K-Digital Training

codingOn

C# WPF

Dataframe

DataFrame

- Python의 pandas 라이브러리의 DataFrame과 유사한 데이터 구조
- 데이터 조작 및 분석을 위해서 설계
- 표 형식으로 데이터를 처리하는 방법 제공

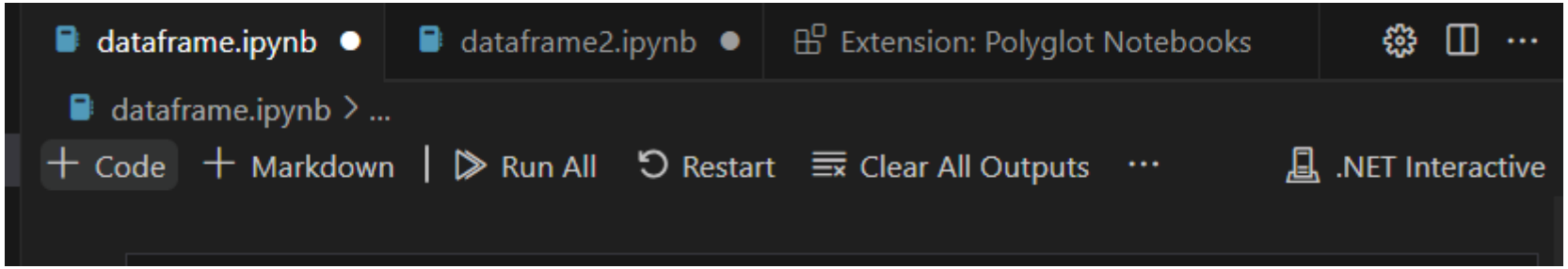
- **Microsoft.Data.Analysis 에 포함**

DataFrame 설치방법

- VS code에 설치
  - **Polyglot Notebooks** extension 설치

DataFrame 설치방법

- VS code에 설치
  - 커널을 **.NET Interactive** 선택



DataFrame 설치방법

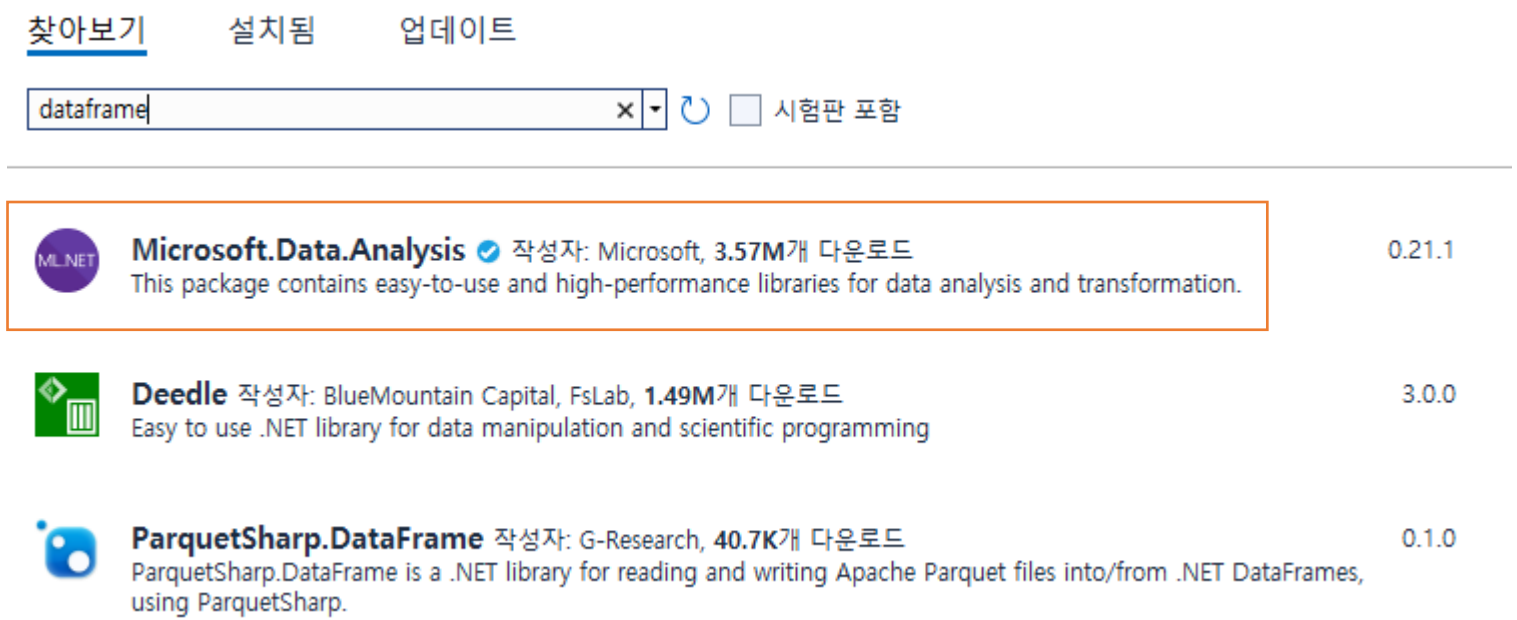
- Jupiter Notebook에 설치
  - dotnet tool install -g Microsoft.dotnet-interactive
  - 실행 안될 경우에
    - dotnet nuget add source https://api.nuget.org/v3/index.json -n nuget.org
    - 위 명령어 실행 후 다시 실행
  - dotnet interactive jupyter install
  - mkdir -p C:\Users\SPREATICS\AppData\Roaming\jupyter\kernels

