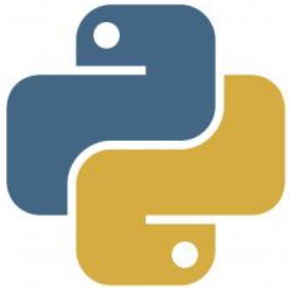
Working With



python

Data Frames 4

Missing Data

Missing Data can occur when no information is provided for one or more items or for a whole unit. Missing Data is a very big problem in a real-life scenarios.

In Pandas missing data is represented by:

•NaN : NaN (an acronym for Not a Number), is a special floating-point value recognized by all systems that use the standard IEEE floating-point representation

HiiT ...Your Premium Training Partner | World Class IT Services since 1996

Sample Data

Dirtydata.xlsx

	CustomerID	First_name	Last_Name	Phone_Number	Address	Paying Customer	Do_Not_Contact	Not_Useful_Column
0	1001	Frodo	Baggins	123-545-5421	123 Shire Lane, Shire	Yes	No	True
1	1002	Abed	Nadir	123/643/9775	93 West Main Street	No	Yes	False
2	1003	Walter	/White	7066950392	298 Drugs Driveway	N	nan	True
3	1004	Dwayne	Shurlle	123-543-2345	980 Paper Avenue, Pennsylvania, 18503	Yes	Yes	True
4	1005	John	Snow	876|678|3469	123 Dragons Road	Y	No	True
5	1006	Ron	Swanson	304-762-2467	768 City Parkway	Yes	Yes	True
6	1007	Jeff	Winger	nan	1209 South Street	No	No	False
7	1008	Sherlock	Holmes	876|678|3469	98 Clue Drive	N	No	False
8	1009	Gandalf	nan	N/a	123 Middle Earth	Yes	nan	False
9	1010	Peter	...Parker	123-545-5421	25th Main Street, New York	Yes	No	True
10	1011	Sammy	George	nan	612 Shire Lane, Shire	Yes	No	True
11	1012	Harry	Potter	7066950392	2394 Hogwarts Avenue	Y	nan	True
12	1013	Don	Draper	123-543-2345	2039 Main Street	Yes	N	False
13	1014	Leslie	Houston	876|678|3469	343 City Parkway	Yes	No	False
14	1015	Toby	Richardson_	304-762-2467	214 HR Avenue	Y	No	False
15	1016	Ron	Weasley	123-545-5421	2395 Hogwarts Avenue	No	N	False
16	1017	Michael	Scott	123/643/9775	121 Paper Avenue, Pennsylvania	Yes	No	False
17	1018	Clark	Kent	7066950392	3496 Super Lane	Y	nan	True
18	1019	Creed	Braton	N/a	N/a	N/a	Yes	True
19	1020	Anakin	Skywalker	876|678|3469	910 Tatooine Road, Tatooine	Yes	N	True
20	1020	Anakin	Skywalker	876|678|3469	910 Tatooine Road, Tatooine	Yes	N	True

Checking Null Values

df.isnull()

df.isna()

	CustomerID	First_name	Last_Name	Phone_Number	Address	Paying Customer	Do_Not_Contact	Not_Useful_Column
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	True	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
5	False	False	False	False	False	False	False	False
6	False	False	False	True	False	False	False	False
7	False	False	False	False	False	False	False	False

```
CustomerID           0
First_name           0
Last_Name            1
Phone_Number         2
Address              0
Paying Customer      0
Do_Not_Contact       4
Not_Useful_Column    0
dtype: int64
```

df.isnull().sum()

df.isna().sum()

Checking Non-Null values

df.notnull()

df.notna()

	CustomerID	First_name	Last_Name	Phone_Number	Address	Paying Customer	Do_Not_Contact	Not_Useful_Column
0	True	True	True	True	True	True	True	True
1	True	True	True	True	True	True	True	True
2	True	True	True	True	True	True	False	True
3	True	True	True	True	True	True	True	True
4	True	True	True	True	True	True	True	True
5	True	True	True	True	True	True	True	True
6	True	True	True	False	True	True	True	True

```
CustomerID          21
First_name          21
Last_Name           20
Phone_Number        19
Address             21
Paying Customer     21
Do_Not_Contact      17
Not_Useful_Column   21
dtype: int64
```

df.notnull().sum()

df.notna().sum()

Checking Duplicate Values

df.duplicated().sum()

