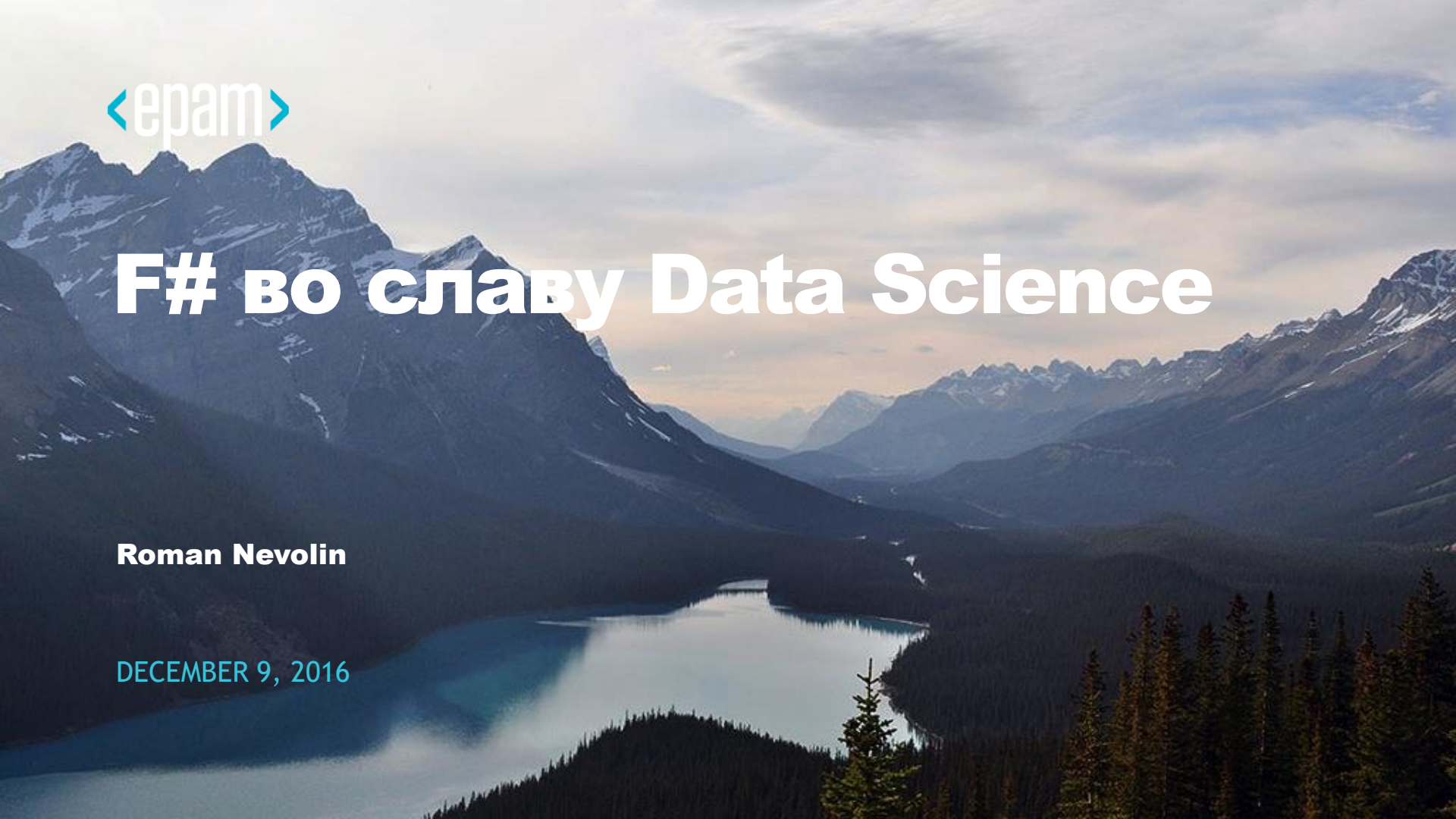


F# во славу Data Science

Roman Nevolin

DECEMBER 9, 2016

Jeremy Jarvis
@jeremyjarvis

Читать

"A data scientist is a statistician who lives in San Fransisco" #monkigras

РЕТВИТОВ	ОТМЕТКИ «НРАВИТСЯ»
1 498	892

(((Josh Wills)))
@josh_wills

Читать

Data Scientist (n.): Person who is better at statistics than any software engineer and better at software engineering than any statistician.

Показать перевод

РЕТВИТОВ 1 497

ОТМЕТКА «НРАВИТСЯ» 1 061

Data Scientist – Person who is exploring data

Programmer – Person who is … what?

Programmer – Person who is processing data

А что нужно для изучения данных?

- Интерактивные вычисления

А что нужно для изучения данных?

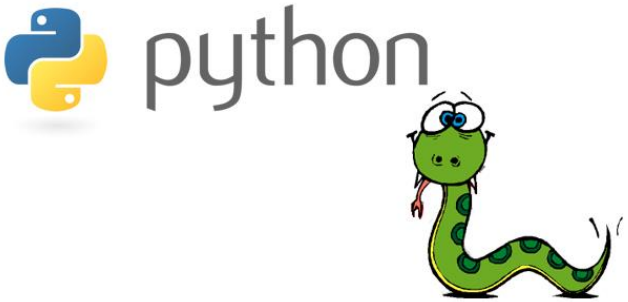
- Интерактивные вычисления
- Понятные визуализации

А что нужно для изучения данных?

