unique()** |

Now to check the features

* Unique() function widely used in data analysis and machine learning. The unique function in pandas is used to find the unique values from a series.
* The values in the data series are unique and correct.
* A series is a single column of a data frame.

**OUTPUT:**

|  |
| --- |
| array(['01/02/1965', '01/04/1965', '01/05/1965', ..., '12/28/2016', '12/29/2016', '12/30/2016'], dtype=object) |

|  |
| --- |
| **df.Time.unique()** |

Now to check the feature

* Unique() function widely used in data analysis and machine learning.
* The values in the time series are unique and correct.
* The unique function in pandas is used to find the unique values from a series. A series is a single column of a data frame.

**OUTPUT:**

|  |
| --- |
| array(['13:44:18', '11:29:49', '18:05:58', ..., '12:38:51', '22:30:19', '20:08:28'], dtype=object) |

|  |
| --- |
| **df.Latitude.unique()** |

Now to check the feature

* Unique() function widely used in data analysis and machine learning.
* The values in the latitude series are unique and correct.
* The unique function in pandas is used to find the unique values from a series. A series is a single column of a data frame.

**OUTPUT:**

|  |
| --- |
| array([ 19.246 , 1.863 , -20.579 , ..., 36.9179, -9.0283, 37.3973]) |

|  |
| --- |
| **df.Longitude.unique()** |

Now to check the feature

Unique() function widely used in data analysis and machine learning.

The values in the longitude series are unique and correct.

The values in the longitude series are unique and correct.

* The unique function in pandas is used to find the unique values from a series. A series is a single column of a data frame.

**OUTPUT:**

|  |
| --- |
| array([ 145.616 , 127.352 , -173.972 , ..., 140.4262, 118.6639, 141.4103]) |

|  |
| --- |
| **df.Type.unique()** |

* Unique() function widely used in data analysis and machine learning. The unique function in pandas is used to find the unique values from a series. A series is a single column of a data frame.

**OUTPUT:**

|  |
| --- |
| array(['Earthquake', 'Nuclear Explosion', 'Explosion', 'Rock Burst'], dtype=object) |

|  |
| --- |
| **df.Depth.unique()** |

* Unique() function widely used in data analysis and machine learning. The unique function in pandas is used to find the unique values from a series. A series is a single column of a data frame.

**OUTPUT:**

|  |
| --- |
| array([131.6 , 80. , 20. , ..., 12.05, 14.93, 11.94]) |

|  |
| --- |
| **df['Depth Error'].unique()** |

* Unique() function widely used in data analysis and machine learning. The unique function in pandas is used to find the unique values from a series. A series is a single column of a data frame.

