DATA WRANGLING

In [187]:

```
import pandas as pd

#online link
#url='https://raw.githubusercontent.com/chrisalbon/simulated_datasets/master/titanic.csv'

#csv file
url = 'C:/Users/Prerna/Desktop/ML_jupyter_notenooks/titanic.csv'

#Creating a dataframe
dataframe = pd.read_csv(url)

#printing the dataframe
dataframe

```

Out[187]:

	Name	PClass	Age	Sex	Survived	SexCode
0	Allen, Miss Elisabeth Walton	1st	29.00	female	1	1
1	Allison, Miss Helen Loraine	1st	2.00	female	0	1
2	Allison, Mr Hudson Joshua Creighton	1st	30.00	male	0	0
3	Allison, Mrs Hudson JC (Bessie Waldo Daniels)	1st	25.00	female	0	1
4	Allison, Master Hudson Trevor	1st	0.92	male	1	0
5	Anderson, Mr Harry	1st	47.00	male	1	0
6	Andrews, Miss Kornelia Theodosia	1st	63.00	female	1	1
7	Andrews, Mr Thomas, jr	1st	39.00	male	0	0
8	Appleton, Mrs Edward Dale (Charlotte Lamson)	1st	58.00	female	1	1
9	Artagaveytia, Mr Ramon	1st	71.00	male	0	0
10	Astor, Colonel John Jacob	1st	47.00	male	0	0

In [188]:
```
#Reading first five records with head()
dataframe.head(5)
```

Out[188]:

	Name	PClass	Age	Sex	Survived	SexCode
0	Allen, Miss Elisabeth Walton	1st	29.00	female	1	1
1	Allison, Miss Helen Loraine	1st	2.00	female	0	1
2	Allison, Mr Hudson Joshua Creighton	1st	30.00	male	0	0
3	Allison, Mrs Hudson JC (Bessie Waldo Daniels)	1st	25.00	female	0	1
4	Allison, Master Hudson Trevor	1st	0.92	male	1	0

In [189]:
```
#Reading last five records with tail()
dataframe.tail(5)
```

Out[189]:

	Name	PClass	Age	Sex	Survived	SexCode
1308	Zakarian, Mr Artun	3rd	27.0	male	0	0
1309	Zakarian, Mr Maprieder	3rd	26.0	male	0	0
1310	Zenni, Mr Philip	3rd	22.0	male	0	0
1311	Lievens, Mr Rene	3rd	24.0	male	0	0
1312	Zimmerman, Leo	3rd	29.0	male	0	0

In [190]:
```
#showing dimensions of the data
dataframe.shape
```

Out[190]: (1313, 6)

(1313,6) = (no of rows, no of columns)

In [191]:
```
#get the descriptive statistics of data
dataframe.describe()
```

Out[191]:

	Age	Survived	SexCode
count	756.000000	1313.000000	1313.000000
mean	30.397989	0.342727	0.351866
std	14.259049	0.474802	0.477734
min	0.170000	0.000000	0.000000
25%	21.000000	0.000000	0.000000
50%	28.000000	0.000000	0.000000
75%	39.000000	1.000000	1.000000
max	71.000000	1.000000	1.000000

Navigating the dataframes using loc and iloc.

All rows in a pandas DataFrame have a unique index value.

By default, this index is an integer indicating the row position in the DataFrame however, it does not have to be.

DataFrame indexes can be set to be unique alphanumeric strings or customer numbers.

To select individual rows and slices of rows, pandas provides two methods:

• loc is useful when the index of the DataFrame is a label (e.g., a string).

• iloc works by looking for the position in the DataFrame.

In [192]:
```
# selecting one of more rows with iloc.
#Selecting the first row
dataframe.iloc[0]
```

Out[192]:
```
Name        Allen, Miss Elisabeth Walton
PClass                               1st
Age                                   29
Sex                               female
Survived                               1
SexCode                                1
Name: 0, dtype: object
```

In [193]:
```
#Selecting more than one rows using slicing with iloc
#selecting first 4 rows from the dataset
dataframe.iloc[1:4]
```

Out[193]:

	Name	PClass	Age	Sex	Survived	SexCode
1	Allison, Miss Helen Loraine	1st	2.0	female	0	1
2	Allison, Mr Hudson Joshua Creighton	1st	30.0	male	0	0
3	Allison, Mrs Hudson JC (Bessie Waldo Daniels)	1st	25.0	female	0	1

In [194]:
```
#selecting first 4 rows from the dataset
dataframe.iloc[:4]

#one more variation for selecting first 4 rows
#dataframe.iloc[0:4]
```

Out[194]:

	Name	PClass	Age	Sex	Survived	SexCode
0	Allen, Miss Elisabeth Walton	1st	29.0	female	1	1
1	Allison, Miss Helen Loraine	1st	2.0	female	0	1
2	Allison, Mr Hudson Joshua Creighton	1st	30.0	male	0	0
3	Allison, Mrs Hudson JC (Bessie Waldo Daniels)	1st	25.0	female	0	1

Using loc for selecting the rows.