- Интерактивные вычисления
- Понятные визуализации
- Хорошие инструменты для их анализа

А что нужно для изучения данных?

- Интерактивные вычисления
- Понятные визуализации
- Хорошие инструменты для их анализа

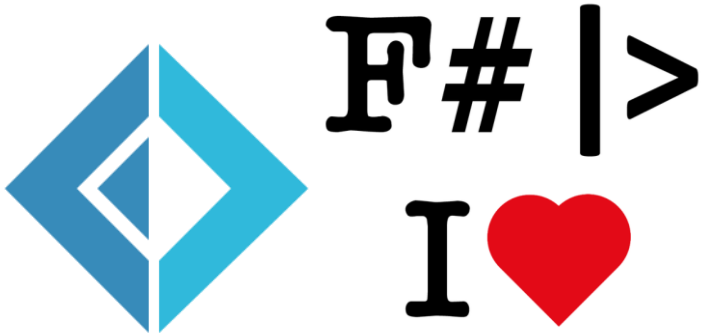


А что нужно для изучения данных?

- Интерактивные вычисления
- Понятные визуализации
- Хорошие инструменты для их анализа

А что нужно для изучения данных?

- Интерактивные вычисления
- Понятные визуализации
- Хорошие инструменты для их анализа



К слову о данных


```
{
  "items": [
    {
      "owner": {
        "reputation": 542,
        "user_id": 5662524,
        "user_type": "registered",
        "accept_rate": 64,
        "profile_image": "https://i.stack.imgur.com/kdOkn.jpg?s=128&g=1",
        "display_name": "RAUSHAN KUMAR",
        "link": "http://stackoverflow.com/users/5662524/raushan-kumar"
      },
      "is_accepted": false,
      "score": 0,
      "last_activity_date": 1477569620,
      "creation_date": 1477569620,
      "answer_id": 40284077,
      "question_id": 40284012
    }
  ],
  "has_more": true,
  "backoff": 10,
  "quota_max": 300,
  "quota_remaining": 297
}
```

К слову о данных

```
instant,dteday,season,yr,mnth,holiday,weekday,workingday,weathersit,temp,atemp,hum
1,2011-01-01,1,0,1,0,6,0,2,0.344167,0.363625,0.805833,0.160446,331,654,985
2,2011-01-02,1,0,1,0,0,0,2,0.363478,0.353739,0.696087,0.248539,131,670,801
3,2011-01-03,1,0,1,0,1,1,1,0.196364,0.189405,0.437273,0.248309,120,1229,1349
4,2011-01-04,1,0,1,0,2,1,1,0.2,0.212122,0.590435,0.160296,108,1454,1562
5,2011-01-05,1,0,1,0,3,1,1,0.226957,0.22927,0.436957,0.1869,82,1518,1600
6,2011-01-06,1,0,1,0,4,1,1,0.204348,0.233209,0.518261,0.0895652,88,1518,1606
7,2011-01-07,1,0,1,0,5,1,2,0.196522,0.208839,0.498696,0.168726,148,1362,1510
8,2011-01-08,1,0,1,0,6,0,2,0.165,0.162254,0.535833,0.266804,68,891,959
9,2011-01-09,1,0,1,0,0,0,1,0.138333,0.116175,0.434167,0.36195,54,768,822
10,2011-01-10,1,0,1,0,1,1,1,0.150833,0.150888,0.482917,0.223267,41,1280,1321
11,2011-01-11,1,0,1,0,2,1,2,0.169091,0.191464,0.686364,0.122132,43,1220,1263
12,2011-01-12,1,0,1,0,3,1,1,0.172727,0.160473,0.599545,0.304627,25,1137,1162
13,2011-01-13,1,0,1,0,4,1,1,0.165,0.150883,0.470417,0.301,38,1368,1406
14,2011-01-14,1,0,1,0,5,1,1,0.16087,0.188413,0.537826,0.126548,54,1367,1421
15,2011-01-15,1,0,1,0,6,0,2,0.233333,0.248112,0.49875,0.157963,222,1026,1248
16,2011-01-16,1,0,1,0,0,0,1,0.231667,0.234217,0.48375,0.188433,251,953,1204
17,2011-01-17,1,0,1,1,1,0,2,0.175833,0.176771,0.5375,0.194017,117,883,1000
18,2011-01-18,1,0,1,0,2,1,2,0.216667,0.232333,0.861667,0.146775,9,674,683
19,2011-01-19,1,0,1,0,3,1,2,0.292174,0.298422,0.741739,0.208317,78,1572,1650
20,2011-01-20,1,0,1,0,4,1,2,0.261667,0.25505,0.538333,0.195904,83,1844,1927
21,2011-01-21,1,0,1,0,5,1,1,0.1775,0.157833,0.457083,0.353242,75,1468,1543
22,2011-01-22,1,0,1,0,6,0,1,0.0591304,0.0790696,0.4,0.17197,93,888,981
23,2011-01-23,1,0,1,0,0,0,1,0.0965217,0.0988391,0.436522,0.2466,150,836,986
24,2011-01-24,1,0,1,0,1,1,1,0.0973913,0.11793,0.491739,0.15833,86,1330,1416
25,2011-01-25,1,0,1,0,2,1,2,0.223478,0.234526,0.616957,0.129796,186,1799,1985
```