**OUTPUT:**

|  |
| --- |
| array([1.8000e+00, 3.1610e+01, 6.8370e+00, 1.4500e+00, 5.3000e+00,  3.6000e-01, 1.7800e+00, 5.6000e-01, 1.4400e+00, 2.5200e+00,  1.3700e+00, 3.7900e+00, 1.0800e+00, 4.5000e-01, 3.8000e+00,  3.6000e+00, 3.4000e+00, 8.9000e+00, 5.4000e+00, 3.1000e+00,  2.0000e+00, 3.7000e+00, 1.9000e+00, 8.6000e+00, 2.1000e+00,  2.6000e+00, 1.3000e+00, 2.2000e+00, 7.4000e+00, 1.1000e+00,  2.4000e+00, 1.6000e+00, 7.5000e+00, 7.3000e+00, 2.7000e+00,  6.7000e-01, 2.9000e+00, 5.0000e+00, 1.1900e+01, 1.5000e+00,  5.1000e+00, 7.1000e+00, 4.1000e+00, 4.0000e+00, 2.5000e+00,  2.3000e+00, 3.5000e+00, 4.3000e+00, 1.2000e+00, 5.9000e+00,  3.0000e+00, 4.9000e-01, 1.7000e+00, 5.2000e+00, 3.2000e+00,  2.8000e+00, 6.3000e+00, 4.4000e+00, 7.8000e+00, 7.0000e+00,  6.9000e+00, 3.3000e+00, 4.2000e+00, 4.9000e+00, 5.8000e+00,  4.5000e+00, 1.0800e+01, 4.7000e+00, 4.8000e+00, 1.0000e+00,  5.6000e+00, 1.0400e+01, 8.8000e+00, 9.4000e+00, 9.6000e+00,  8.0000e+00, 8.4000e+00, 6.0000e+00, 9.2000e+00, 9.0000e-01,  5.7000e+00, 1.5400e+01, 6.2000e+00, 1.4000e+00, 7.9000e+00,  8.5000e+00, 2.4000e-01, 9.3000e+00, 6.8000e+00, 1.1700e+01,  2.1800e+00, 6.7000e+00, 3.9000e+00, 4.6000e+00, 2.0610e+01,  7.7000e+00, 3.1000e-01, 6.5000e+00, 6.6000e+00, 2.3200e+01,  1.4900e+01, 1.0600e+01, 9.1000e+00, 6.1000e+00, 7.2000e+00,  8.4000e-01, 6.8000e-01, 1.2400e+00, 5.5000e+00, 8.1000e+00,  1.2100e+01, 1.0100e+01, 8.2000e+00, 1.4000e+01, 6.4000e+00,  7.6000e+00, 9.0000e+00, 5.5000e-01, 1.3300e+01, 2.3800e+01,  1.0200e+01, 1.6300e+01, 1.1100e+01, 2.1000e-01, 9.8000e+00,  ...  1.3100e+00, 7.4100e+00, 2.3100e+01, 2.0200e+01, 8.3000e-01,  2.6100e+01, 2.0600e+01, 3.3400e+01, 2.4000e+01, 2.3300e+01,  2.4600e+01, 2.0900e+01, 1.9200e+01, 2.0700e+01, 7.0000e-01,  5.0000e-01, 2.0000e-01, 1.5000e-01, 4.0000e-01, 3.0000e-01,  8.0000e-01, 3.3000e-01]) |

|  |
| --- |
| **df['Depth Seismic Stations'].unique()** |

Unique() function widely used in data analysis and machine learning. The unique function in pandas is used to find the unique values from a series. A series is a single column of a data frame.

**OUTPUT:**

|  |
| --- |
| array([ 0., 13., 16., 15., 17., 20., 19., 27., 56., 52., 54.,  95., 78., 7., 18., 11., 42., 4., 12., 10., 5., 81.,  64., 41., 86., 101., 31., 47., 35., 45., 91., 87., 28.,  84., 124., 119., 32., 90., 9., 34., 25., 40., 201., 234.,  271., 180., 76., 144., 48., 73., 60., 158., 304., 415., 412.,  379., 147., 331., 113., 262., 186., 229., 418., 383., 209., 389.,  394., 402., 352., 233., 238., 166., 129., 237., 80., 137., 400.,  473., 149., 169., 116., 377., 79., 160., 202., 386., 255., 305.,  298., 382., 364., 131., 406., 405., 211., 178., 153., 351., 207.,  70., 213., 55., 282., 260., 182., 273., 355., 388., 566., 267.,  333., 100., 571., 661., 320., 61., 140., 275., 162., 151., 367.,  138., 419., 480., 467., 152., 321., 67., 299., 71., 235., 281.,  177., 428., 218., 163., 135., 280., 165., 98., 159., 593., 185.,  193., 588., 517., 481., 222., 196., 337., 198., 471., 345., 112.,  291., 287., 252., 470., 132., 106., 161., 257., 184., 175., 225.,  236., 246., 336., 168., 194., 231., 427., 312., 284., 264., 265.,  94., 130., 145., 261., 66., 190., 477., 432., 232., 263., 167.,  208., 223., 22., 97., 256., 251., 122., 205., 437., 385., 157.,  216., 307., 51., 454., 357., 390., 217., 89., 88., 117., 323.,  189., 192., 63., 65., 121., 96., 134., 58., 290., 370., 314.,  123., 398., 146., 259., 148., 120., 92., 258., 62., 187., 173.,  141., 69., 462., 191., 38., 179., 426., 125., 111., 68., 450.,  350., 142., 292., 170., 254., 21., 133., 294., 75., 99., 224.,  33., 77., 43., 203., 136., 346., 241., 82., 338., 155., 215.,  115., 156., 181., 176., 283., 446., 268., 212., 425., 128., 384.,  ...  648., 663., 600., 543., 541., 676., 519., 654., 592., 630., 619.,  777., 598., 633., 399., 668., 629., 791., 720., 779., 674., 707.,  693., 665., 832., 580., 624., 599., 584., 729., 678., 687., 760.,  814., 857., 862., 546., 682., 703., 882., 934., 738., 691., 807.,  751., 821., 793., 647., 740., 918., 745., 770., 688., 742.]) |

