

F#4DS

Загрузка и визуализация

[@anton_shumakov](#)

F# или Python? @fsharp_chat

Привет,

Подскажите, пожалуйста, в какой ситуации, с базовым знанием математики, python и F#, человек может выбрать для анализа данных F#? Мб там есть какие-то киллер фиши или крутые либы (кроме тайп провайдеров), которые недостаточно хорошо сделаны на python?

Если питон устраивает, особых причин нет

А в целом, если Питон устраивает, смысла использовать F# в этой нише нет.

Почему выбрать фарш, а не питон?

1. Есть дотнет приложение куда нужно интегрировать какой-то дата анализ и интеграция через отдельный сервис не проходит по каким-то причинам. 2. Есть команда которая хорошо знает дотнет и знает фарш, но совсем не знает питон, а какой-то анализ нужно делать. В остальном на питоне скорее всего будет больше библиотек и больше документации, в остальном дело вкуса

Особенности F#

1. Поставщики типов
2. Функциональность
3. .NET

Настройка удаленного окружения

Binder



```
[1]: printfn "%A" "HI!"
```

```
"HI!"
```

Настройка локального окружения с .NET Interactive

```
> python -m pip install jupyterlab  
> dotnet tool install -g dotnet-try  
> dotnet try jupyter install  
> jupyter lab
```



Notebook



Python 3



.NET (C#)



.NET (F#)

[.NET Core with Jupyter Notebooks | devblogs.microsoft](https://devblogs.microsoft.com/dotnet/core-with-jupyter-notebooks/)

Введение в Type Providers

```
type Person =  
    { Name:string; Age:int; Countries:string list; }
```

Стало:

```
[2]: #r "nuget: FSharp.Data"  
      open FSharp.Data
```

Installed package FSharp.Data version 4.1.1

```
[ ]: let wb = WorldBankData.GetDataContext()  
      wb.Countries.Ru|
```

Russian Federation

Resource rich Sub-Saharan Africa countries

IBRD countries classified as high income

Type Providers для CSV

```
columnInt,columnString,columnFloat  
1,asad,1.2  
7,gfsgf,-12.2  
4,k1k1k,3.0
```

```
type Stocks = CsvProvider<"data.csv">  
Stocks.GetSample()
```

Rows	Headers
[(1, asdad, 1.2), (7, gfsgf, -12.2), (4, k1k1k, 3.0)]	{ Some(System.String[]): Value: [columnInt, columnString, columnFloat] }

Работа Type Providers с сетью

```
[ ]: type GitHub = JsonProvider<"https://api.github.com/repos/dotnet/coreclr/issues">

let topRecentlyUpdatedIssues =
    GitHub.GetSamples()
    |> Seq.sortBy (fun x -> x.UpdatedAt)
    |> Seq.truncate 5

for issue in topRecentlyUpdatedIssues do
    printfn "%d %s" issue.Id issue.Title
```

Title

```
.....^....[...^q.P.w.....4.K.D;..*.,.+..0./.....$.#.(.'..
.
.....=<.5./..
...T.....api.github.com.
.....
.....#.....A...=...%1
X.....o...v....K+.DOWNGRD...+.....5..1.
...
...0...0.....h.u...(.....Y.0
..*.H.=...0g1.0...U....US1.0...U.
```


Исследование API

<https://freegeoip.app/json/>

1. Какие есть поля в ответе?
2. В какой вы стране?
3. Как обработается ответ от <https://xkcd.com/info.0.json> ?

Решение

1.

```
type GeoIp = JsonProvider<"https://freegeoip.app/json/">  
let geoIp = GeoIp.GetSample()  
geoIp.CountryName
```

2.

United States

3.

```
[-13]: GeoIp.Load("https://xkcd.com/info.0.json").CountryName
```

```
[-37]:
```

Применение Type Providers для смены формата

data.csv:

```
Title,Author,Average Rating
Readme,Octocat,10
Robots.txt,Admin,3
Name,Author Name,3
Name2,Name2 Author,7
Some,Another,1
Another,Name,1
Titorial,Kio,10
```

Применение Type Providers для смены формата

```
type Data = CsvProvider<"data.csv">
type BookTypes = JsonProvider<"sample.json">

Data.GetSample().Rows
|> Seq.sortByDescending (fun book -> ???)
|> Seq.take 5
|> Seq.map (fun book -> BookTypes.Root(???, ???))
BookTypes.GetSample()
```

1. Дополнить код так, чтобы он выделял 5 лучших по рейтингу книг и их авторов
2. Создать `sample.json` для поставщика JSON