15	1016	Ron	Weasley	123-545-5421	2395 Hogwarts Avenue	No	N	False
16	1017	Michael	Scott	123/643/9775	121 Paper Avenue, Pennsylvania	Yes	No	False
17	1018	Clark	Kent	7066950392	3496 Super Lane	Y	NaN	True
18	1019	Creed	Braton	N/a	N/a	N/a	Yes	True
19	1020	Anakin	Skywalker	876|678|3469	910 Tatooine Road, Tatooine	Yes	N	True
20	1020	Anakin	Skywalker	876|678|3469	910 Tatooine Road, Tatooine	Yes	N	True

df['First_name'].duplicated().sum()

15	1016	Ron	Weasley	123-545-5421	2395 Hogwarts Avenue	No	N	False
16	1017	Michael	Scott	123/643/9775	121 Paper Avenue, Pennsylvania	Yes	No	False
17	1018	Clark	Kent	7066950392	3496 Super Lane	Y	NaN	True
18	1019	Creed	Braton	N/a	N/a	N/a	Yes	True
19	1020	Anakin	Skywalker	876|678|3469	910 Tatooine Road, Tatooine	Yes	N	True
20	1020	Anakin	Skywalker	876|678|3469	910 Tatooine Road, Tatooine	Yes	N	True

Dropping Duplicates

df.drop_duplicates()

15	1016	Ron	Weasley	123-545-5421	2395 Hogwarts Avenue	No	N	False
16	1017	Michael	Scott	123/643/9775	121 Paper Avenue, Pennsylvania	Yes	No	False
17	1018	Clark	Kent	7066950392	3496 Super Lane	Y	NaN	True
18	1019	Creed	Braton	N/a	N/a	N/a	Yes	True
19	1020	Anakin	Skywalker	876|678|3469	910 Tatooine Road, Tatooine	Yes	N	True
20	1020	Anakin	Skywalker	876|678|3469	910 Tatooine Road, Tatooine	Yes	N	True

df.drop_duplicates(subset='Paying Customer')

	CustomerID	First_name	Last_Name	Phone_Number	Address	Paying Customer	Do_Not_Contact	Not_Useful_Column
0	1001	Frodo	Baggins	123-545-5421	123 Shire Lane, Shire	Yes	No	True
1	1002	Abed	Nadir	123/643/9775	93 West Main Street	No	Yes	False
2	1003	Walter	/White	7066950392	298 Drugs Driveway	N	NaN	True
4	1005	John	Snow	876|678|3469	123 Dragons Road	Y	No	True
18	1019	Creed	Braton	N/a	N/a	N/a	Yes	True

Dropping Columns


```
df.drop(columns='Not_Useful_Column')
```

Address	Paying Customer	Do_Not_Contact	
123 Shire Lane, Shire	Yes	No	
93 West Main Street	No	Yes	
298 Drugs Driveway	N	NaN	
980 Paper Avenue, Pennsylvania, 18503	Yes	Yes	
123 Dragons Road	Y	No	
768 City Parkway	Yes	Yes	
1209 South Street	No	No	
98 Clue Drive	N	No	
123 Middle Earth	Yes	NaN	
25th Main Street, New York	Yes	No	
612 Shire Lane, Shire	Yes	No	

Dropping Index

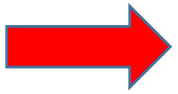
df.drop(index=[1,2,5,6])

	CustomerID	First_name	Last_Name	Phone_Number	Address
0	1001	Frodo	Baggins	123-545-5421	123 Shire Lane, Shire
3	1004	Dwayne	Shurlle	123-543-2345	980 Paper Avenue, Pennsylvania, 18503
4	1005	John	Snow	876|678|3469	123 Dragons Road
7	1008	Sherlock	Holmes	876|678|3469	98 Clue Drive
8	1009	Gandalf	NaN	N/a	123 Middle Earth
9	1010	Peter	...Parker	123-545-5421	25th Main Street, New York
10	1011	Sammy	George	NaN	612 Shire Lane, Shire

L-Strip Method


```
df['Last_Name'].str.lstrip('/')
```

	Last_Name
0	Baggins
1	Nadir
2	/White
3	Shurlle
4	Snow
5	Swanson
6	Winger
7	Holmes
8	NaN
9	...Parker
10	George
11	Potter
12	Draper
13	Houston
14	Richardson_
15	Weasley
16	Scott
17	Kent
18	Braton
19	Skywalker
20	Skywalker



	Last_Name
0	Baggins
1	Nadir
2	White
3	Shurlle
4	Snow
5	Swanson
6	Winger
7	Holmes
8	NaN
9	...Parker
10	George
11	Potter
12	Draper
13	Houston
14	Richardson_
15	Weasley
16	Scott
17	Kent
18	Braton
19	Skywalker
20	Skywalker