К слову о данных

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
1	1	2011-01-01	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.160446	331	654	985
2	2	2011-01-02	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.248539	131	670	801
3	3	2011-01-03	1	0	1	0	1	1	1	0.196364	0.189405	0.437273	0.248309	120	1229	1349
4	4	2011-01-04	1	0	1	0	2	1	1	0.2	0.212122	0.590435	0.160296	108	1454	1562
5	5	2011-01-05	1	0	1	0	3	1	1	0.226957	0.22927	0.436957	0.1869	82	1518	1600
6	6	2011-01-06	1	0	1	0	4	1	1	0.204348	0.233209	0.518261	0.0895652	88	1518	1606
7	7	2011-01-07	1	0	1	0	5	1	2	0.196522	0.208839	0.498696	0.168726	148	1362	1510
8	8	2011-01-08	1	0	1	0	6	0	2	0.165	0.162254	0.535833	0.266804	68	891	959
9	9	2011-01-09	1	0	1	0	0	0	1	0.138333	0.116175	0.434167	0.36195	54	768	822
10	10	2011-01-10	1	0	1	0	1	1	1	0.150833	0.150888	0.482917	0.223267	41	1280	1321
11	11	2011-01-11	1	0	1	0	2	1	2	0.169091	0.191464	0.686364	0.122132	43	1220	1263
12	12	2011-01-12	1	0	1	0	3	1	1	0.172727	0.160473	0.599545	0.304627	25	1137	1162
13	13	2011-01-13	1	0	1	0	4	1	1	0.165	0.150883	0.470417	0.301	38	1368	1406
14	14	2011-01-14	1	0	1	0	5	1	1	0.16087	0.188413	0.537826	0.126548	54	1367	1421
15	15	2011-01-15	1	0	1	0	6	0	2	0.233333	0.248112	0.49875	0.157963	222	1026	1248
16	16	2011-01-16	1	0	1	0	0	0	1	0.231667	0.234217	0.48375	0.188433	251	953	1204
17	17	2011-01-17	1	0	1	1	1	0	2	0.175833	0.176771	0.5375	0.194017	117	883	1000
18	18	2011-01-18	1	0	1	0	2	1	2	0.216667	0.232333	0.861667	0.146775	9	674	683
19	19	2011-01-19	1	0	1	0	3	1	2	0.292174	0.298422	0.741739	0.208317	78	1572	1650
20	20	2011-01-20	1	0	1	0	4	1	2	0.261667	0.25505	0.538333	0.195904	83	1844	1927
21	21	2011-01-21	1	0	1	0	5	1	1	0.1775	0.157833	0.457083	0.353242	75	1468	1543
22	22	2011-01-22	1	0	1	0	6	0	1	0.0591304	0.0790696	0.4	0.17197	93	888	981

К слову о данных

```
<wb:data>
  <wb:indicator id="NY.GDP.PCAP.CD">GDP per capita (current US$)</wb:indicator>
  <wb:country id="1A">Arab World</wb:country>
  <wb:date>2010</wb:date>
  <wb:value>5957.94958642587</wb:value>
  <wb:decimal>1</wb:decimal>
</wb:data>
<wb:data>
  <wb:indicator id="NY.GDP.PCAP.CD">GDP per capita (current US$)</wb:indicator>
  <wb:country id="S3">Caribbean small states</wb:country>
  <wb:date>2010</wb:date>
  <wb:value>8699.51769881953</wb:value>
  <wb:decimal>1</wb:decimal>
</wb:data>
```

К слову о данных

```
<div class="wrapper">
    <header>
        <div class="uui-header  orange">
            <nav role="navigation">
                <div class="events-responsive-navigation">
                    <div class="sidebar-toggle-box gray dropdown-toggle" data-toggle="dropdown">
                        <span class="fa fa-reorder"></span>
                    </div>
                    <ul class="uui-promo-menu dropdown-menu blue" role="menu">
            <li>
                <a href="/events/itsubibotnik-samara-2016/agenda">
                    <span>Agenda</span>
                </a>
            </li>
            <li>
                <a href="/events/itsubibotnik-samara-2016/speakers">
                    <span>Speakers</span>
                </a>
            </li>
        <li class="">
            <a href="/events/itsubibotnik-samara-2016/my-agenda">
              <span>My Agenda</span>
            </a>
        </li>

</ul>
```

К слову о данных

DOTNEXT

Программа Регистрация F.A.Q. Спонсоры Контакты EN

F# во славу Data Science

Зал 2 | 11:25

Что нужно Data Scientist'у для полного счастья? Конечно, данные, и чем больше — тем лучше. А ещё удобный инструментарий для их обработки и анализа, возможность интерактивного взаимодействия и много разнообразных визуализаций. На платформе .NET такой инструмент — это F#.

В этом докладе будут рассмотрены возможности языка, делающие его таким привлекательным для Data Science, самые полезные библиотеки для этой цели. Также будут разобраны некоторые реальные кейсы применения.

Базовые знания F# будут полезны, но не обязательны.

Роман Неволин nevoroman
EPAM

Разработчик компании EPAM, специализирующийся на работе с данными в наукоемких проектах. Активно занимается исследованиями в области Machine Learning и разработкой собственных инструментов машинного обучения.

