Working with Missing Data in Pandas

February 12, 2025

1 Working with Missing Data in Pandas

https://www.geeksforgeeks.org/working-with-missing-data-in-pandas/

1.1 Checking for Missing Values in Pandas DataFrame

1.1.1 1. Checking for Missing Values Using isnull() and Non-Missing Values using notnull()

[5]:
```
import pandas as pd
import numpy as np

data = {'first score': [100,90,np.nan,95],
        'second score': [30,45,56,np.nan],
        'third score': [np.nan,40,80,98]}

df = pd.DataFrame(data)

#missing_values = df.isnull()
nonMissing_values = df.notnull()

#print(missing_values)
print(nonMissing_values)
```

```
   first score  second score  third score
0         True          True        False
1         True          True         True
2        False          True         True
3         True         False         True
```

1.1.2 2. Filtering Data based on missing values and non-missing values

[24]:
```
import pandas as pd

data = pd.read_csv("~/Desktop/myenv/Datasets/employees.csv")

#bool_series = pd.isnull(data["Gender"])
bool_series = pd.notnull(data["Gender"])
```

```
#missing_gender_data = data[bool_series]
nonMissing_gender_data = data[bool_series]

#print(missing_gender_data)
print(nonMissing_gender_data)
```

```
    First Name  Gender Start Date Last Login Time  Salary  Bonus %  \
0      Douglas    Male   8/6/1993        12:42 PM   97308    6.945
1       Thomas    Male  3/31/1996         6:53 AM   61933    4.170
2        Maria  Female  4/23/1993        11:17 AM  130590   11.858
3        Jerry    Male   3/4/2005         1:00 PM  138705    9.340
4        Larry    Male  1/24/1998         4:47 PM  101004    1.389
..         …     …        …              …     …      …
994     George    Male  6/21/2013         5:47 PM   98874    4.479
996    Phillip    Male  1/31/1984         6:30 AM   42392   19.675
997    Russell    Male  5/20/2013        12:39 PM   96914    1.421
998      Larry    Male  4/20/2013         4:45 PM   60500   11.985
999     Albert    Male  5/15/2012         6:24 PM  129949   10.169

    Senior Management                 Team
0                True            Marketing
1                True                  NaN
2               False              Finance
3                True              Finance
4                True       Client Services
..                …                    …
994              True            Marketing
996             False              Finance
997             False              Product
998             False  Business Development
999              True                Sales

[855 rows x 8 columns]
```

1.2 Filling missing values in Pandas using fillna(), replace(), and interpolate()

1.2.1 1. Filling missing values with a specific value using fillna(), ffill(), bfill()

[20]:
```
import pandas as pd
import numpy as np

dict = {'first score': [100,90,np.nan,95],
        'second score': [30,45,56,np.nan],
        'third score': [np.nan, 40,80,98]}

df = pd.DataFrame(dict)

#df.fillna(0) #specific value
```

```
#df.ffill() #forward fill
df.bfill() #backward fill
```

[20]:

```
   first score  second score  third score
0        100.0          30.0         40.0
1         90.0          45.0         40.0
2         95.0          56.0         80.0
3         95.0           NaN         98.0
```

[22]:

```
import pandas as pd
import numpy as np

data = pd.read_csv("~/Desktop/myenv/Datasets/employees.csv")

#print records from 51st row to 70th row
data[51:71]
```

[22]:

```
   First Name  Gender  Start Date Last Login Time  Salary  Bonus %  \
51        NaN     NaN  12/17/2011          8:29 AM   41126   14.009
52       Todd    Male   2/18/1990          2:41 AM   49339    1.695
53       Alan     NaN    3/3/2014          1:28 PM   40341   17.578
54       Sara  Female   8/15/2007          9:23 AM   83677    8.999
55      Karen  Female  11/30/1999          7:46 AM  102488   17.653
56       Carl    Male    5/3/2006          5:55 PM  130276   16.084
57      Henry    Male   6/26/1996          1:44 AM   64715   15.107
58    Theresa  Female   4/11/2010          7:18 AM   72670    1.481
59      Irene  Female    5/7/1997          9:32 AM   66851   11.279
60      Paula     NaN  11/23/2005          2:01 PM   48866    4.271
61     Denise  Female   11/6/2001         12:03 PM  106862    3.699
62        NaN  Female   6/12/2007          5:25 PM   58112   19.414
63    Matthew    Male    1/2/2013         10:33 PM   35203   18.040
64   Kathleen     NaN   4/11/1990          6:46 PM   77834   18.771
65      Steve    Male  11/11/2009         11:44 PM   61310   12.428
66      Nancy  Female  12/15/2012         11:57 PM  125250    2.672
67     Rachel  Female   8/16/1999          6:53 AM   51178    9.735
68       Jose    Male  10/30/2004          1:39 PM   84834   14.330
69      Irene     NaN   7/14/2015          4:31 PM  100863    4.382
70       Todd     NaN   6/10/2003          2:26 PM   84692    6.617

   Senior Management                Team
51                NaN               Sales
52               True     Human Resources
53               True             Finance
54              False         Engineering
55               True             Product
56               True             Finance
57               True     Human Resources
```

```
58              True           Engineering
59             False           Engineering
60             False          Distribution
61             False  Business Development
62               NaN             Marketing
63             False       Human Resources
64             False  Business Development
65              True          Distribution
66              True  Business Development
67              True               Finance
68              True               Finance
69              True               Finance
70             False       Client Services
```