R-Strip Method


```
df['Last_Name'].str.rstrip('_')
```

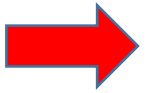
0	Baggins
1	Nadir
2	/White
3	Shurlle
4	Snow
5	Swanson
6	Winger
7	Holmes
8	NaN
9	...Parker
10	George
11	Potter
12	Draper
13	Houston
14	Richardson_
15	Weasley
16	Scott
17	Kent
18	Braton
19	Skywalker
20	Skywalker

0	Baggins
1	Nadir
2	/White
3	Shurlle
4	Snow
5	Swanson
6	Winger
7	Holmes
8	NaN
9	...Parker
10	George
11	Potter
12	Draper
13	Houston
14	Richardson
15	Weasley
16	Scott
17	Kent
18	Braton
19	Skywalker
20	Skywalker

Strip Method


```
df['Last_Name'].str.strip('._/')
```

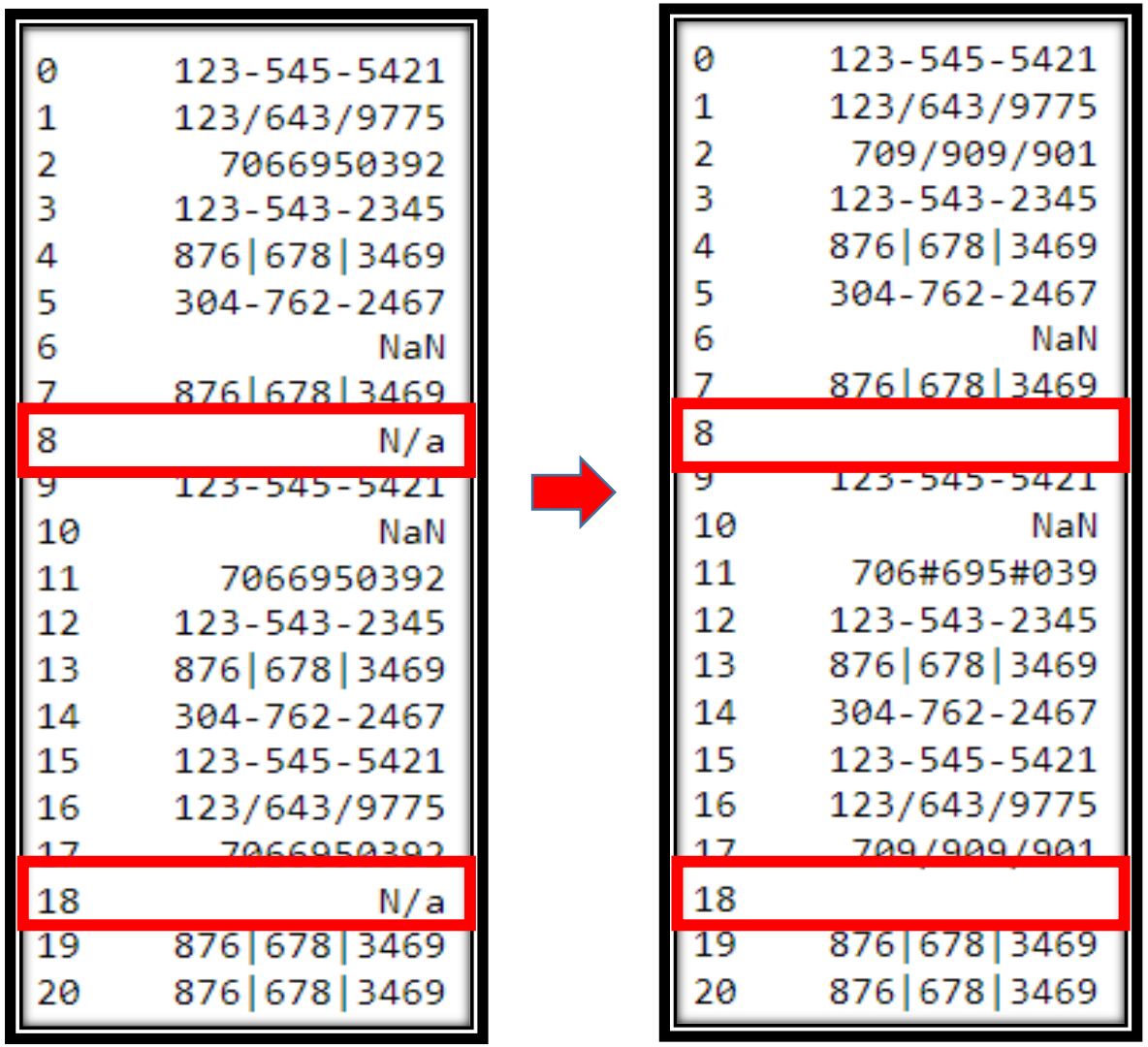
0	Baggins
1	Nadir
2	/White
3	Shurlie
4	Snow
5	Swanson
6	Winger
7	Holmes
8	NaN
9	...Parker
10	George
11	Potter
12	Draper
13	Houston
14	Richardson_
15	Weasley
16	Scott
17	Kent
18	Braton
19	Skywalker
20	Skywalker