|  |
| --- |
| **df['Magnitude'].unique()** |

Unique() function widely used in data analysis and machine learning. The unique function in pandas is used to find the unique values from a series. A series is a single column of a data frame.

**OUTPUT:**

|  |
| --- |
| array([6. , 5.8 , 6.2 , 6.7 , 5.9 , 8.2 , 5.5 , 5.6 , 6.1 , 8.7 , 5.7 ,  7.3 , 6.5 , 6.4 , 6.3 , 7. , 7.4 , 7.6 , 6.8 , 7.7 , 7.2 , 7.8 ,  6.9 , 6.6 , 7.5 , 7.1 , 6.35, 8.1 , 5.62, 5.63, 7.9 , 5.52, 5.82,  5.54, 8. , 5.64, 5.55, 5.67, 5.84, 5.81, 6.47, 6.31, 5.75, 5.66,  5.51, 6.45, 6.57, 5.77, 5.53, 5.69, 5.89, 5.58, 8.3 , 5.94, 8.4 ,  6.48, 5.97, 9.1 , 8.6 , 5.88, 8.8 , 5.72, 6.02, 5.73]) |

|  |
| --- |
| **df['Magnitude Type'].unique()** |

* Unique() function widely used in data analysis and machine learning. The unique function in pandas is used to find the unique values from a series. A series is a single column of a data frame.

**OUTPUT:**

|  |
| --- |
| array(['MW', 'ML', 'MH', 'MS', 'MB', 'MWC', 'MD', 'MWB', 'MWW', 'MWR'],  dtype=object) |

|  |
| --- |
| **df['Magnitude Error'].unique()** |

* Unique() function widely used in data analysis and machine learning. The unique function in pandas is used to find the unique values from a series. A series is a single column of a data frame.

**OUTPUT:**

|  |
| --- |
| array([0.048, 0.123, 0.245, 0.125, 0.219, 0.187, 0.41 , 0.155, 0.055,  0.341, 0.302, 0.188, 0.246, 0.092, 0.288, 0.08 , 0.197, 0.182,  0.206, 0.161, 0.157, 0.02 , 0.05 , 0.03 , 0.15 , 0.071, 0.045,  0.106, 0.061, 0.054, 0.039, 0.043, 0.06 , 0.044, 0.063, 0.042,  0.033, 0.041, 0.059, 0.047, 0.062, 0.083, 0.078, 0.066, 0.04 ,  0.038, 0.052, 0.093, 0.049, 0.073, 0.075, 0.053, 0.046, 0.035,  0.069, 0.116, 0.098, 0.067, 0.103, 0.058, 0.175, 0.132, 0.068,  0. , 0.07 , 0.031, 0.025, 0.057, 0.028, 0.027, 0.037, 0.032,  0.065, 0.122, 0.026, 0.089, 0.127, 0.056, 0.109, 0.108, 0.076,  0.09 , 0.029, 0.081, 0.051, 0.137, 0.099, 0.079, 0.13 , 0.158,  0.036, 0.086, 0.169, 0.146, 0.021, 0.12 , 0.11 , 0.35 , 0.32 ,  0.26 ]) |

|  |
| --- |
| **df['Magnitude Seismic Stations'].unique()** |

* Unique() function widely used in data analysis and machine learning. The unique function in pandas is used to find the unique values from a series. A series is a single column of a data frame.