К слову о данных

К слову о данных

```
type Data = CsvProvider<dataPath>
let dataset = Data.GetSample()
let data = dataset.Rows |> Seq.toArray

let shuffle (r : Random) xs = xs |> Array.sortBy (fun _ -> r.Next())
let randomedData = data |> shuffle(Random(10))
let training,validation =  randomedData.[..600],randomedData.[600..]


let count =  data |> Seq.map (fun x -> float x.Cnt)
Chart.Line count
```

К слову о данных


```
{
  "items": [
    {
      "owner": {
        "reputation": 542,
        "user_id": 5662524,
        "user_type": "registered",
        "accept_rate": 64,
        "profile_image": "https://i.stack.imgur.com/kdOkn.jpg?s=128&g=1",
        "display_name": "RAUSHAN KUMAR",
        "link": "http://stackoverflow.com/users/5662524/raushan-kumar"
      },
      "is_accepted": false,
      "score": 0,
      "last_activity_date": 1477569620,
      "creation_date": 1477569620,
      "answer_id": 40284077,
      "question_id": 40284012
    }
  ],
  "has_more": true,
  "backoff": 10,
  "quota_max": 300,
  "quota_remaining": 297
}
```

Парсинг

```
using( var httpClient = new HttpClient() ) {
    var json = httpClient
            .GetStringAsync( "https://api.stackexchange.com/2.2/questions/123/answers?site=stackoverflow" )
            .Result;
    var result = JsonConvert.DeserializeObject<Result<Answer>>( json );

    // Make some stuff
}
```

Парсинг

```
using( var httpClient = new HttpClient() ) {
    var json = httpClient
            .GetStringAsync( "https://api.stackexchange.com/2.2/questions/123/answers?site=stackoverflow" )
            .Result;
    var result = JsonConvert.DeserializeObject<Result<Answer>>( json );

    // Make some stuff
}


class Result<T> {
    public T[] Items { get; set; }
    public bool HasMore { get; set; }
    public int QuotaMax { get; set; }
    public int QuotaRemaining { get; set; }
}
```

Парсинг

```
using( var httpClient = new HttpClient() ) {
    var json = httpClient
            .GetStringAsync( "https://api.stackexchange.com/2.2/questions/123/answers?site=stackoverflow" )
            .Result;
    var result = JsonConvert.DeserializeObject<Result<Answer>>( json );

    // Make some stuff
}
```

```
class Result<T> {
    public T[] Items { get; set; }
    public bool HasMore { get; set; }
    public int QuotaMax { get; set; }
    public int QuotaRemaining { get; set; }
}
```

```
class Answer {
    public User Owner { get; set; }
    public bool IsAccepted { get; set; }
    public int Score { get; set; }
    public long LastActivityDate { get; set; }
    public long LastEditDate { get; set; }
    public long CreationDate { get; set; }
    public int AnswerId { get; set; }
    public int QuestionId { get; set; }
}
```

Парсинг

```
using( var httpClient = new HttpClient() ) {
    var json = httpClient
            .GetStringAsync( "https://api.stackexchange.com/2.2/questions/123/answers?site=stackoverflow" )
            .Result;
    var result = JsonConvert.DeserializeObject<Result<Answer>>( json );

    // Make some stuff
}
```

```
class Result<T> {
    public T[] Items { get; set; }
    public bool HasMore { get; set; }
    public int QuotaMax { get; set; }
    public int QuotaRemaining { get; set; }
}
```

```
class User {
    public int Reputation { get; set; }
    public int UserId { get; set; }
    public string UserType { get; set; }
    public string ProfileImage { get; set; }
    public string DisplayName { get; set; }
    public string Link { get; set; }
}
```

```
class Answer {
    public User Owner { get; set; }
    public bool IsAccepted { get; set; }
    public int Score { get; set; }
    public long LastActivityDate { get; set; }
    public long LastEditDate { get; set; }
    public long CreationDate { get; set; }
    public int AnswerId { get; set; }
    public int QuestionId { get; set; }
}
```

А что, если...

- ...поменяется API?

А что, если...

- ...поменяется API?
- ...мы допустили мелкую опечатку?

А что, если...

- ...поменяется API?
- ...мы допустили мелкую опечатку?
- ...таких типов – несколько сотен?

Meet
Type Providers

```
type QuestionsResult = JsonProvider<"""https://api.stackexchange.com/2.2/questions?site=stackoverflow""">

let getQuestionsByTag tag =
    QuestionsResult.Load (
        "https://api.stackexchange.com/2.2/search?order=desc&sort=activity&site=stackoverflow&tagged=" + tag)
```

```
type QuestionsResult = XmlProvider<"""https://api.stackexchange.com/2.2/questions?site=stackoverflow""">

let getQuestionsByTag tag =
    QuestionsResult.Load (
        "https://api.stackexchange.com/2.2/search?order=desc&sort=activity&site=stackoverflow&tagged=" + tag)
```

```
type QuestionsResult = JsonProvider<"""https://api.stackexchange.com/2.2/questions?site=stackoverflow""">

let getQuestionsByTag tag =
    QuestionsResult.Load (
        "https://api.stackexchange.com/2.2/search?order=desc&sort=activity&site=stackoverflow&tagged=" + tag)
```