DataFrames do not need to be numerically indexed.

We can set the index of a Data- Frame to any value where the value is unique to each row.

For example, we can set the index to be passenger names and then select rows using a name.

Use set_index() for setting the index.

In [195]:
```
#Set the Name columne as index value for dataframe
dataframe = dataframe.set_index(dataframe['Name'])

#Select the rows with loc.
dataframe.loc['Allison, Miss Helen Loraine']
```

Out[195]:
```
Name        Allison, Miss Helen Loraine
PClass                              1st
Age                                   2
Sex                              female
Survived                              0
SexCode                               1
Name: Allison, Miss Helen Loraine, dtype: object
```

In [196]:
```
#Selecting the rows with loc using slicing
dataframe.loc['Allen, Miss Elisabeth Walton':'Allison, Mr Hudson Joshua Creighton']
```

Out[196]:

Name	Name	PClass	Age	Sex	Survived	SexCode
Allen, Miss Elisabeth Walton	Allen, Miss Elisabeth Walton	1st	29.0	female	1	1
Allison, Miss Helen Loraine	Allison, Miss Helen Loraine	1st	2.0	female	0	1
Allison, Mr Hudson Joshua Creighton	Allison, Mr Hudson Joshua Creighton	1st	30.0	male	0	0

Selecting Rows Based on Conditionals

In [197]:

```
import pandas as pd

#Creating URL
url = 'C:/Users/Prerna/Desktop/ML_jupyter_notenooks/titanic.csv'

#Creating a dataframe
dataframe = pd.read_csv(url)

# 1) Select all the women on the titanic and display top 5 records of women

dataframe[dataframe['Sex']=='female'].head(5)

```

Out[197]:

	Name	PClass	Age	Sex	Survived	SexCode
0	Allen, Miss Elisabeth Walton	1st	29.0	female	1	1
1	Allison, Miss Helen Loraine	1st	2.0	female	0	1
3	Allison, Mrs Hudson JC (Bessie Waldo Daniels)	1st	25.0	female	0	1
6	Andrews, Miss Kornelia Theodosia	1st	63.0	female	1	1
8	Appleton, Mrs Edward Dale (Charlotte Lamson)	1st	58.0	female	1	1

In [198]:

```
# Filter rows with multiple conditions
'''2) select all the 2nd class male passengers on the titanic having
age more than 50 years '''

dataframe[(dataframe['Sex']=='male') & (dataframe['Age']>50)
          & (dataframe['PClass']=='2nd')]

```

Out[198]:

	Name	PClass	Age	Sex	Survived	SexCode
328	Ashby, Mr John	2nd	57.0	male	0	0
333	Bateman, Rev Robert James	2nd	51.0	male	0	0
362	Carter, Rev Ernest Courtenay	2nd	54.0	male	0	0
364	Chapman, Mr Charles Henry	2nd	52.0	male	0	0
447	Hodges, Mr Henry Price	2nd	52.0	male	0	0
505	Mitchell, Mr Henry Michael	2nd	71.0	male	0	0
506	Moraweck, Dr Ernest	2nd	54.0	male	0	0
509	Myles, Mr Thomas Francis	2nd	64.0	male	0	0
556	Sjostedt, Mr Ernst Adolf	2nd	59.0	male	0	0

In [200]:

```
#3) select 1st class male passangers that has survived or more than 50 years in titanic

dataframe[(((dataframe['Sex']=='male') & (dataframe['PClass']=='1st'))
           & ((dataframe['Survived']==1) | (dataframe['Age']>50)))]
```

Out[200]:

	Name	PClass	Age	Sex	Survived	SexCode
4	Allison, Master Hudson Trevor	1st	0.92	male	1	0
5	Anderson, Mr Harry	1st	47.00	male	1	0
9	Artagaveytia, Mr Ramon	1st	71.00	male	0	0
13	Barkworth, Mr Algernon H	1st	NaN	male	1	0
18	Beckwith, Mr Richard Leonard	1st	37.00	male	1	0
20	Behr, Mr Karl Howell	1st	26.00	male	1	0
22	Bishop, Mr Dickinson H	1st	25.00	male	1	0
24	Bjornstrm-Steffansson, Mr Mauritz Hakan	1st	28.00	male	1	0
26	Blank, Mr Henry	1st	39.00	male	1	0
32	Bradley, Mr George	1st	NaN	male	1	0
40	Calderhead, Mr Edward P	1st	NaN	male	1	0
43	Cardeza, Mr Thomas Drake Martinez	1st	36.00	male	1	0
47	Carter, Mr William Ernest	1st	36.00	male	1	0
50	Carter, Master William T II	1st	11.00	male	1	0
57	Chambers, Mr Norman Campbell	1st	27.00	male	1	0
60	Chevre, Mr Paul	1st	NaN	male	1	0
72	Crosby, Captain Edward Gifford	1st	70.00	male	0	0
77	Daly, Mr Peter Denis	1st	NaN	male	1	0
78	Daniel, Mr Robert Williams	1st	27.00	male	1	0
82	Dick, Mr Albert Adrian	1st	31.00	male	1	0
84	Dodge, Dr Washington	1st	NaN	male	1	0
86	Dodge, Master Washington	1st	4.00	male	1	0

	Name	PClass	Age	Sex	Survived	SexCode
90	Duff Gordon, Sir Cosmo Edmund	1st	49.00	male	1	0
97	Flynn, Mr John Irving	1st	NaN	male	1	0
103	Fortune, Mr Mark	1st	64.00	male	0	0
106	Frauenthal, Dr Henry William	1st	49.00	male	1	0
108	Frauenthal, Mr Isaac Gerald	1st	44.00	male	1	0
110	Frolicher-Stehli, Mr Maxmillian	1st	60.00	male	1	0
117	Goldenberg, Mr Samuel L	1st	49.00	male	1	0
119	Goldschmidt, Mr George B	1st	71.00	male	0	0
...	...	...	...	...	...	...
221	Ryerson, Master John Borie	1st	13.00	male	1	0
223	Saalfeld, Mr Adolphe	1st	NaN	male	1	0
224	Salomon, Mr Abraham L	1st	NaN	male	1	0
226	Seward, Mr Frederic Kimber	1st	34.00	male	1	0
228	Silverthorne, Mr Spencer Victor	1st	36.00	male	1	0
231	Simonius-Blumer, Col Alfons	1st	56.00	male	1	0
232	Sloper, Mr William Thompson	1st	28.00	male	1	0
233	Smart, Mr John Montgomery	1st	56.00	male	0	0
234	Smith, Mr James Clinch	1st	56.00	male	0	0
238	Snyder, Mr John Pillsbury	1st	24.00	male	1	0
240	Spedden, Mr Frederick Oakley	1st	45.00	male	1	0
242	Spedden, Master Robert Douglas	1st	6.00	male	1	0
243	Spencer, Mr William Augustus	1st	57.00	male	0	0
245	Staehlin, Dr Max	1st	32.00	male	1	0
246	Stead, Mr William Thomas	1st	62.00	male	0	0
247	Stengel, Mr Charles Emil Henry	1st	54.00	male	1	0
252	Straus, Mr Isidor	1st	67.00	male	0	0

	Name	PClass	Age	Sex	Survived	SexCode
254	Sutton, Mr Frederick	1st	61.00	male	0	0
256	Taussig, Mr Emil	1st	52.00	male	0	0
259	Taylor, Mr Elmer Zebley	1st	48.00	male	1	0
263	Thayer, Mr John Borland, jr	1st	17.00	male	1	0
266	Tucker, Mr Gilbert Milligan, jr	1st	31.00	male	1	0
268	Van Derhoef, Mr Wyckoff	1st	61.00	male	0	0
270	Warren, Mr Frank Manley	1st	64.00	male	0	0
272	Weir, Col John	1st	60.00	male	0	0
274	White, Mr Percival Wayland	1st	54.00	male	0	0
276	Wick, Mr George Dennick	1st	57.00	male	0	0
283	Williams, Mr Charles Duane	1st	51.00	male	0	0
285	Williams, Mr Richard Norris II	1st	21.00	male	1	0
286	Woolner, Mr Hugh	1st	NaN	male	1	0

86 rows × 6 columns

Replacing Values

You can replace the values in the coloumns with replace().