**OUTPUT:**

|  |
| --- |
| array([ 1., 4., 10., 0., 6., 8., 13., 9., 12., 2., 32.,  66., 75., 71., 44., 74., 55., 38., 46., 25., 7., 80.,  16., 19., 5., 40., 3., 15., 11., 62., 14., 33., 49.,  61., 18., 23., 29., 78., 53., 86., 21., 72., 59., 85.,  17., 102., 22., 28., 60., 30., 73., 54., 50., 82., 41.,  47., 20., 43., 34., 48., 63., 24., 67., 27., 99., 31.,  51., 69., 52., 90., 87., 83., 26., 45., 36., 93., 91.,  39., 79., 58., 76., 106., 65., 42., 96., 114., 35., 68.,  37., 94., 98., 77., 81., 84., 64., 56., 70., 113., 118.,  97., 88., 108., 92., 57., 89., 100., 121., 95., 116., 111.,  136., 135., 125., 126., 115., 130., 124., 139., 127., 107., 128.,  123., 131., 112., 122., 134., 104., 110., 129., 168., 156., 147.,  141., 101., 194., 138., 150., 133., 140., 132., 105., 148., 175.,  166., 109., 178., 145., 137., 270., 119., 171., 120., 142., 149.,  177., 157., 200., 146., 159., 227., 117., 182., 258., 103., 191.,  289., 251., 152., 165., 197., 211., 216., 198., 170., 226., 274.,  188., 162., 239., 254., 161., 235., 282., 295., 203., 253., 230.,  201., 219., 172., 299., 154., 214., 193., 209., 186., 206., 260.,  250., 183., 179., 208., 292., 153., 252., 224., 275., 294., 255.,  247., 207., 285., 189., 151., 296., 231., 298., 190., 218., 217.,  205., 174., 505., 388., 164., 417., 621., 229., 484., 526., 248.,  548., 184., 328., 263., 329., 483., 410., 238., 160., 338., 210.,  821., 386., 266., 428.]) |

|  |
| --- |
| **df['Azimuthal Gap'].unique()** |

* Unique() function widely used in data analysis and machine learning. The unique function in pandas is used to find the unique values from a series. A series is a single column of a data frame.

**OUTPUT:**

|  |
| --- |
| array([ 19. , 335. , 261. , ..., 35.86, 42.47, 48.58]) |

|  |
| --- |
| **df.info()** |

* The info() method prints information about the DataFrame.
* The information contains the number of columns, column labels, column data types, memory usage, range index, and the number of cells in each column (non-null values).

**OUTPUT:**

|  |
| --- |
| <class 'pandas.core.frame.DataFrame'>  RangeIndex: 23412 entries, 0 to 23411  Data columns (total 21 columns):  # Column Non-Null Count Dtype  --- ------ -------------- -----  0 Date 23412 non-null object  1 Time 23412 non-null object  2 Latitude 23412 non-null float64  3 Longitude 23412 non-null float64  4 Type 23412 non-null object  5 Depth 23412 non-null float64  6 Depth Error 23412 non-null float64  7 Depth Seismic Stations 23412 non-null float64  8 Magnitude 23412 non-null float64  9 Magnitude Type 23412 non-null object  10 Magnitude Error 23412 non-null float64  11 Magnitude Seismic Stations 23412 non-null float64  12 Azimuthal Gap 23412 non-null float64  13 Horizontal Distance 23412 non-null float64  14 Horizontal Error 23412 non-null float64  15 Root Mean Square 23412 non-null float64  16 ID 23412 non-null object  17 Source 23412 non-null object  18 Location Source 23412 non-null object  19 Magnitude Source 23412 non-null object  20 Status 23412 non-null object  dtypes: float64(12), object(9)  memory usage: 3.8+ MB |

|  |
| --- |
| **df['Horizontal Distance'].unique()** |

* Unique() function widely used in data analysis and machine learning. The unique function in pandas is used to find the unique values from a series. A series is a single column of a data frame.

**OUTPUT:**

|  |
| --- |
| array([0.778 , 1.476 , 0.6012, ..., 0.129 , 0.992 , 3.553 ]) |

|  |
| --- |
| **df['Horizontal Error'].unique()** |

* Unique() function widely used in data analysis and machine learning. The unique function in pandas is used to find the unique values from a series. A series is a single column of a data frame.