DataFrame 설치방법

- Jupiter Notebook에 설치

DataFrame 설치방법

- NuGet에서 dataframe 검색하여 설치



DataFrame

- 패키지 설치

```
1 #r "nuget:Microsoft.Data.Analysis,0.21.0“
```

- Dataframe import

```
3 using Microsoft.Data.Analysis;
```

DataFrame

- csv 로드

```
 5 using System.IO;
 6 using System.Linq;
 7
 8 // Define data path
 9 var dataPath = Path.GetFullPath(@"prices.csv");
10
11 // Load the data into the data frame
12 var dataFrame = DataFrame.LoadCsv(dataPath);
```

DataFrame

- Data 확인
  - 최상단에는 열의 이름
  - 그 아래에는 각 열의 실제 값

```
dataFrame
```

index	Id	Size	HistoricalPrice	CurrentPrice
0	1	600f	100000	170000
1	2	1000f	200000	225000
2	3	1000f	126000	195000
3	4	850f	150000	205000
4	5	900f	155000	210000
5	6	550f	99000	180000

DataFrame

- Description()
  - Data의 요약 제공
  - 길이, 최대, 최소, 평균 값
- Info()
  - DataType, 길이 정보 제공

```
dataFrame.Description()
```

index	Description	Id	HistoricalPrice	CurrentPrice
0	Length (excluding null values)	6	6	6
1	Max	6	200000	225000
2	Min	1	99000	170000
3	Mean	3.5	138333.33	197500

```
dataFrame.Info()
```

index	Info	Id	Size	HistoricalPrice	CurrentPrice
0	DataType	System.Single	System.String	System.Single	System.Single
1	Length (excluding null values)	6	6	6	6

ScottPlot

- 패키지 설치

```
18 #r "nuget:ScottPlot, 5.0.36"
```

- 그래프 출력 설정

```
20 using Microsoft.DotNet.Interactive.Formatting;
21 Formatter.Register(typeof(ScottPlot.Plot), (p, w) =>
22     w.Write(((ScottPlot.Plot)p).GetImageHtml(400, 300)), HtmlFormatter.MimeType);
```

ScottPlot

- 패키지 import

```
24 using ScottPlot;
25 using Microsoft.Data.Analysis;
26 using System;
27 using System.Linq;
```

- 랜덤 데이터 생성 및 정렬

```
29 ScottPlot.RandomDataGenerator generator = new ScottPlot.RandomDataGenerator();
30 double[] tempData = generator.RandomWalk(10);
31 double[] humidData = generator.RandomWalk(10);
32
33 Array.Sort(tempData);
34 Array.Sort(humidData);
```

