Practical 5: Data Wrangling

Data Wrangling

[34]:
```
import pandas as pd
```

[35]:
```
data ={"Name" :["Nikhil","Nikita","Jay","Anuj","Ravi","Natasha","Riya"],
       "Age":[17,19,16,18,17,18,19],
       "Gender" :['M',"F","M","M","M","F","F"],
       "Marks" : [90,78,'NaN',7,65,'NaN',71]
        }
df=pd.DataFrame(data)
```

[36]:
```
df
```

[36]:

	Name	Age	Gender	Marks
0	Nikhil	17	M	90
1	Nikita	19	F	78
2	Jay	16	M	NaN
3	Anuj	18	M	7

Dealing with missing values in Python

[37]:
```
c=avg=0
for i in df['Marks']:
    if str(i).isnumeric():
        c+=1
        avg += i
avg/=c
# replace missing value
df=df.replace(to_replace="NaN",value=avg)
df
```

[37]:

	Name	Age	Gender	Marks
0	Nikhil	17	M	90.0
1	Nikita	19	F	78.0
2	Jay	16	M	62.2
3	Anuj	18	M	7.0
4	Ravi	17	M	65.0
5	Natasha	18	F	62.2
6	Riya	19	F	71.0

Data Replacing in Data Wrangling

[5]:
```
# Data Replacing
df['Gender'] =df['Gender'].map({'M':0,"F":1}).astype(float)
df
```

[5]:

	Name	Age	Gender	Marks
0	Nikhil	17	0.0	90
1	Nikita	19	1.0	78
2	Jay	16	0.0	311
3	Anuj	18	0.0	7
4	Ravi	17	0.0	65
5	Natasha	18	1.0	311
6	Riya	19	1.0	71

Filtering data in Data Wrangling

[6]:
```
# Filteing data
df=df[df['Marks']>=75].copy()
# Remove age column from data
df.drop('Age',axis=1,inplace=True)
```

[7]:
```
df
```

[7]:

	Name	Gender	Marks
0	Nikhil	0.0	90
1	Nikita	1.0	78
2	Jay	0.0	311
5	Natasha	1.0	311

Data Wrangling Using Merge Operation

[8]:
```
details = pd.DataFrame({
    'ID': [101, 102, 103, 104, 105, 106,107, 108, 109, 110],
    'NAME': ['Jagroop', 'Praveen', 'Harjot', 'Pooja', 'Rahul', 'Nikita','Saurabh', 'Ayush', 'Dolly', "Mohit"],
    'BRANCH': ['CSE', 'CSE', 'CSE', 'CSE', 'CSE', 'CSE', 'CSE', 'CSE', 'CSE', 'CSE']
})

fees_status = pd.DataFrame({
    'ID': [101, 102, 103, 104, 105, 106, 107, 108, 109, 110],
    'PENDING': ['5000', '250', 'NIL', '9000', '15000', 'NIL','4500', '1800', '250', 'NIL']
})
```

[9]:
```
print(pd.merge(details, fees_status, on='ID'))
```

```
    ID     NAME BRANCH PENDING
0  101  Jagroop    CSE    5000
1  102  Praveen    CSE     250
2  103   Harjot    CSE     NIL
3  104    Pooja    CSE    9000
4  105    Rahul    CSE   15000
5  106   Nikita    CSE     NIL
6  107  Saurabh    CSE    4500
7  108    Ayush    CSE    1800
8  109    Dolly    CSE     250
```

Data Wrangling Using Grouping Method

[10]:
```
car_selling_data = {'Brand': ['Maruti', 'Maruti', 'Maruti',
                              'Maruti', 'Hyundai', 'Hyundai',
                              'Toyota', 'Mahindra', 'Mahindra',
                              'Ford', 'Toyota', 'Ford'],
                    'Year':  [2010, 2011, 2009, 2013,
                              2010, 2011, 2011, 2010,
                              2013, 2010, 2010, 2011],
                    'Sold': [6, 7, 9, 8, 3, 5,
                             2, 8, 7, 2, 4, 2]}
df = pd.DataFrame(car_selling_data)
df
```

[10]:

	Brand	Year	Sold
0	Maruti	2010	6
1	Maruti	2011	7
2	Maruti	2009	9
3	Maruti	2013	8
4	Hyundai	2010	3
5	Hyundai	2011	5
6	Toyota	2011	2
7	Mahindra	2010	8

Group the data when year = 2010

[11]:
```
grouped = df.groupby('Year')
print(grouped.get_group(2010))
```

```
       Brand  Year  Sold
0     Maruti  2010     6
4    Hyundai  2010     3
7   Mahindra  2010     8
9       Ford  2010     2
10    Toyota  2010     4
```

[12]:
```
student_data = {'Name': ['Amit', 'Praveen', 'Jagroop',
                         'Rahul', 'Vishal', 'Suraj',
                         'Rishab', 'Satyapal', 'Amit',
                         'Rahul', 'Praveen', 'Amit'],

                'Roll_no': [23, 54, 29, 36, 59, 38,
                            12, 45, 34, 36, 54, 23],

                'Email': ['xxxx@gmail.com', 'xxxxxx@gmail.com',
                          'xxxxxx@gmail.com', 'xx@gmail.com',
                          'xxxx@gmail.com', 'xxxxx@gmail.com',
                          'xxxxx@gmail.com', 'xxxxx@gmail.com',
                          'xxxxx@gmail.com', 'xxxxxx@gmail.com',
                          'xxxxxxxxxx@gmail.com', 'xxxxxxxxxx@gmail.com']}
df = pd.DataFrame(student_data)
df
```

[12]:

	Name	Roll_no	Email
0	Amit	23	xxxx@gmail.com
1	Praveen	54	xxxxxx@gmail.com
2	Jagroop	29	xxxxxx@gmail.com

Removing Duplicate data from the Dataset using Data wrangling:

[13]:
```
non_duplicate = df[~df.duplicated('Roll_no')]

non_duplicate
```

[13]:

	Name	Roll_no	Email
0	Amit	23	xxxx@gmail.com
1	Praveen	54	xxxxxx@gmail.com
2	Jagroop	29	xxxxxx@gmail.com
3	Rahul	36	xx@gmail.com
4	Vishal	59	xxxx@gmail.com
5	Suraj	38	xxxxx@gmail.com
6	Rishab	12	xxxxx@gmail.com
7	Satyapal	45	xxxxx@gmail.com
8	Amit	34	xxxxx@gmail.com

Concatenation of Two Datasets

[14]:
```
data1 = {'Name':['Jai', 'Princi', 'Gaurav', 'Anuj'],
        'Age':[27, 24, 22, 32],
        'Address':['Nagpur', 'Kanpur', 'Allahabad', 'Kannuaj'],
        'Qualification':['Msc', 'MA', 'MCA', 'Phd'],
        'Mobile No': [97, 91, 58, 76]}

data2 = {'Name':['Gaurav', 'Anuj', 'Dhiraj', 'Hitesh'],
        'Age':[22, 32, 12, 52],
        'Address':['Allahabad', 'Kannuaj', 'Allahabad', 'Kannuaj'],
        'Qualification':['MCA', 'Phd', 'Bcom', 'B.hons'],
        'Salary':[1000, 2000, 3000, 4000]}
df = pd.DataFrame(data1,index=[0, 1, 2, 3])
df1 = pd.DataFrame(data2, index=[2, 3, 6, 7])
```

[15]:
```
res = pd.concat([df, df1])
```

[15]:
```
res = pd.concat([df, df1])
```

[16]:
```
res
```

[16]:

	Name	Age	Address	Qualification	Mobile No	Salary
0	Jai	27	Nagpur	Msc	97.0	NaN
1	Princi	24	Kanpur	MA	91.0	NaN
2	Gaurav	22	Allahabad	MCA	58.0	NaN
3	Anuj	32	Kannuaj	Phd	76.0	NaN
2	Gaurav	22	Allahabad	MCA	NaN	1000.0
3	Anuj	32	Kannuaj	Phd	NaN	2000.0
6	Dhiraj	12	Allahabad	Bcom	NaN	3000.0
7	Hitesh	52	Kannuaj	B.hons	NaN	4000.0