**OUTPUT:**

|  |
| --- |
| array([6.40e+00, 9.90e+01, 8.80e-01, 3.50e-01, 8.57e+00, 1.50e-01,  1.26e+00, 1.03e+00, 6.82e-01, 1.71e+00, 1.99e+00, 3.61e+00,  1.82e+00, 1.65e+00, 4.90e-01, 4.10e+00, 7.10e-01, 4.51e+00,  8.02e+00, 5.60e-01, 2.70e-01, 2.80e-01, 3.82e+00, 1.10e-01,  1.43e+01, 3.90e-01, 2.50e-01, 3.40e-01, 1.20e-01, 1.30e-01,  1.90e-01, 2.20e-01, 3.00e+00, 2.60e-01, 8.40e-01, 2.10e-01,  2.30e-01, 2.18e+00, 3.60e-01, 1.73e+00, 3.30e-01, 1.31e+00,  1.09e+00, 9.34e-01, 4.41e-01, 4.00e-01, 2.04e+00, 2.58e+00,  1.23e+00, 9.40e-01, 9.30e-01, 8.50e-02, 1.50e+00, 5.17e-01,  4.65e+00, 2.44e+00, 1.60e-01, 1.40e-01, 1.61e+00, 1.83e+00,  5.50e-01, 1.42e+00, 1.81e+00, 3.14e+00, 4.10e-01, 1.90e+00,  3.00e-01, 6.00e-01, 1.03e+01, 5.50e+00, 2.00e-01, 2.90e+00,  2.10e+00, 3.10e+00, 4.00e+00, 1.09e+01, 5.90e+00, 6.80e+00,  4.70e+00, 4.90e+00, 8.20e+00, 7.40e+00, 7.90e+00, 6.20e+00,  1.05e+01, 3.80e+00, 8.10e+00, 6.60e+00, 9.90e+00, 6.10e+00,  7.20e+00, 8.90e+00, 8.60e+00, 5.70e+00, 6.50e+00, 6.00e+00,  4.60e+00, 7.00e+00, 8.50e+00, 6.90e+00, 7.10e+00, 7.80e+00,  9.50e+00, 4.80e+00, 6.30e+00, 5.40e+00, 1.01e+01, 9.30e+00,  1.08e+01, 5.30e+00, 4.20e+00, 7.30e+00, 5.60e+00, 9.20e+00,  8.00e+00, 9.40e+00, 8.80e+00, 1.04e+01, 7.50e+00, 3.40e+00,  5.80e+00, 9.80e+00, 1.07e+01, 8.30e+00, 5.10e+00, 4.30e+00,  1.00e+01, 3.60e+00, 5.00e+00, 1.02e+01, 9.60e+00, 9.10e+00,  4.50e+00, 1.14e+01, 9.00e+00, 1.11e+01, 7.70e+00, 1.46e+01,  5.40e-01, 1.18e+01, 6.70e+00, 7.60e+00, 1.19e+01, 1.12e+01,  1.50e+01, 8.40e+00, 3.30e+00, 4.40e+00, 3.70e+00, 9.70e+00,  ...  1.33e+01, 1.31e+01, 1.39e+01, 3.20e+00, 2.30e+00, 2.20e+00,  2.80e+00, 1.25e+01, 2.40e+00, 1.34e+01, 1.70e+00, 2.70e+00,  3.50e+00, 1.20e+01, 2.00e+00, 1.28e+01, 1.30e+01, 1.80e+00,  1.21e+01, 1.06e+01, 1.32e+01, 1.60e+00, 1.29e+01, 2.60e+00,  1.17e+01, 3.32e+00, 1.20e+00, 1.30e+00, 2.50e+00, 1.13e+01]) |

|  |
| --- |
| **df['Root Mean Square'].unique()** |

* Unique() function widely used in data analysis and machine learning. The unique function in pandas is used to find the unique values from a series. A series is a single column of a data frame.