Применение Type Providers для смены формата

```
[ ( Readme, Octocat, 10 ), ( Robots.txt, Admin, 3 ), ( Name, Author  
    Name, 3 ), ( Name2, Name2 Author, 7 ), ( Some, Another, 1 ), ( { Some(System.String[]): Value: [ Title,  
        Another, Name, 1 ), ( Titorial, Kio, 10 ) ] }  
                                Author, Average Rating ] }
```

```
{ "author": "Octocat", "rating": 10 }
```

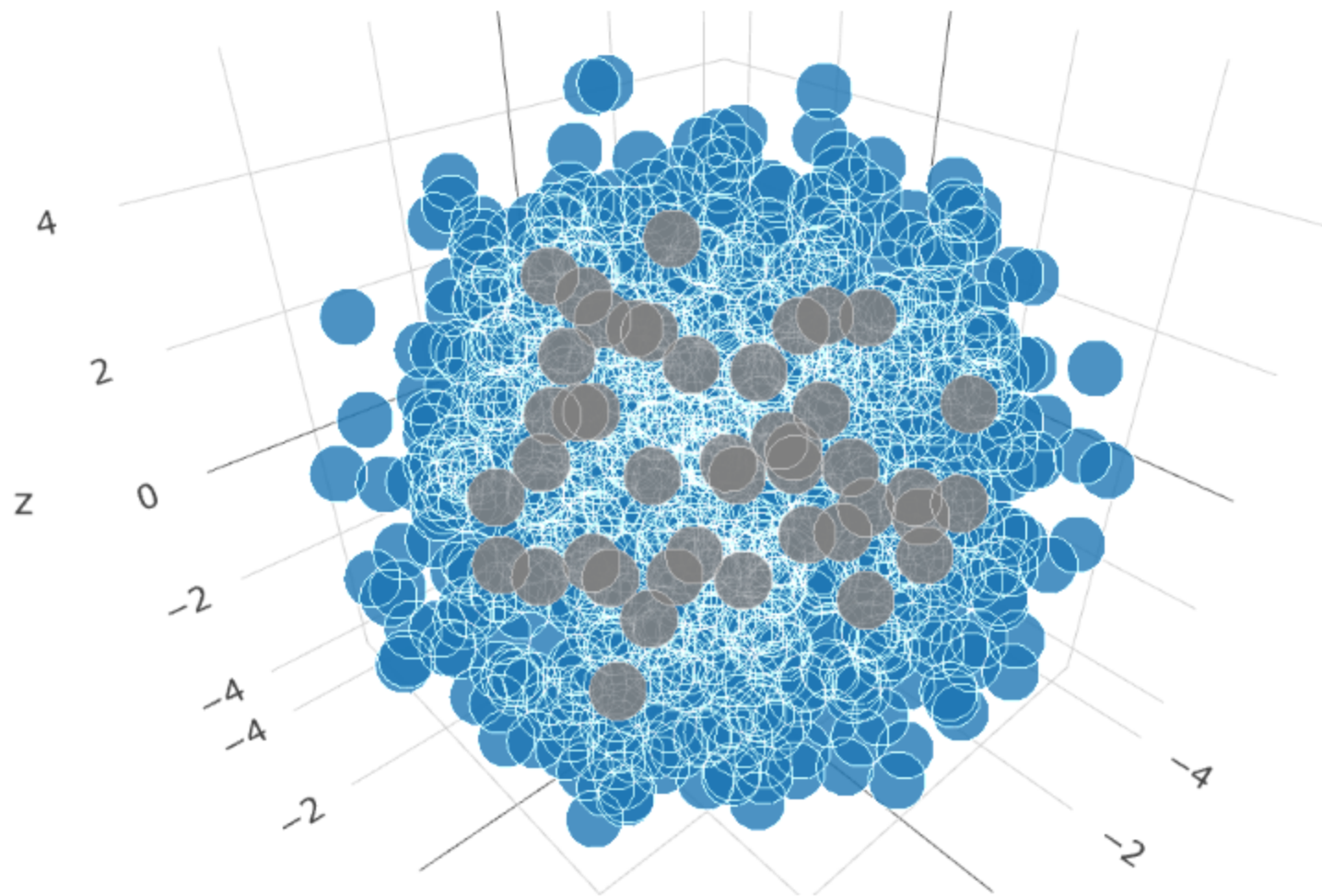
```
{ "author": "Kio", "rating": 10 }
```

```
{ "author": "Name2 Author", "rating":  
    7 }
```

```
{ "author": "Admin", "rating": 3 }
```