ScottPlot

- DataFrame 생성

```
36 DoubleDataFrameColumn colTemp = new DoubleDataFrameColumn("Temperature", tempData);
37 DoubleDataFrameColumn colHumid = new DoubleDataFrameColumn("Humidity", humidData);
38 DataFrameColumn[] columns = { colTemp, colHumid };
39 DataFrame df = new DataFrame(columns);
```

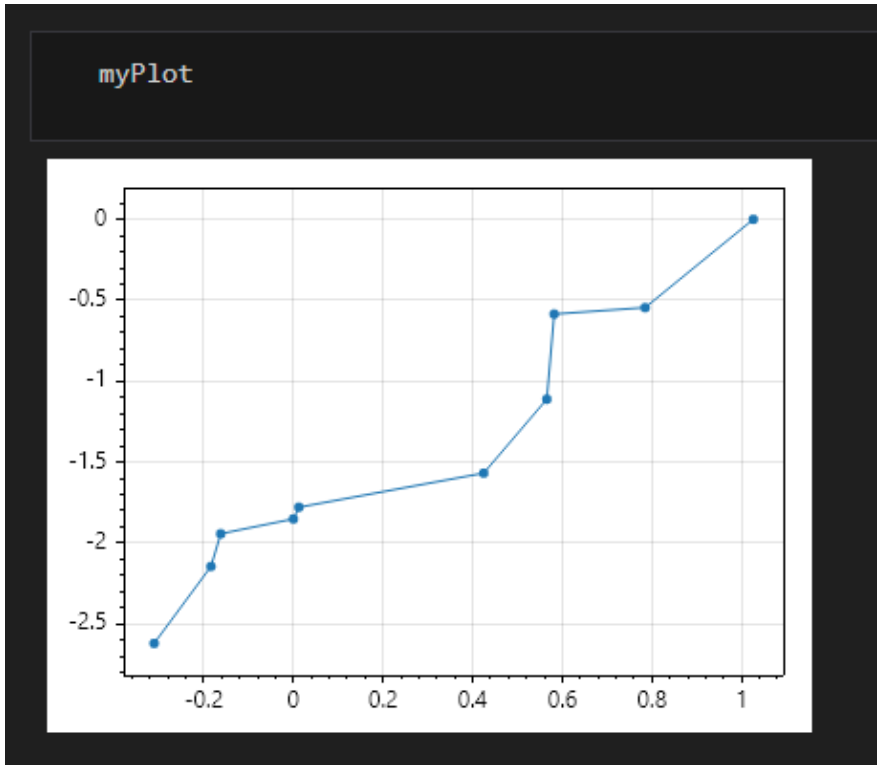
- ScottPlot 축 설정

```
41 double[] xs = df["Temperature"].Cast<double>().ToArray();
42 double[] ys = df["Humidity"].Cast<double>().ToArray();
43
44 ScottPlot.Plot myPlot = new();
45 myPlot.Add.Scatter(xs, ys);
```

ScottPlot

- 그래프 표시

```
// myPlot.SavePng("quickstart.png", 400, 300); // 이미지로 저장
```



DataFrame 상세

- DataFrame 생성

```
using Microsoft.Data.Analysis;

string[] names = { "Oliver", "Charlotte", "Henry", "Amelia", "Owen" };
int[] ages = { 23, 19, 42, 64, 35 };
double[] heights = { 1.91, 1.62, 1.72, 1.57, 1.85 };

DataFrameColumn[] columns = {
    new StringDataFrameColumn("Name", names),
    new PrimitiveDataFrameColumn<int>("Age", ages),
    new PrimitiveDataFrameColumn<double>("Height", heights),
};

DataFrame df = new(columns);
```

DataFrame 상세

- DataFrameColumn
  - DataFrame의 각 컬럼
  - StringDataFrameColumn
    - String으로 이루어진 Data
  - PrimitiveDataFrameColumn<T>
    - int, float등 기본데이터형으로 이루어진 Data
    - Int16DataFrameColumn
    - CharDataFrameColumn

DataFrame 상세

- 행 추가

```
List<KeyValuePair<string, object>> newRowData = new()
{
    new KeyValuePair<string, object>("Name", "Scott"),
    new KeyValuePair<string, object>("Age", 36),
    new KeyValuePair<string, object>("Height", 1.65),
};

df.Append(newRowData, inPlace: true);
```