**OUTPUT:**

|  |
| --- |
| array([1. , 0.6 , 0.86 , 0.33 , 1.03 , 1.36 , 0.35 , 0.249 ,  0.324 , 0.005 , 0.55 , 3.22 , 3.44 , 0.07 , 2.75 , 0.44 ,  0.42 , 0.38 , 0.39 , 0.09 , 1.1 , 0.06 , 0.5 , 0.17 ,  0.22 , 0.15 , 1.35 , 0.57 , 0.1 , 0.04 , 0.03 , 0.79 ,  0.12 , 0.62 , 0.9 , 1.2 , 1.4 , 0.8 , 1.3 , 0.34 ,  0.251 , 1.5 , 0.7 , 0.14 , 0.23 , 1.6 , 1.7 , 1.18 ,  0.08 , 0.05 , 0.453 , 0.092 , 0.102 , 0.112 , 1.8 , 0.178 ,  0.321 , 0.388 , 0.27 , 0.13 , 0.264 , 0.256 , 0.194 , 0.224 ,  0.146 , 0.226 , 0.272 , 0. , 0.344 , 0.31 , 0.494 , 0.317 ,  0.195 , 1.11 , 1.17 , 1.01 , 1.06 , 1.31 , 0.91 , 0.82 ,  0.74 , 0.97 , 1.13 , 1.21 , 1.02 , 0.98 , 1.28 , 1.07 ,  0.95 , 1.29 , 0.93 , 1.12 , 0.88 , 0.87 , 1.38 , 0.85 ,  1.49 , 0.99 , 0.94 , 1.34 , 1.14 , 1.37 , 1.04 , 1.05 ,  1.27 , 0.92 , 0.83 , 0.77 , 0.72 , 1.09 , 1.16 , 0.96 ,  0.78 , 1.25 , 1.15 , 0.89 , 0.84 , 1.19 , 1.23 , 1.22 ,  1.24 , 1.26 , 1.42 , 1.32 , 1.43 , 1.33 , 1.08 , 1.59 ,  1.39 , 1.41 , 1.45 , 0.65 , 0.75 , 0.81 , 0.68 , 1.54 ,  1.52 , 0.76 , 1.51 , 0.66 , 0.71 , 0.21 , 0.73 , 0.446 ,  0.67 , 0.29 , 1.44 , 0.2 , 0.3 , 0.69 , 1.48 , 0.61 ,  1.47 , 1.55 , 1.53 , 1.46 , 1.63 , 1.56 , 0.1855, 1.65 ,  1.89 , 0.19 , 0.11 , 0.63 , 1.78 , 0.47 , 0.64 , 0.48 ,  0.58 , 0.52 , 0.59 , 0.54 , 0.46 , 0.51 , 0.45 , 1.68 ,  0.56 , 0.41 , 0.53 , 0.18 , 0.49 , 0.32 , 0.43 , 2.1 ,  1.95 , 2.11 , 0.28 , 0.1988, 0.1898, 0.2187] |

|  |
| --- |
| **df['ID'].unique()** |

* Unique() function widely used in data analysis and machine learning. The unique function in pandas is used to find the unique values from a series. A series is a single column of a data frame.

**OUTPUT:**

|  |
| --- |
| array(['ISCGEM860706', 'ISCGEM860737', 'ISCGEM860762', ..., 'US10007NAF', 'US10007NL0', 'US10007NTD'], dtype=object) |

|  |
| --- |
| **df['Source'].unique()** |

* Unique() function widely used in data analysis and machine learning. The unique function in pandas is used to find the unique values from a series. A series is a single column of a data frame.

**OUTPUT:**

|  |
| --- |
| array(['ISCGEM', 'ISCGEMSUP', 'OFFICIAL', 'CI', 'US', 'NC', 'GCMT','UW',  'ATLAS', 'NN', 'SE', 'AK', 'PR'], dtype=object) |

|  |
| --- |
| **df['Location Source'].unique()** |

* Unique() function widely used in data analysis and machine learning. The unique function in pandas is used to find the unique values from a series. A series is a single column of a data frame.

**OUTPUT:**

|  |
| --- |
| array([‘ISCGEM’, ‘CI’, ‘US’, ‘H’, ‘U’, ‘G’, ‘NC’, ‘B’, ‘GCMT’, ‘AG’, ‘UW’,  ‘SPE’, ‘HVO’, ‘BRK’, ‘ATLAS’, ‘AGS’, ‘PGC’, ‘BOU’, ‘SLC’, ‘OTT’,  ‘AEI’, ‘AEIC’, ‘CASC’, ‘ISK’, ‘ATH’, ‘THE’, ‘ROM’, ‘MDD’, ‘WEL’,  ‘GUC’, ‘UNM’, ‘CSEM’, ‘RSPR’, ‘JMA’, ‘NN’, ‘CAR’, ‘SJA’, ‘THE’,  ‘BEO’, ‘UCR’, ‘SE’, ‘TUL’, ‘TAP’, ‘THR’, ‘LIM’, ‘US\_WEL’, ‘AK’,  ‘PR’], dtype=object) |

|  |
| --- |
| **df[‘Magnitude Source’].unique()** |

* Unique() function widely used in data analysis and machine learning. The unique function in pandas is used to find the unique values from a series. A series is a single column of a data frame.