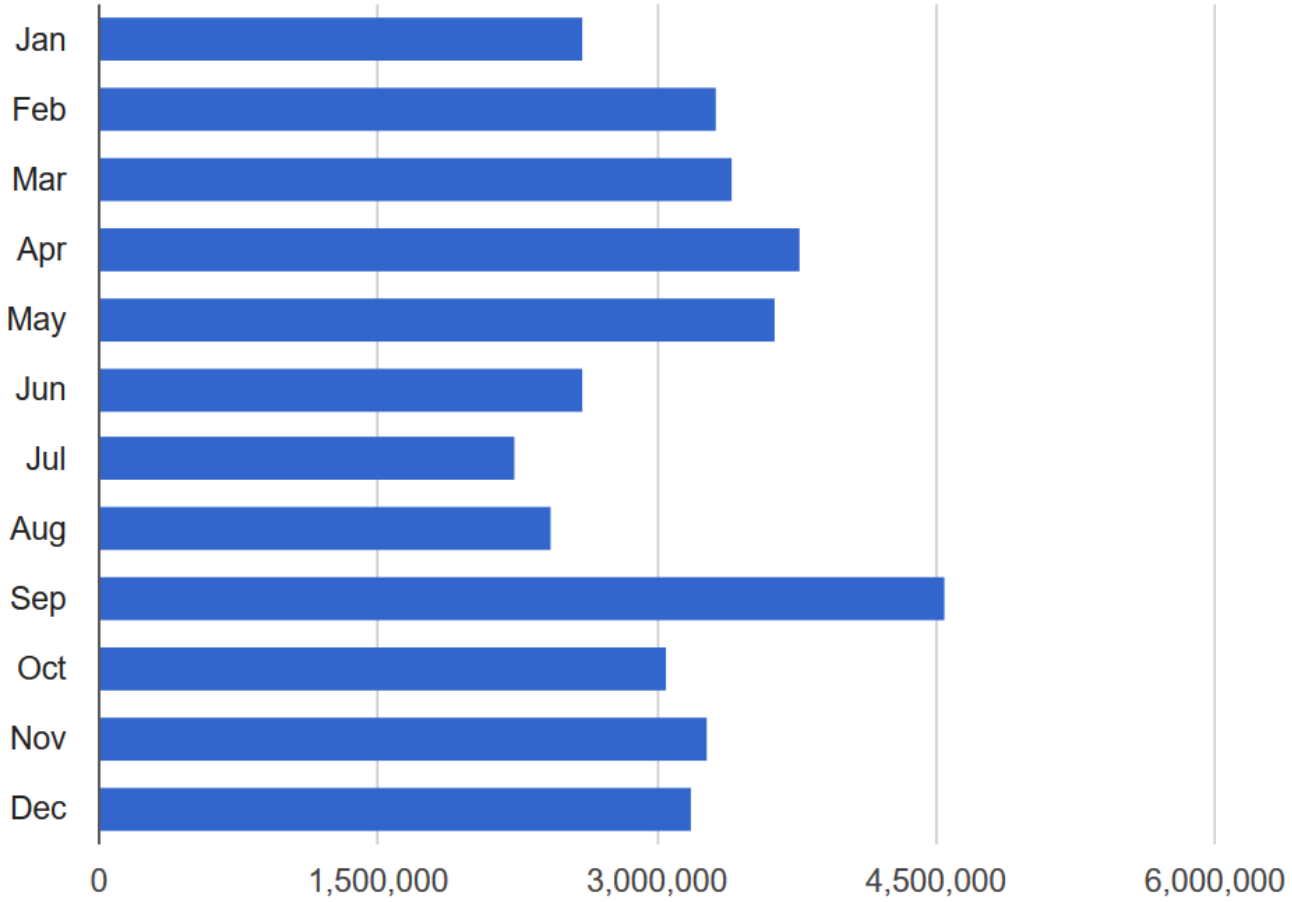
```
tag).Items.[0].
```

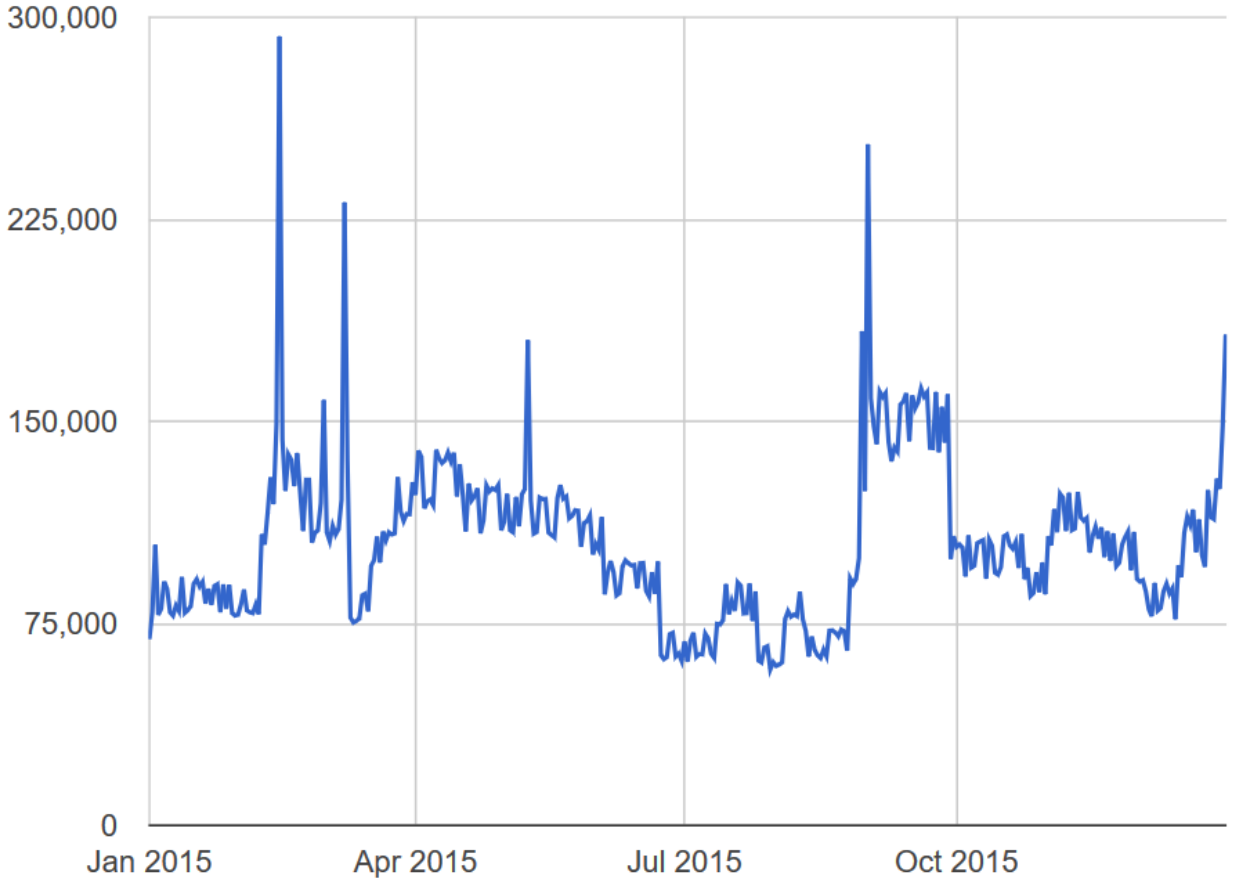
- AcceptedAnswerId
- AnswerCount
- ClosedDate
- ClosedReason
- CreationDate
- IsAnswered
- JsonValue
- LastActivityDate
- LastEditDate

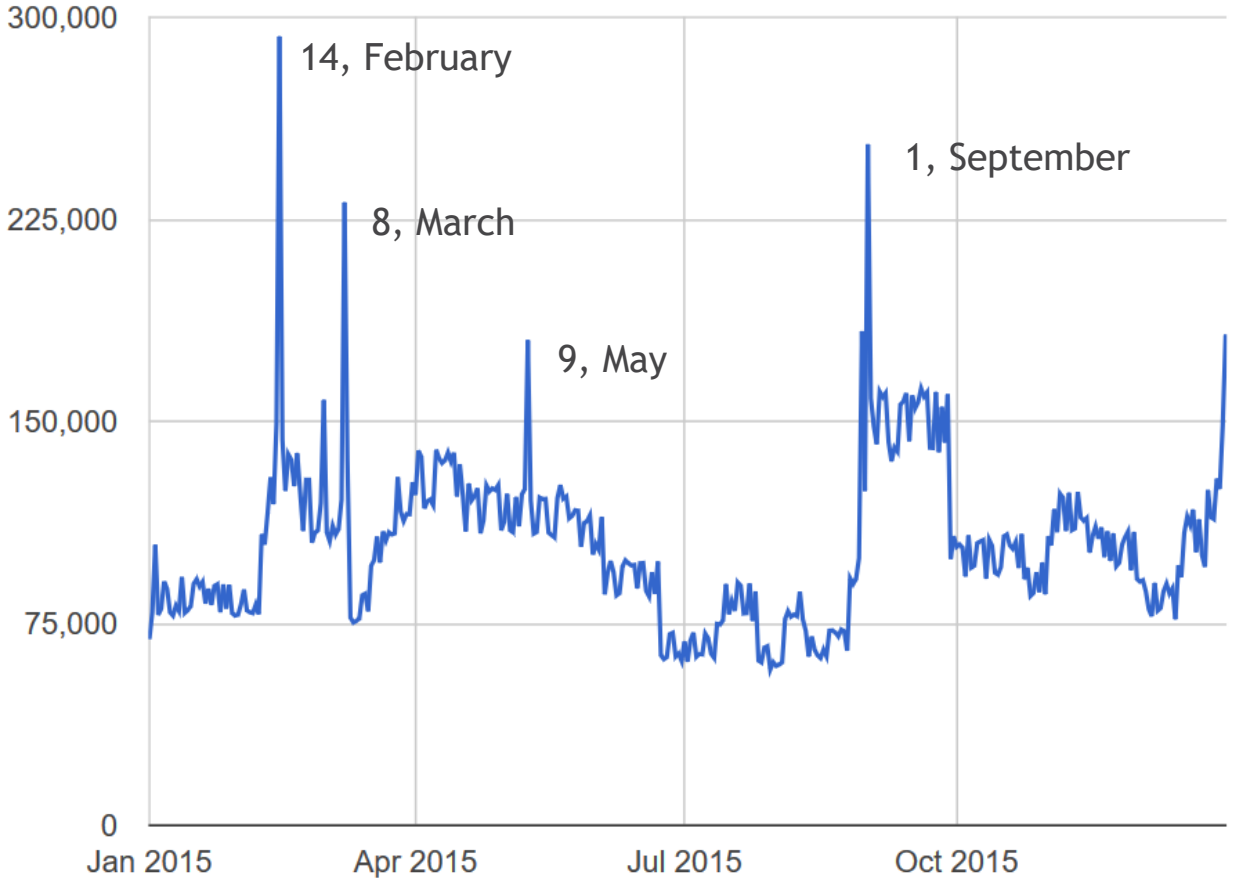
```
April,April,"1/6/2014 12:00:00 AM",50000,0,0
April,April,"1/13/2014 12:00:00 AM",40000,0,0
April,April,"1/20/2014 12:00:00 AM",220000,0,0
April,April,"1/27/2014 12:00:00 AM",120000,0,0
April,April,"2/3/2014 12:00:00 AM",80000,0,0
April,April,"2/10/2014 12:00:00 AM",90000,0,0
April,April,"2/17/2014 12:00:00 AM",100000,0,0
April,April,"2/24/2014 12:00:00 AM",100000,0,0
April,April,"3/3/2014 12:00:00 AM",80000,0,0
April,April,"3/10/2014 12:00:00 AM",110000,0,0
April,April,"3/17/2014 12:00:00 AM",110000,0,0
April,April,"3/24/2014 12:00:00 AM",70000,0,0
April,April,"3/31/2014 12:00:00 AM",110000,0,0
April,April,"4/7/2014 12:00:00 AM",90000,0,0
April,April,"4/14/2014 12:00:00 AM",110000,0,0
```