[29]:
```
import pandas as pd
import numpy as np

#data = pd.read_csv("~/Desktop/myenv/Datasets/employees.csv")

# Reading the CSV file
try:
    data = pd.read_csv("~/Desktop/myenv/Datasets/employees.csv")
    print("Data loaded successfully")
except Exception as e:
    print(f"Error loading data: {e}")

#filling null names with fillna()
data['First Name'] = data['First Name'].fillna('No First Name')

#print(data[501:1001])  # Display records from 51st row to 70th row

print(data.head())  # Default is 5 rows, but you can specify the number of rows␣
 ↪like data.head(10)

print(data.tail())  # Default is 5 rows, but you can specify the number of rows␣
 ↪like data.tail(10)

print(data.sample(10))  # Randomly select 10 rows

print(data.info())  # Provides a concise summary of the DataFrame including␣
 ↪data types and non-null counts

print(data['First Name'])  # Display the 'First Name' column, access specific␣
 ↪column
```

```
Data loaded successfully
  First Name  Gender Start Date Last Login Time  Salary  Bonus %  \
```

```
0    Douglas    Male   8/6/1993        12:42 PM   97308    6.945
1     Thomas    Male  3/31/1996         6:53 AM   61933    4.170
2      Maria  Female  4/23/1993        11:17 AM  130590   11.858
3      Jerry    Male   3/4/2005         1:00 PM  138705    9.340
4      Larry    Male  1/24/1998         4:47 PM  101004    1.389

  Senior Management            Team
0              True       Marketing
1              True             NaN
2             False         Finance
3              True         Finance
4              True  Client Services
    First Name Gender  Start Date Last Login Time  Salary  Bonus %  \
995      Henry    NaN  11/23/2014         6:09 AM  132483   16.655
996    Phillip   Male   1/31/1984         6:30 AM   42392   19.675
997    Russell   Male   5/20/2013        12:39 PM   96914    1.421
998      Larry   Male   4/20/2013         4:45 PM   60500   11.985
999     Albert   Male   5/15/2012         6:24 PM  129949   10.169

    Senior Management                 Team
995             False         Distribution
996             False              Finance
997             False              Product
998             False  Business Development
999              True                Sales
        First Name  Gender  Start Date Last Login Time  Salary  Bonus %  \
812  No First Name    Male  12/13/1994        10:34 AM  141311    5.478
934         Samuel    Male    8/7/1997        12:40 PM   43694    3.787
41       Christine     NaN   6/28/2015         1:08 AM   66582   11.308
850        Charles    Male    9/3/1997        10:04 AM  148291    6.002
830        Michael    Male   8/31/2002         1:20 AM   81206   19.908
72           Bobby    Male    5/7/2007        10:01 AM   54043    3.833
715          Peter    Male   3/22/1982         7:28 AM   77933   13.132
647         Donald    Male    4/6/1988        10:00 AM  122920    5.320
237         Cheryl  Female   9/23/2008         2:57 AM   52080    9.375
178           Jane  Female    9/3/1997         2:01 AM  144474   17.648

    Senior Management                 Team
812               NaN              Product
934              True          Engineering
41               True  Business Development
850             False                  NaN
830              True         Distribution
72              False              Product
715              True          Engineering
647             False                  NaN
237             False                Legal
178             False              Product
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   First Name         1000 non-null   object
 1   Gender             855 non-null    object
 2   Start Date         1000 non-null   object
 3   Last Login Time    1000 non-null   object
 4   Salary             1000 non-null   int64
 5   Bonus %            1000 non-null   float64
 6   Senior Management  933 non-null    object
 7   Team               957 non-null    object
dtypes: float64(1), int64(1), object(6)
memory usage: 62.6+ KB
None
0      Douglas
1       Thomas
2        Maria
3        Jerry
4        Larry
        …
995      Henry
996    Phillip
997    Russell
998      Larry
999     Albert
Name: First Name, Length: 1000, dtype: object
```

1.2.2 2. Replacing Missing values using replace()

[28]:
```
import pandas as pd
import numpy as np

data = pd.read_csv("~/Desktop/myenv/Datasets/employees.csv")

replaced_data = data.replace(to_replace=np.nan, value=-99)

print(replaced_data[["First Name", "Gender", "Salary"]].sample(20))
```

```
     First Name  Gender  Salary
102         Jack    Male  103902
887        David    Male   92242
557         Jane  Female   42424
529  Christopher    Male   82401
716         Eric    Male   51070
310       Harold    Male   66775
717        Jason     -99   97480
```

```
606      Mildred  Female   47266
286         Todd    Male   69989
632      Rebecca  Female  134673
496       Johnny    Male   76394
425        Alice  Female   51395
843       Louise  Female  106362
822      Deborah  Female  118043
628          -99     -99  147309
504          -99  Female   38275
7            -99  Female   45906
124      Marilyn  Female   76078
772      Lillian  Female  113554
862       Ronald    Male   50426
```