**OUTPUT:**

|  |
| --- |
| array(['ISCGEM', 'OFFICIAL', 'CI', 'US', '1020', 'BRK', 'NC', '1000',  'GCMT', '1009', 'UW', '1023', 'ATLAS', 'HRV', 'PAR', 'NIED', 'NN',  'SE', 'PGC', 'US\_GCMT', 'US\_PGC', 'AK', 'PR', 'GUC'], dtype=object) |

|  |
| --- |
| **df['Status'].unique()** |

* Unique() function widely used in data analysis and machine learning. The unique function in pandas is used to find the unique values from a series. A series is a single column of a data frame.

**OUTPUT:**

|  |
| --- |
| array(['Automatic', 'Reviewed'], dtype=object) |

|  |
| --- |
| **df.describe()** |

* describe function is used to get a descriptive statistics summary of a given dataframe.
* This includes mean, count, std deviation, percentiles, and min-max values of all the features. In this article, you will learn about different features of the describe function.

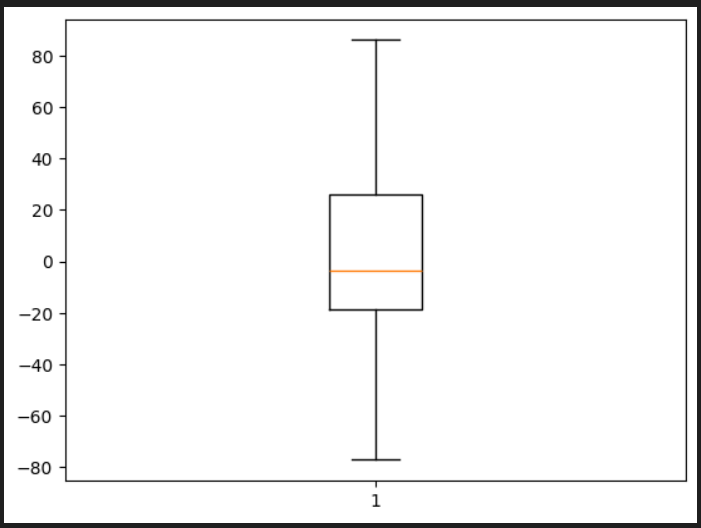
**OUTPUT:**

|  |
| --- |
| **plt.boxplot(df['Latitude'])** |

* Box Plot is also known as Whisker plot is created to display the summary of the set of data values having properties like minimum, first quartile, median, third quartile and maximum.
* In latitude series there is no outliers are present are found by using box plot

**OUTPUT:**

|  |
| --- |
| {'whiskers': [<matplotlib.lines.Line2D at 0x14cea1a3590>,  <matplotlib.lines.Line2D at 0x14cea1a2a50>],  'caps': [<matplotlib.lines.Line2D at 0x14cea1a2050>,  <matplotlib.lines.Line2D at 0x14cea1a1790>],  'boxes': [<matplotlib.lines.Line2D at 0x14ce916a910>],  'medians': [<matplotlib.lines.Line2D at 0x14cea1b0c10>],  'fliers': [<matplotlib.lines.Line2D at 0x14cea1a0890>],  'means': []} |