Syntax: dataframe(columnname).replace(value to be replaced, Value to replace with)

In [201]:
```
import pandas as pd
#Creating URL
url = 'C:/Users/Prerna/Desktop/ML_jupyter_notenooks/titanic.csv'

#Creating a dataframe
dataframe = pd.read_csv(url)

#Replacing values in Survived column  1 with Yes.
dataframe['Survived'].replace(1,'yes').head(5)

```

Out[201]:
```
0    yes
1      0
2      0
3      0
4    yes
Name: Survived, dtype: object
```

In [202]:
```
#Replacing values of 1 with Yes and 0 with NO, and Pclass in whole dataframe.
new_df = dataframe.replace([0,1,'1st','2nd','3rd'],
                           ['Yes','No','First','Second','Third'])
```

Renaming Columns and Deleting Columns

In [203]:

```
#Drop column with drop() with axis=1 for column
new_df = new_df.drop('SexCode',axis=1)

#Rename column PClass to Passenger class using rename()
new_df = new_df.rename(columns = {'PClass': 'Passenger Class', 'Sex': 'Gender'})
new_df.head(5)
```

Out[203]:

	Name	Passenger Class	Age	Gender	Survived
0	Allen, Miss Elisabeth Walton	First	29	female	No
1	Allison, Miss Helen Loraine	First	2	female	Yes
2	Allison, Mr Hudson Joshua Creighton	First	30	male	Yes
3	Allison, Mrs Hudson JC (Bessie Waldo Daniels)	First	25	female	Yes
4	Allison, Master Hudson Trevor	First	0.92	male	No

In [204]:

```
#Merging the whole code

import pandas as pd
#Creating URL
url = 'C:/Users/Prerna/Desktop/ML_jupyter_notenooks/titanic.csv'

#Creating a dataframe
dataframe = pd.read_csv(url)

#Drop column with drop() with axis=1 for column
dataframe = dataframe.drop('SexCode',axis=1)

#Rename column PClass to Passenger class using rename()
dataframe = dataframe.rename(columns = {'PClass': 'Passenger Class',
                                        'Sex': 'Gender'})

#Replacing values of 1 with Yes and 0 with NO, and Pclass in whole dataframe.
dataframe = dataframe.replace([0,1,'1st','2nd','3rd'],
                              ['Yes','No','First','Second','Third'])

dataframe.head(5)

```

Out[204]:

	Name	Passenger Class	Age	Gender	Survived
0	Allen, Miss Elisabeth Walton	First	29	female	No
1	Allison, Miss Helen Loraine	First	2	female	Yes
2	Allison, Mr Hudson Joshua Creighton	First	30	male	Yes
3	Allison, Mrs Hudson JC (Bessie Waldo Daniels)	First	25	female	Yes
4	Allison, Master Hudson Trevor	First	0.92	male	No

Finding the Minimum, Maximum, Sum, Average, and Count

In [205]:

```python
#Creating URL
import pandas as pd
url = 'C:/Users/Prerna/Desktop/ML_jupyter_notenooks/titanic.csv'

#Creating a dataframe
dataframe = pd.read_csv(url)

#Printing the maximum with max() of age
print("Maximum Age:", dataframe['Age'].max())

#Printing the minimum with min() of age
print("Minimum Age:", dataframe['Age'].min())

#Printing the Sum with sum() of age
print("Total Age:", dataframe['Age'].sum())

#Printing the mean with mean() of age
print("Mean of Age:", dataframe['Age'].mean())

#Printing the median with median() of age
print("Median of Age:", dataframe['Age'].median())

#Printing the mode with mode() of age
print("Mode of Age:", dataframe['Age'].mode())

#Printing the count with count() of age
print("Count of Age:", dataframe['Age'].count())

#Printing the variance with var() of age
print("Variance of Age:", dataframe['Age'].var())

#Printing the standard deviation with std() of age
print("Standard Deviation of Age:", dataframe['Age'].std())
```

```
Maximum Age: 71.0
Minimum Age: 0.17
Total Age: 22980.88
Mean of Age: 30.397989417989415
Median of Age: 28.0
Mode of Age: 0    22.0
```

```
dtype: float64
Count of Age: 756
Variance of Age: 203.32047012439133
Standard Deviation of Age: 14.259048710359023
```

In [206]:

```
#Show counts
dataframe.count()
```

Out[206]:

```
Name        1313
PClass      1313
Age          756
Sex         1313
Survived    1313
SexCode     1313
dtype: int64
```

Finding Unique Values

In [207]:

```
#To find the unique values in a column with unique()

#Creating URL
url = 'C:/Users/Prerna/Desktop/ML_jupyter_notenooks/titanic.csv'

#Creating a dataframe
dataframe = pd.read_csv(url)

#select unique values
dataframe['PClass'].unique()