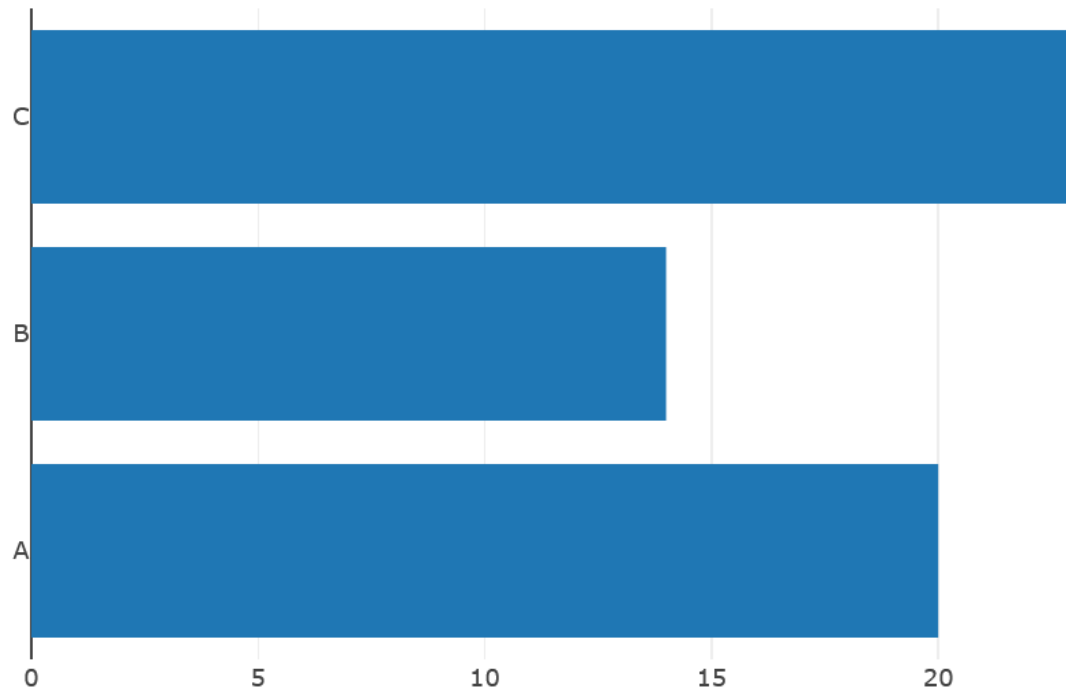
```
{ "author": "Author Name", "rating": 3  
    }
```