  - Append 함수 이용
    - inPlace : true ->기본 dataframe 변경, false 새로운 dataframe 리턴

DataFrame 상세

- 열 추가

```
int[] weights = { 123, 321, 111, 121, 131, 141 };
PrimitiveDataFrameColumn<int> weightCol = new("Weight", weights);
df.Columns.Add(weightCol);
```

  - Columns.Add() 함수 이용

DataFrame 상세

- 필터 기능
  - Age 열 중에 값이 30이상만 필터링

```
df.Filter(df["Age"].ElementwiseGreaterThan(30))
```

index	Name	Age	Height
0	Henry	42	1.72
1	Amelia	64	1.57
2	Owen	35	1.85
3	Scott	36	1.65

DataFrame 상세

- 필터 기능(Linq 이용)
  - Age 열 중에 값이 30이상인 행만 필터링

```
var filteredRows = df.Rows
                   .Where(row => (int)row["Age"] >= 30)
                   .ToList();
filteredRows
```

index	value
0	▶ [Henry, 42, 1.72, 111]
1	▶ [Amelia, 64, 1.57, 121]
2	▶ [Owen, 35, 1.85, 131]
3	▶ [Scott, 36, 1.65, 141]

DataFrame 상세

- 필터 기능(Linq 이용)
  - Age 열 중에 값이 30이상인 행만 필터링 후 특정 열 선택

```
var selectedNames = df.Rows
                      .Where(row => (int)row["Age"] >= 30)
                      .Select(row => row["Name"].ToString())
                      .ToList();
selectedNames
```

```
[ Henry, Amelia, Owen, Scott ]
```

DataFrame 상세

- 정렬 기능
  - “Name” 열 오름차순 정렬

```
df.OrderBy("Name")
```

index	Name	Age	Height
0	Amelia	64	1.57
1	Charlotte	19	1.62
2	Henry	42	1.72
3	Oliver	23	1.91
4	Owen	35	1.85
5	Scott	36	1.65

DataFrame 상세

- 수학 연산
  - 특정 열의 값을 이용하여 새로운 열 생성

```
DataFrameColumn iqCol = df["Age"] * df["Height"] * 1.5;

double[] iqs = Enumerable.Range(0, (int)iqCol.Length)
    .Select(x => (double)iqCol[x])
    .ToArray();

df.Columns.Add(new PrimitiveDataFrameColumn<double>("IQ", iqs));
df
```

index	Name	Age	Height	Weight	IQ
0	Oliver	23	1.91	123	65.895
1	Charlotte	19	1.62	321	46.17
2	Henry	42	1.72	111	108.35999999999999
3	Amelia	64	1.57	121	150.72
4	Owen	35	1.85	131	97.125
5	Scott	36	1.65	141	89.1

DataFrame 상세

- 통계 작업

```
foreach (DataFrameColumn col in df.Columns.Skip(1))
{
    // warning: additional care must be taken for datasets which contain null
    double[] values = Enumerable.Range(0, (int)col.Length).Select(x => Convert.ToDouble(col[x])).ToArray();
    (double mean, double std) = MeanAndStd(values);
    Console.WriteLine($"{col.Name} = {mean} +/- {std:N3} (n={values.Length})");
}
```

```
Age = 36.5 +/- 14.592 (n=6)
Height = 1.72 +/- 0.123 (n=6)
Weight = 158 +/- 73.473 (n=6)
IQ = 92.895 +/- 32.983 (n=6)
```

DataFrame 상세

- 저장

```
DataFrame.SaveCsv(df, "result.csv", ',');
```

실습1. Dataframe

https://www.kaggle.com/code/alexisbcook/hello-seaborn/data?select=fifa.csv

fifa.csv 파일을 다운로드 후 데이터 분석

1. 전체 기간에서 각 나라의 평균 순위
2. 2000년의 독일 평균 순위
3. 아르헨티나와 브라질의 순위 그래프를 한 그래프에 표시
4. 2001년 이후 프랑스 순위 그래프와 추세선