```

Out[207]:

```
array(['1st', '2nd', '*', '3rd'], dtype=object)
```

In [208]:
```
'''value_counts() will display all unique values with the number
   of times each value appears '''

dataframe['PClass'].value_counts()
```

Out[208]:
```
3rd    711
1st    322
2nd    279
*        1
Name: PClass, dtype: int64
```

In [209]:
```
# Show number of unique values with nunique()

dataframe['PClass'].nunique()
```

Out[209]:
```
4
```

Handling Missing Values

In [241]:

```
'''select missing values in a DataFrame. isnull and notnull return booleans
indicating whether a value is missing.'''
import pandas as pd
#Creating URL
url = 'C:/Users/Prerna/Desktop/ML_jupyter_notenooks/titanic.csv'

#Creating a dataframe
dataframe = pd.read_csv(url)

#Selecting the missing values from Age using isnull()

print(dataframe[dataframe['Age'].isnull()].head(5))

# Replacing Nan values t0 0 using fillna()
#The fillna() is used to replace NaN values with zeros in Pandas DataFrame

dataframe = dataframe.fillna(0)

#checking is any null value left
print(dataframe.isnull().sum())

#checking is any null value left
print(dataframe.isnull().values.any())

#You can also replace Nan values with zeros when loading the data
#dataframe = pd.read_csv(url).fillna(0)

```

```
                          Name PClass  Age     Sex  Survived  SexCode
12  Aubert, Mrs Leontine Pauline    1st  NaN  female         1        1
13      Barkworth, Mr Algernon H    1st  NaN    male         1        0
14            Baumann, Mr John D    1st  NaN    male         0        0
29       Borebank, Mr John James    1st  NaN    male         0        0
32            Bradley, Mr George    1st  NaN    male         1        0
Name        0
PClass      0
Age         0
Sex         0
Survived    0
SexCode     0
```

```
dtype: int64
False
```

Dropping Duplicates

drop_duplicates() defaults to only dropping rows that match perfectly across all columns.

In [211]:
```
#Creating URL
url ='C:/Users/Prerna/Desktop/ML_jupyter_notenooks/titanic_duplicatevalues.csv'

#Creating a dataframe
dataframe = pd.read_csv(url)

print("Original Data Frame:", len(dataframe))
print("Dataframe after dropping duplicate values:",
                    len(dataframe.drop_duplicates()))
```

```
Original Data Frame: 1323
Dataframe after dropping duplicate values: 1313
```

Grouping Rows by Values

groupby is one of the most powerful features in pandas.

groupby needs to be paired with some operation we want to apply to each group, such as calculating an aggregate statistic (e.g., mean, median, sum)

In [212]:

```
#Creating URL
url = 'C:/Users/Prerna/Desktop/ML_jupyter_notenooks/titanic.csv'

#Creating a dataframe
dataframe = pd.read_csv(url)

#Group by Gender
dataframe.groupby('Sex').count()
```

Out[212]:

Sex	Name	PClass	Age	Survived	SexCode
female	462	462	288	462	462
male	851	851	468	851	851

In [213]:

```
#Group by Gender
dataframe.groupby('Sex')['Name'].count()
```

Out[213]:

```
Sex
female    462
male      851
Name: Name, dtype: int64
```

In [214]:
```
#Group by multiple columns.
'''Find the count of males and females by their survival wise and
passenger class wise in titanic.'''

dataframe.groupby(['Sex','Survived','PClass'])['Name'].count()

```

Out[214]:
```
Sex     Survived  PClass
female  0         1st         9
                  2nd        13
                  3rd       132
        1         1st       134
                  2nd        94
                  3rd        80
male    0         *           1
                  1st       120
                  2nd       147
                  3rd       441
        1         1st        59
                  2nd        25
                  3rd        58
Name: Name, dtype: int64
```

Looping Over a Column

To iterate over every element in a column and apply some action You can treat a pandas column like any other sequence in Python.

In [215]:

```
#Creating URL
url = 'C:/Users/Prerna/Desktop/ML_jupyter_notenooks/titanic.csv'

#Creating a dataframe
dataframe = pd.read_csv(url)

for name in dataframe['Name'][0:len(dataframe)]:
    print(name.upper())
```

```
FARTHING, MR JOHN
FLEMING, MS MARGARET
FRANCATELLI, MS LAURA MABEL
FRY, MR RICHARD
GEIGER, MISS EMILY
GIGLIO, MR VICTOR
HARRINGTON, MR CHARLES
HARRISON, MR WILLIAM   HENRY
HASSAH, MR HAMAD
ICABAD (ICABOD), MS
KEEPING, MR EDWIN
KENCHEN, MS AMELIA
LEROY, MISS BERTHE
LESNEUR, MR GUSTAVE
MALONEY, MS
OLIVA, MLLE
PERICAULT, MS
RINGHINI, MR SANTE
ROBBINS, MR VICTOR
SEGESSER, MLLE EMMA
```

Applying a Function Over All Elements in a Column

Use apply() to apply a built-in or custom function on every element in a column

In [216]:

```
#Creating URL
url = 'C:/Users/Prerna/Desktop/ML_jupyter_notenooks/titanic.csv'

#Creating a dataframe
dataframe = pd.read_csv(url)

#Create a userdefined function
def uppercase(x):
    return x.upper()

#Apply custom function to the rows of the dataframe with apply()
dataframe['Name'].apply(uppercase)[0:len(dataframe)]
```

Out[216]:

```
0                             ALLEN, MISS ELISABETH WALTON
1                              ALLISON, MISS HELEN LORAINE
2                      ALLISON, MR HUDSON JOSHUA CREIGHTON
3           ALLISON, MRS HUDSON JC (BESSIE WALDO DANIELS)
4                            ALLISON, MASTER HUDSON TREVOR
5                                  ANDERSON, MR HARRY
6                         ANDREWS, MISS KORNELIA THEODOSIA
7                                  ANDREWS, MR THOMAS, JR
8            APPLETON, MRS EDWARD DALE (CHARLOTTE LAMSON)
9                                  ARTAGAVEYTIA, MR RAMON
10                               ASTOR, COLONEL JOHN JACOB
11       ASTOR, MRS JOHN JACOB (MADELEINE TALMADGE FORCE)
12                           AUBERT, MRS LEONTINE PAULINE
13                               BARKWORTH, MR ALGERNON H
14                                 BAUMANN, MR JOHN D
15         BAXTER, MRS JAMES (HELENE DELAUDENIERE CHAPUT)
16                               BAXTER, MR QUIGG EDMOND
17                                   BEATTIE, MR THOMSON
18                           BECKWITH, MR RICHARD LEONARD
19        BECKWITH, MRS RICHARD LEONARD (SALLIE MONYPENY)
20                                 BEHR, MR KARL HOWELL
21                                   BIRNBAUM, MR JAKOB
22                                BISHOP, MR DICKINSON H
23                  BISHOP, MRS DICKINSON H (HELEN WALTON)
24                BJORNSTRM-STEFFANSSON, MR MAURITZ HAKAN
```

```
25                           BLACKWELL, MR STEPHEN WEART
26                                    BLANK, MR HENRY
27                                BONNELL, MISS CAROLINE
28                               BONNELL, MISS ELIZABETH
29                               BOREBANK, MR JOHN JAMES
                                 ...
1283                  VESTROM, MISS HULDA AMANDA ADOLFINA
1284                                    VONK, MR JENKO
1285                                 WARE, MR FREDERICK
1286                            WARREN, MR CHARLES WILLIAM
1287                                   WAZLI, MR YOUSIF
1288                                   WEBBER, MR JAMES
1289                        WENNERSTROM, MR AUGUST EDVARD
1290                                 WENZEL, MR LINHART
1291                            WIDEGREN, MR CHARLES PETER
1292                              WIKLUND, MR JACOB ALFRED
1293                                    WILKES, MRS ELLEN
1294                                   WILLER, MR AARON
1295                                  WILLEY, MR EDWARD
1296                              WILLIAMS, MR HOWARD HUGH
1297                                   WILLIAMS, MR LESLIE
1298                                  WINDELOV, MR EINAR
1299                                     WIRZ, MR ALBERT
1300                                WISEMAN, MR PHILLIPPE
1301                             WITTEVRONGEL, MR CAMIEL
1302                                   YALSEVAC, MR IVAN
1303                                  YASBECK, MR ANTONI
1304                                 YASBECK, MRS ANTONI
1305                                  YOUSSEF, MR GERIOS
1306                                 ZABOUR, MISS HILENI
1307                                 ZABOUR, MISS TAMINI
1308                                  ZAKARIAN, MR ARTUN
1309                               ZAKARIAN, MR MAPRIEDER
1310                                     ZENNI, MR PHILIP
1311                                     LIEVENS, MR RENE
1312                                       ZIMMERMAN, LEO
Name: Name, Length: 1313, dtype: object
```

Applying a Function to Groups

You have grouped rows using groupby and want to apply a function to each group.

Combine groupby and apply.

In [217]:
```
#Creating URL
url = 'C:/Users/Prerna/Desktop/ML_jupyter_notenooks/titanic.csv'

#Creating a dataframe
dataframe = pd.read_csv(url)

# use groupby and apply

dataframe.groupby('Sex').apply(lambda x: x.count())
```

Out[217]:

Sex	Name	PClass	Age	Sex	Survived	SexCode
female	462	462	288	462	462	462
male	851	851	468	851	851	851

In [249]:

```
#DATA CLEANING CODE

import pandas as pd

#Creating URL
url = 'C:/Users/Prerna/Desktop/ML_jupyter_notenooks/titanic.csv'

#Creating a dataframe and replacing NAN values with 0
dataframe = pd.read_csv(url).fillna(0)


#checking is any null value left
print("Any null values in DataFrame:", dataframe.isnull().values.any())

#Drop duplicates.
dataframe.drop_duplicates()

#Drop column with drop() with axis=1 for column
dataframe = dataframe.drop('SexCode',axis=1)

#Rename column PClass to Passenger class using rename()
dataframe = dataframe.rename(columns = {'PClass': 'Passenger Class',
                                        'Sex': 'Gender'})

#Replacing Survived column 0 and 1 with No and Yes
dataframe['Survived'] = dataframe['Survived'].replace([0,1],
                                ['No','Yes'])

#Create a userdefined function
def uppercase(x):
    return x.upper()

#Apply custom function to the rows of the dataframe with apply()
dataframe['Name'] = dataframe['Name'].apply(uppercase)[0:len(dataframe)]

dataframe

# Writing the dataframe to a csv file using to_csv(filename)
dataframe.to_csv('Cleaneddata_titanic.csv')

```

```
Any null values in DataFrame: False
```

Out[249]:

	Name	Passenger Class	Age	Gender	Survived
0	ALLEN, MISS ELISABETH WALTON	1st	29.00	female	Yes
1	ALLISON, MISS HELEN LORAINE	1st	2.00	female	No
2	ALLISON, MR HUDSON JOSHUA CREIGHTON	1st	30.00	male	No
3	ALLISON, MRS HUDSON JC (BESSIE WALDO DANIELS)	1st	25.00	female	No
4	ALLISON, MASTER HUDSON TREVOR	1st	0.92	male	Yes
5	ANDERSON, MR HARRY	1st	47.00	male	Yes
6	ANDREWS, MISS KORNELIA THEODOSIA	1st	63.00	female	Yes
7	ANDREWS, MR THOMAS, JR	1st	39.00	male	No
8	APPLETON, MRS EDWARD DALE (CHARLOTTE LAMSON)	1st	58.00	female	Yes
9	ARTAGAVEYTIA, MR RAMON	1st	71.00	male	No
10	ASTOR, COLONEL JOHN JACOB	1st	47.00	male	No
11	ASTOR, MRS JOHN JACOB (MADELEINE TALMADGE FORCE)	1st	19.00	female	Yes
12	AUBERT, MRS LEONTINE PAULINE	1st	0.00	female	Yes
13	BARKWORTH, MR ALGERNON H	1st	0.00	male	Yes
14	BAUMANN, MR JOHN D	1st	0.00	male	No
15	BAXTER, MRS JAMES (HELENE DELAUDENIERE CHAPUT)	1st	50.00	female	Yes
16	BAXTER, MR QUIGG EDMOND	1st	24.00	male	No
17	BEATTIE, MR THOMSON	1st	36.00	male	No
18	BECKWITH, MR RICHARD LEONARD	1st	37.00	male	Yes
19	BECKWITH, MRS RICHARD LEONARD (SALLIE MONYPENY)	1st	47.00	female	Yes
20	BEHR, MR KARL HOWELL	1st	26.00	male	Yes
21	BIRNBAUM, MR JAKOB	1st	25.00	male	No
22	BISHOP, MR DICKINSON H	1st	25.00	male	Yes
23	BISHOP, MRS DICKINSON H (HELEN WALTON)	1st	19.00	female	Yes

	Name	Passenger Class	Age	Gender	Survived
24	BJORNSTRM-STEFFANSSON, MR MAURITZ HAKAN	1st	28.00	male	Yes
25	BLACKWELL, MR STEPHEN WEART	1st	45.00	male	No
26	BLANK, MR HENRY	1st	39.00	male	Yes
27	BONNELL, MISS CAROLINE	1st	30.00	female	Yes
28	BONNELL, MISS ELIZABETH	1st	58.00	female	Yes
29	BOREBANK, MR JOHN JAMES	1st	0.00	male	No
...	...	...	...	...	...
1283	VESTROM, MISS HULDA AMANDA ADOLFINA	3rd	14.00	female	No
1284	VONK, MR JENKO	3rd	22.00	male	No
1285	WARE, MR FREDERICK	3rd	0.00	male	No
1286	WARREN, MR CHARLES WILLIAM	3rd	0.00	male	No
1287	WAZLI, MR YOUSIF	3rd	0.00	male	No
1288	WEBBER, MR JAMES	3rd	0.00	male	No
1289	WENNERSTROM, MR AUGUST EDVARD	3rd	0.00	male	Yes
1290	WENZEL, MR LINHART	3rd	0.00	male	No
1291	WIDEGREN, MR CHARLES PETER	3rd	51.00	male	No
1292	WIKLUND, MR JACOB ALFRED	3rd	18.00	male	No
1293	WILKES, MRS ELLEN	3rd	45.00	female	Yes
1294	WILLER, MR AARON	3rd	0.00	male	No
1295	WILLEY, MR EDWARD	3rd	0.00	male	No
1296	WILLIAMS, MR HOWARD HUGH	3rd	0.00	male	No
1297	WILLIAMS, MR LESLIE	3rd	28.00	male	No
1298	WINDELOV, MR EINAR	3rd	21.00	male	No
1299	WIRZ, MR ALBERT	3rd	27.00	male	No
1300	WISEMAN, MR PHILLIPPE	3rd	0.00	male	No
1301	WITTEVRONGEL, MR CAMIEL	3rd	36.00	male	No

	Name	Passenger Class	Age	Gender	Survived
1302	YALSEVAC, MR IVAN	3rd	0.00	male	Yes
1303	YASBECK, MR ANTONI	3rd	27.00	male	No
1304	YASBECK, MRS ANTONI	3rd	15.00	female	Yes
1305	YOUSSEF, MR GERIOS	3rd	0.00	male	No
1306	ZABOUR, MISS HILENI	3rd	0.00	female	No
1307	ZABOUR, MISS TAMINI	3rd	0.00	female	No
1308	ZAKARIAN, MR ARTUN	3rd	27.00	male	No
1309	ZAKARIAN, MR MAPRIEDER	3rd	26.00	male	No
1310	ZENNI, MR PHILIP	3rd	22.00	male	No
1311	LIEVENS, MR RENE	3rd	24.00	male	No
1312	ZIMMERMAN, LEO	3rd	29.00	male	No

1313 rows × 5 columns

Concatenating DataFrames

You want to concatenate two DataFrames.

Use concat with axis=0 to concatenate along the row axis.

In [ ]:

```
#creating url
url = 'C:/Users/Prerna/Desktop/ML_jupyter_notenooks/titanic.csv'

#Creating a dataframe and replacing NAN values with 0
dataframe = pd.read_csv(url).fillna(0)

# Creating new data frames
df1 = dataframe.iloc[0:5]
print(df1)
df2 = dataframe.iloc[5:10]
print(df2)

# Concetenating data Frames by row axis
df3 = pd.concat([df1,df2],axis=0)
print(df3)
# Concetenating data Frames by column axis
df4 = pd.concat([df1,df2],axis=1)
print(df4)


```

Merging DataFrames

To inner join, use merge with the on parameter to specify the column to merge on.

In [260]:

```
import pandas as pd

# Create DataFrame
employee_data = {'employee_id': ['1', '2', '3', '4'],
'name': ['Amy Jones', 'Allen Keys', 'Alice Bees',
'Tim Horton']}
dataframe_employees = pd.DataFrame(employee_data, columns = ['employee_id',
'name'])
# Create DataFrame
sales_data = {'employee_id': ['3', '4', '5', '6'],
'total_sales': [23456, 2512, 2345, 1455]}
dataframe_sales = pd.DataFrame(sales_data, columns = ['employee_id',
'total_sales'])
# Merge DataFrames using inner join
print(pd.merge(dataframe_employees, dataframe_sales, on='employee_id'))

# Merge DataFrames using outer join. Values for how = outer,left,right
print(pd.merge(dataframe_employees, dataframe_sales, on='employee_id', how='outer'))


```

```
  employee_id        name  total_sales
0           3  Alice Bees        23456
1           4  Tim Horton         2512
  employee_id        name  total_sales
0           1   Amy Jones          NaN
1           2  Allen Keys          NaN
2           3  Alice Bees      23456.0
3           4  Tim Horton       2512.0
4           5         NaN       2345.0
5           6         NaN       1455.0
```