Визуализация с XPlot.Plotly

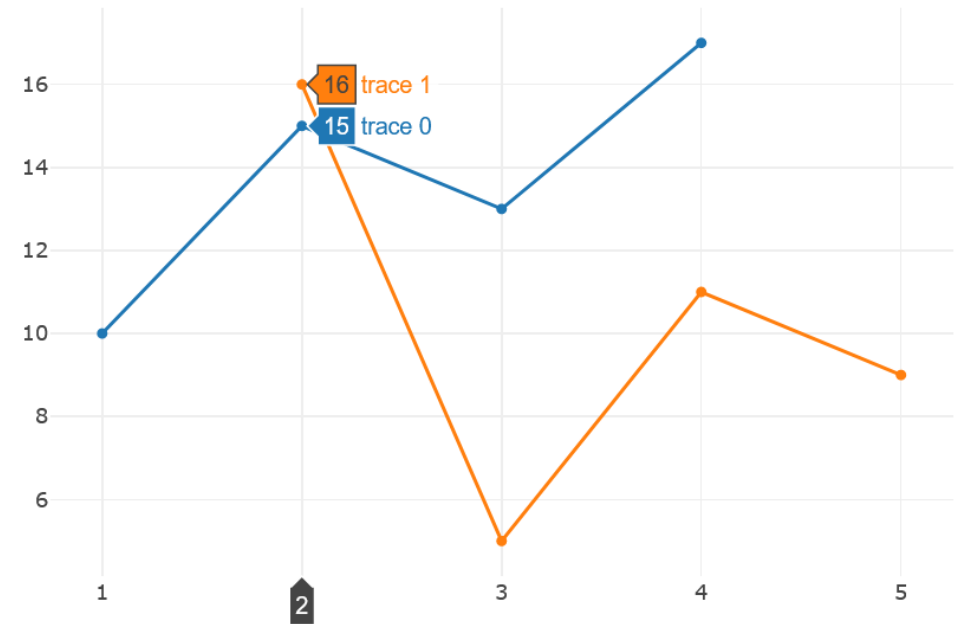


Виды графиков в XPlot.Plotly

Гистограмма



График



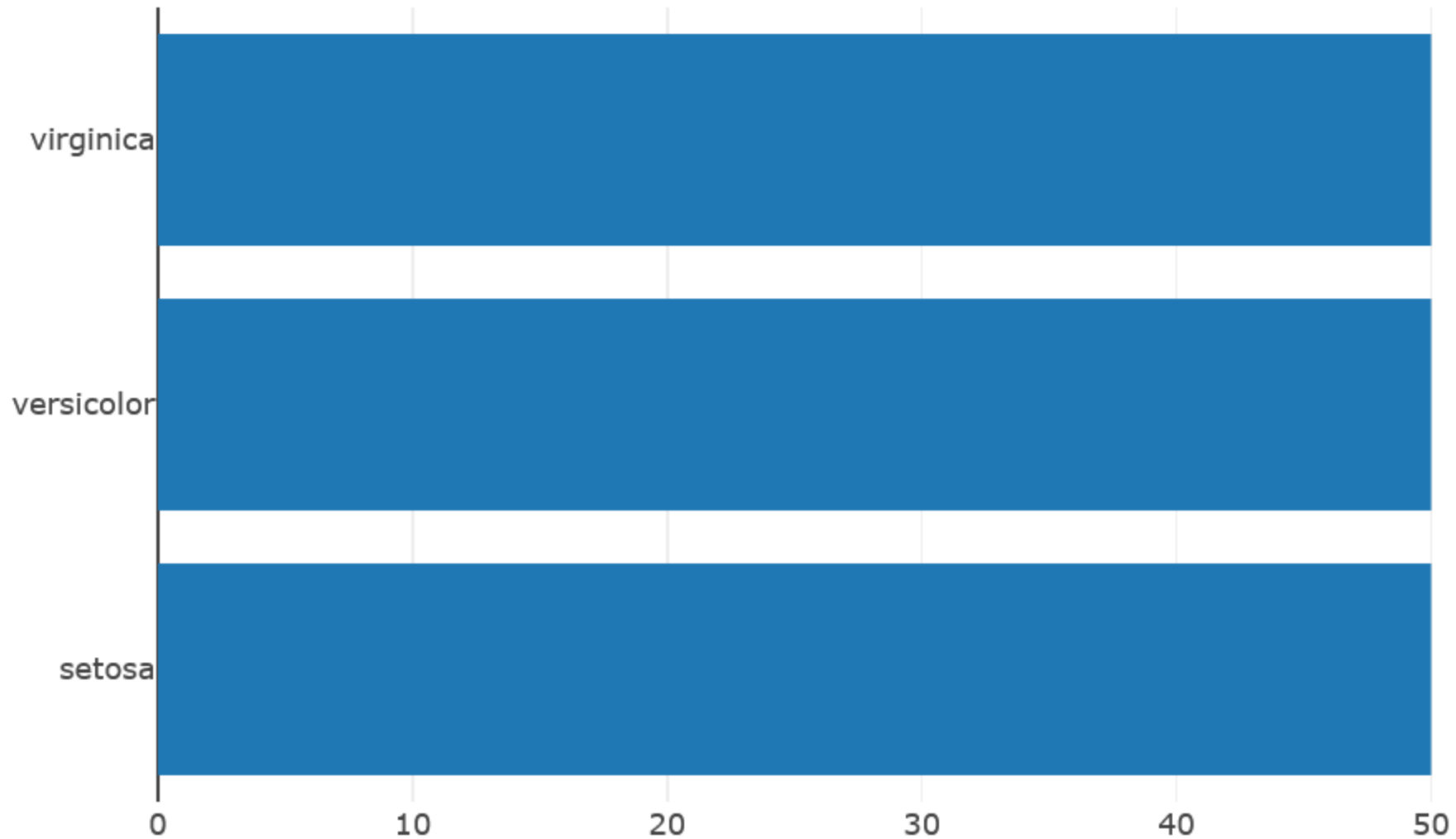
Анализ данных

```
printfn "%A" iris.Headers  
iris.Rows
```

```
Some [|"sepal_length"; "sepal_width"; "petal_length"; "petal_width"; "species"|]
```

<i>index</i>	<i>Item1</i>	<i>Item2</i>	<i>Item3</i>	<i>Item4</i>	<i>Item5</i>
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.4	1.5	0.2	setosa

Анализ данных



Дополнительно

- <https://www.oreilly.com/library/view/analyzing-and-visualizing/9781492048350> <- короткая книга
- <https://www.youtube.com/watch?v=rmSgNJLLfSM> <- F# for DATA ANALYSIS, 2017 год
- <https://jvaneyck.wordpress.com/2017/06/19/exploring-data-and-apis-with-f-type-providers/> <- про поставщики типов