0	Baggins
1	Nadir
2	White
3	Shurlie
4	Snow
5	Swanson
6	Winger
7	Holmes
8	NaN
9	Parker
10	George
11	Potter
12	Draper
13	Houston
14	Richardson
15	Weasley
16	Scott
17	Kent
18	Braton
19	Skywalker
20	Skywalker

Replace Method


```
df['Phone_Number'].str.replace('N/a', ' ')
```



	Phone_Number
0	123-545-5421
1	123/643/9775
2	7066950392
3	123-543-2345
4	876|678|3469
5	304-762-2467
6	NaN
7	876|678|3469
8	N/a
9	123-545-5421
10	NaN
11	7066950392
12	123-543-2345
13	876|678|3469
14	304-762-2467
15	123-545-5421
16	123/643/9775
17	7066950392
18	N/a
19	876|678|3469
20	876|678|3469

	Phone_Number
0	123-545-5421
1	123/643/9775
2	709/909/901
3	123-543-2345
4	876|678|3469
5	304-762-2467
6	NaN
7	876|678|3469
8	
9	123-545-5421
10	NaN
11	706#695#039
12	123-543-2345
13	876|678|3469
14	304-762-2467
15	123-545-5421
16	123/643/9775
17	709/909/901
18	
19	876|678|3469
20	876|678|3469

Replace Method


```
df['Phone'].str.replace('[^A-Za-z0-9]',‘’,regex=True)
```

	Phone
0	123-545-5421
1	123/643/9775
2	709/909/901
3	123-543-2345
4	876|678|3469
5	304-762-2467
6	NaN
7	876|678|3469
8	
9	123-545-5421
10	NaN
11	706#695#039
12	123-543-2345
13	876|678|3469
14	304-762-2467
15	123-545-5421
16	123/643/9775
17	709/909/901
18	
19	876|678|3469
20	876|678|3469

	Phone
0	1235455421
1	1236439775
2	709909901
3	1235432345
4	8766783469
5	3047622467
6	NaN
7	8766783469
8	
9	1235455421
10	NaN
11	706695039
12	1235432345
13	8766783469
14	3047622467
15	1235455421
16	1236439775
17	709909901
18	
19	8766783469
20	8766783469

Replace Method


```
df['Paying Customer'].str.replace('Yes','Y')
```

```
df['Paying Customer'].str.replace('No','N')
```

	Paying Customer
0	Yes
1	No
2	N
3	Yes
4	Y
5	Yes
6	No
7	N
8	Yes
9	Yes
10	Yes
11	Y
12	Yes
13	Yes
14	Y
15	No
16	Yes
17	Y
18	N/a
19	Yes
20	Yes

	Paying Customer
0	Y
1	N
2	N
3	Y
4	Y
5	Y
6	N
7	N
8	Y
9	Y
10	Y
11	Y
12	Y
13	Y
14	Y
15	N
16	Y
17	Y
18	N/a
19	Y
20	Y

Split Method

	Address
0	123 Shire Lane, Shire
1	93 West Main Street
2	298 Drugs Driveway
3	980 Paper Avenue, Pennsylvania, 18503
4	123 Dragons Road
5	768 City Parkway
6	1209 South Street
7	98 Clue Drive
8	123 Middle Earth
9	25th Main Street, New York
10	612 Shire Lane, Shire

```
df[['Street','State','Zip Code']] =

df['Address'].str.split(',',expand=True)
```

	Street	State	Zip Code
0	123 Shire Lane	Shire	None
1	93 West Main Street	None	None
2	298 Drugs Driveway	None	None
3	980 Paper Avenue	Pennsylvania	18503
4	123 Dragons Road	None	None
5	768 City Parkway	None	None
6	1209 South Street	None	None
7	98 Clue Drive	None	None
8	123 Middle Earth	None	None
9	25th Main Street	New York	None
10	612 Shire Lane	Shire	None

Other Methods