Точно! Провайдеры типов!

```
val it : CsvProvider<...>.Row [] =
  [|("April", "April", 1/6/2014 12:00:00 AM, 50000, 0, 0);
    ("April", "April", 1/13/2014 12:00:00 AM, 40000, 0, 0);
    ("April", "April", 1/20/2014 12:00:00 AM, 220000, 0, 0);
    ("April", "April", 1/27/2014 12:00:00 AM, 120000, 0, 0);
    ("April", "April", 2/3/2014 12:00:00 AM, 80000, 0, 0);
    ("April", "April", 2/10/2014 12:00:00 AM, 90000, 0, 0);
```


Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
0
1,500,000
3,000,000
4,500,000
6,000,000


300,000
225,000
150,000
75,000
0
Jan 2015
Apr 2015
Jul 2015
Oct 2015


300,000
225,000
150,000
75,000
0
Jan 2015
Apr 2015
Jul 2015
Oct 2015
14, February
8, March
9, May
1, September

Meet
FSharp Charting

fslab.org/FSharp.Charting

```
let salesData = SalesProvider.Sales
salesData |> Chart.Line
```

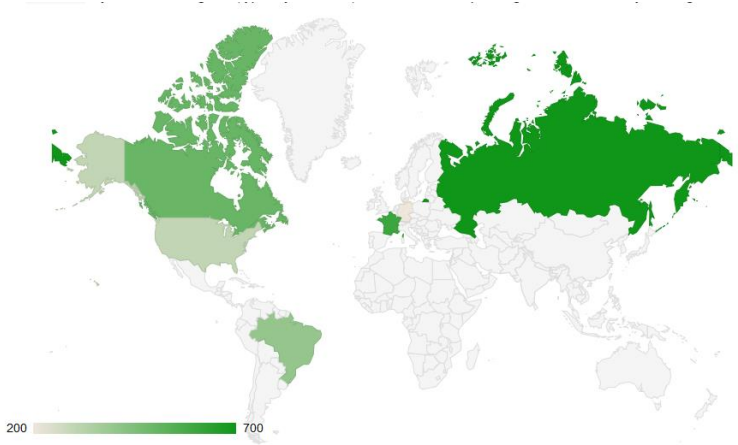
Shay Friedman
@ironshay

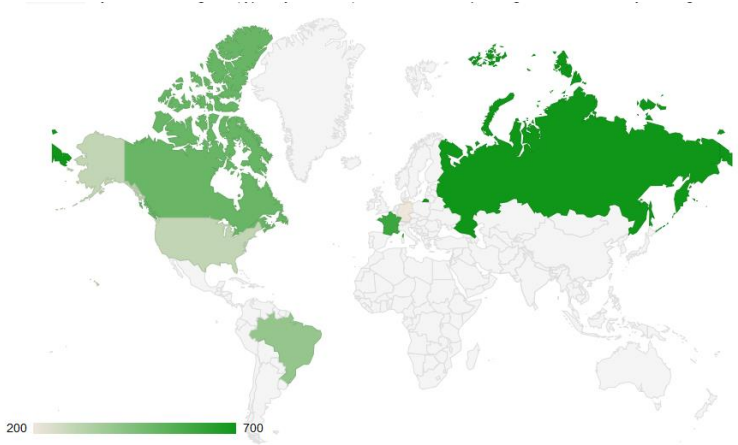
Читать

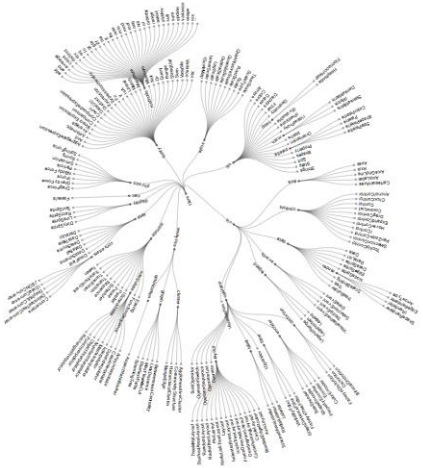
Drinking game for web devs:
(1) Think of a noun
(2) Google "<noun>.js"
(3) If a library with that name exists - drink