1.2.3 3. Filling Missing values using interpolate()

[1]:
```
import pandas as pd

df = pd.DataFrame({"A": [12,4,5,None, 1],
                   "B": [None, 2,54,3,None],
                   "C": [20,16,None,3,8],
                   "D": [14,3,None,None,6]})

print(df)
print("\n")

interpolate_df = df.interpolate(method = 'linear', limit_direction = 'forward')

print(interpolate_df)
```

```
      A     B     C     D
0  12.0   NaN  20.0  14.0
1   4.0   2.0  16.0   3.0
2   5.0  54.0   NaN   NaN
3   NaN   3.0   3.0   NaN
4   1.0   NaN   8.0   6.0


      A     B     C     D
0  12.0   NaN  20.0  14.0
1   4.0   2.0  16.0   3.0
2   5.0  54.0   9.5   4.0
3   3.0   3.0   3.0   5.0
4   1.0   3.0   8.0   6.0
```

1.3 Dropping Missing Values in Pandas using dropna()

The dropna() function in Pandas removes rows or columns with NaN values. It can be used to drop data based on different conditions.

1.3.1 1. Dropping rows with at least one Null/NaN value

[2]:
```
import pandas as pd
import numpy as np

dict = {'1st Score': [100,90,None,95],
        '2nd Score': [30,None,45,56],
        '3rd Score': [52,40,80,98],
        '4th Score': [None,None,None,65]}

df = pd.DataFrame(dict)

df.dropna()
```

[2]:

	1st Score	2nd Score	3rd Score	4th Score
3	95.0	56.0	98	65.0

1.3.2 2. Dropping Rows with All Null Values

[5]:
```
import pandas as pd
import numpy as np

dict = {'1st Score': [100,None,None,95],
        '2nd Score': [30,None,45,56],
        '3rd Score': [52,None,80,98],
        '4th Score': [None,None,None,65]}

df = pd.DataFrame(dict)

df.dropna(how='all')
```

[5]:

	1st Score	2nd Score	3rd Score	4th Score
0	100.0	30.0	52.0	NaN
2	NaN	45.0	80.0	NaN
3	95.0	56.0	98.0	65.0

1.3.3 3. Dropping Columns with At Least One Null Value

[6]:
```
import pandas as pd
import numpy as np

dict = {'1st Score': [100,90,None,95],
        '2nd Score': [30,None,45,56],
```

```
        '3rd Score': [52,40,80,98],
        '4th Score': [None,None,None,65]}

df = pd.DataFrame(dict)

df.dropna(axis=1)
```

[6]:

	3rd Score
0	52
1	40
2	80
3	98

1.3.4 4. Dropping Rows with Missing Values in CSV Files

[9]:
```
import pandas as pd

data = pd.read_csv("~/Desktop/myenv/Datasets/employees.csv")

new_data = data.dropna(axis=0, how='any')

print("Old data frame length:", len(data))
print("New data frame length:", len(new_data))
print("Rows with at least one missing value:", (len(data)-len(new_data)))
```

```
Old data frame length: 1000
New data frame length: 764
Rows with at least one missing value: 236
```

[13]:
```
import pandas as pd

data = pd.read_csv("~/Desktop/myenv/Datasets/employees.csv")

missing_data = data[data.isnull().any(axis=1)]

new_data = data.dropna(axis=0, how='any')

print("Old data frame length:", len(data))
print("New data frame length:", len(new_data))
print("Rows missing at least one value:", len(missing_data))

print("\nRows with missing values:")
print(missing_data)
```

```
Old data frame length: 1000
New data frame length: 764
Rows missing at least one value: 236
```

```
Rows with missing values:
    First Name  Gender  Start Date Last Login Time  Salary  Bonus %  \
1       Thomas    Male   3/31/1996         6:53 AM   61933    4.170
7          NaN  Female   7/20/2015        10:43 AM   45906   11.598
10      Louise  Female   8/12/1980         9:01 AM   63241   15.132
20        Lois     NaN   4/22/1995         7:18 PM   64714    4.934
22      Joshua     NaN    3/8/2012         1:58 AM   90816   18.816
..         …     …         …              …      …      …
961    Antonio     NaN   6/18/1989         9:37 PM  103050    3.050
972     Victor     NaN   7/28/2006         2:49 PM   76381   11.159
985    Stephen     NaN   7/10/1983         8:10 PM   85668    1.909
989     Justin     NaN   2/10/1991         4:58 PM   38344    3.794
995      Henry     NaN  11/23/2014         6:09 AM  132483   16.655

    Senior Management             Team
1                True              NaN
7                 NaN          Finance
10               True              NaN
20               True            Legal
22               True  Client Services
..                …              …
961             False            Legal
972              True            Sales
985             False            Legal
989             False            Legal
995             False     Distribution

[236 rows x 8 columns]
```

[ ]: