Pandas-DataFrame

Sunday, October 8, 2023 5:40 PM

A pandas DataFrame is a two-dimensional data structure that can be thought of as a spreadsheet.

```
import numpy as np
import pandas as pd
```

Creating DataFrame

```
# using lists
student_data = [
   [100,80,10],
   [90,70,7],
   [120,100,14],
   [80,50,2]
]
pd.DataFrame(student_data,columns=['iq','marks','package'])

# using dicts

student_dict = {
   'name':['nitish','ankit','rupesh','rishabh','amit','ankita'],
   'iq':[100,90,120,80,0,0],
   'marks':[80,70,100,50,0,0],
   'package':[10,7,14,2,0,0]
}

students = pd.DataFrame(student_dict)
students.set_index('name',inplace=True)
students

# using read_csv

movies = pd.read_csv('movies.csv')
movies

ipl = pd.read_csv('ipl-matches.csv')
ipl
```

DataFrame Attributes and Methods

```
# shape
movies.shape
ipl.shape

# dtypes
movies.dtypes
ipl.dtypes
```

```
# index
movies.index
ipl.index

# columns
movies.columns
ipl.columns
student.columns

# values
student.values
ipl.values

# head and tail
movies.head(2)
ipl.tail(2)

# sample
ipl.sample(5)

# info
movies.info()
ipl.info()

# describe
movies.describe()
ipl.describe()

# isnull
movies.isnull().sum()

# duplicated
movies.duplicated().sum()
students.duplicated().sum()

# rename
students

students.rename(columns={'marks':'percent','package':'lpa'},inplace=True)
```

Math Methods

```
# sum -> axis argument
students.sum(axis=0)
students.mean(axis=1)
students.var()
```

Selecting cols from a DataFrame

```
# single cols
```

```
movies['title_x']
ipl['Venue']

# multiple cols
movies[['year_of_release','actors','title_x']]
ipl[['Team1','Team2','WinningTeam']]
```

Selecting rows from a DataFrame

- **iloc** - searches using index positions
- **loc** - searches using index labels

```
# single row
movies.iloc[5]

# multiple row
movies.iloc[:5]

# fancy indexing
movies.iloc[[0,4,5]]

# loc
students

students.loc['nitish']
students.loc['nitish':'rishabh':2]
students.loc[['nitish','ankita','rupesh']]
students.iloc[[0,3,4]]
```

Selecting both rows and cols

```
movies.iloc[0:3,0:3]
movies.loc[0:2,'title_x':'poster_path']
```

Filtering a DataFrame

```
ipl.head(2)

# find all the final winners
mask = ipl['MatchNumber'] == 'Final'
new_df = ipl[mask]
new_df[['Season','WinningTeam']]

ipl[ipl['MatchNumber'] == 'Final'][['Season','WinningTeam']]

# how many super over finishes have occured
ipl[ipl['SuperOver'] == 'Y'].shape[0]

# how many matches has csk won in kolkata
ipl[(ipl['City'] == 'Kolkata') & (ipl['WinningTeam'] == 'Chennai Super Kings')].shape[0]
```

```
# toss winner is match winner in percentage
(ipl[ipl['TossWinner'] == ipl['WinningTeam']].shape[0]/ipl.shape[0])*100

# movies with rating higher than 8 and votes>10000
movies[(movies['imdb_rating'] > 8.5) & (movies['imdb_votes'] > 10000)].shape[0]

# Action movies with rating higher than 7.5
# mask1 = movies['genres'].str.split('|').apply(lambda x:'Action' in x)
mask1 = movies['genres'].str.contains('Action')
mask2 = movies['imdb_rating'] > 7.5

movies[mask1 & mask2]

# write a function that can return the track record of 2 teams against each other
```

Adding new cols

```
# completely new
movies['Country'] = 'India'
movies.head()

# from existing ones
movies.dropna(inplace=True)
movies['lead actor'] = movies['actors'].str.split('|').apply(lambda x:x[0])
movies.head()
movies.info()
```

Important DataFrame Functions

```
# astype
ipl.info()

ipl['ID'] = ipl['ID'].astype('int32')

ipl.info()

# ipl['Season'] = ipl['Season'].astype('category')
ipl['Team1'] = ipl['Team1'].astype('category')
ipl['Team2'] = ipl['Team2'].astype('category')

ipl.info()

# value_counts
# find which player has won most potm -> in finals and qualifiers
# Toss decision plot
# how many matches each team has played
# sort_values -> ascending -> na_position -> inplace -> multiple cols
```