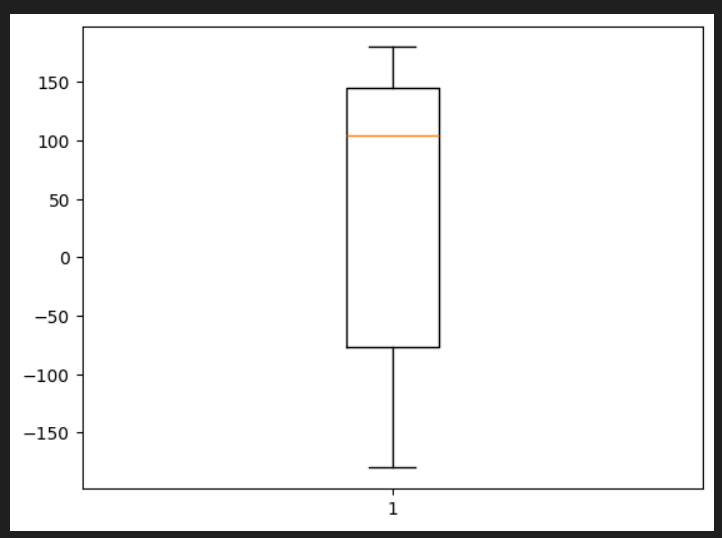


|  |
| --- |
| **plt.boxplot(df['Longitude'])** |

* Box Plot is also known as Whisker plot is created to display the summary of the set of data values having properties like minimum, first quartile, median, third quartile and maximum.
* In longitude series there is no outliers are present are found by using box plot

**OUTPUT:**

|  |
| --- |
| {'whiskers': [<matplotlib.lines.Line2D at 0x14cea18a990>,  <matplotlib.lines.Line2D at 0x14cea18b350>],  'caps': [<matplotlib.lines.Line2D at 0x14cea18bc90>,  <matplotlib.lines.Line2D at 0x14ce9a28550>],  'boxes': [<matplotlib.lines.Line2D at 0x14cea18a190>],  'medians': [<matplotlib.lines.Line2D at 0x14ce9a28d50>],  'fliers': [<matplotlib.lines.Line2D at 0x14ce9a29550>],  'means': []} |

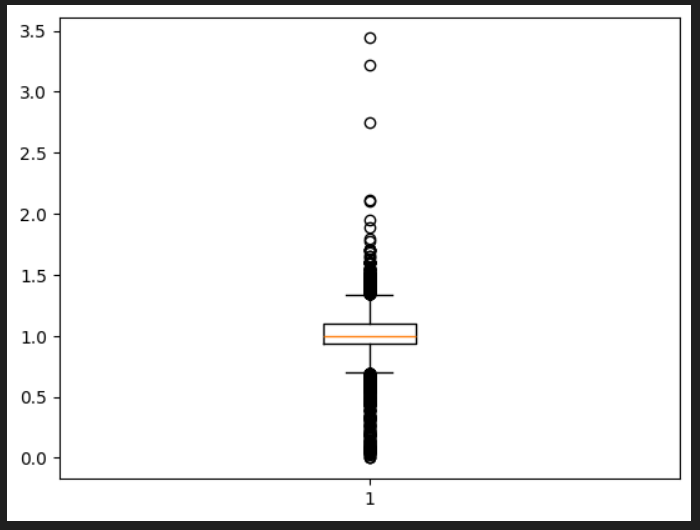


|  |
| --- |
| **plt.boxplot(df['Root Mean Square'])** |

* Box Plot is also known as Whisker plot is created to display the summary of the set of data values having properties like minimum, first quartile, median, third quartile and maximum.
* In Root mean square series there is no outliers are present are found by using box plot

**OUTPUT:**

|  |
| --- |
| {'whiskers': [<matplotlib.lines.Line2D at 0x14ce9a34590>,  <matplotlib.lines.Line2D at 0x14ce9a84a90>],  'caps': [<matplotlib.lines.Line2D at 0x14ce9a854d0>,  <matplotlib.lines.Line2D at 0x14ce9a85c90>],  'boxes': [<matplotlib.lines.Line2D at 0x14ce9a77f90>],  'medians': [<matplotlib.lines.Line2D at 0x14ce9a86590>],  'fliers': [<matplotlib.lines.Line2D at 0x14ce9a86e50>],  'means': []} |



|  |
| --- |
| **df.head()** |

* The head() method returns a specified number of rows, string from the top.
* The head() method returns the first 5 rows if a number is not specified. Note: The column names will also be returned, in addition to the specified rows.
* Now clearly seen first five rows tha datas are cleaned and nan values are removed successfully

|  |
| --- |
| **df.to\_csv("D:\IBM PROJECT\CleanedData")** |

The cleaned data set are obtained