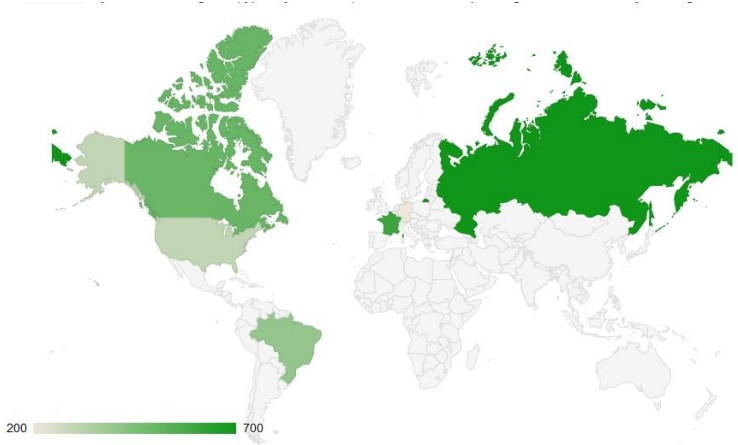
Показать перевод

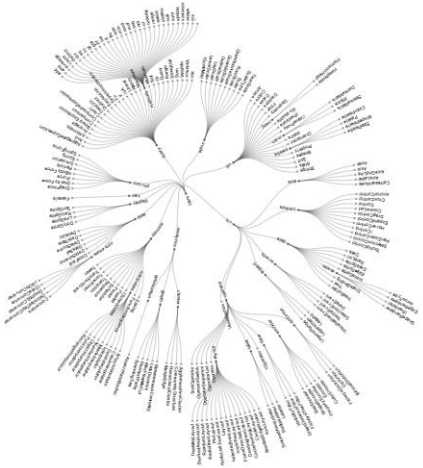
РЕТВИТОВ	ОТМЕТОК «НРАВИТСЯ»
2 287	809

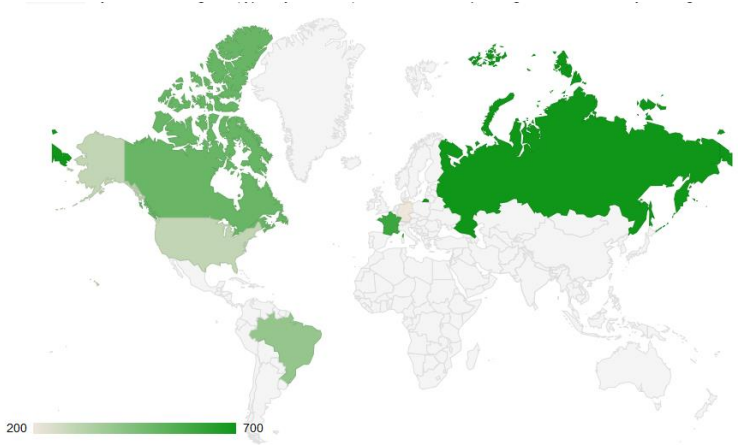

200
700

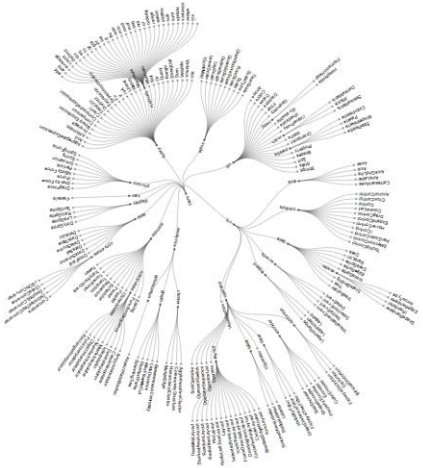

200
700




200
700




200
700



Ну что, уходим писать на JS?


I drink to forget _____.
JavaScript

Meet the Fable (F# |> Babel)

fable.io

D3 world tour

Looping through countries of the world

This demo is a Fable port of Mike Bostock's World Tour D3 demo. It uses the D3 library to create a visualization that loops through all countries of the world and shows them on the globe one by one. You can find the full source code on GitHub.

On the technical side, the demo shows some of the more interesting aspects of calling JavaScript libraries from Fable. You'll learn how to define mappings for imported scripts, how to pass lambdas to JS code and the `?` operator.

```
let projection =
  D3.Geo.Globals.orthographic()
    .translate((width / 2., height / 2.))
    .scale(width / 2. - 20.)
    .clipAngle(90.)
    .precision(0.6)
```

```
let countries =
  topojson?feature(world, world?objects?countries)?features
  |> unbox<obj[]>
  |> Array.filter (fun d ->
      names |> Seq.exists (fun n ->
        if (string d?id) = (string n?id)
        then d?name <- n?name; true
        else false))
  |> Array.sortWith (fun a b ->
        compare (string a?name) (string b?name))
```

crimesandpatterns.net

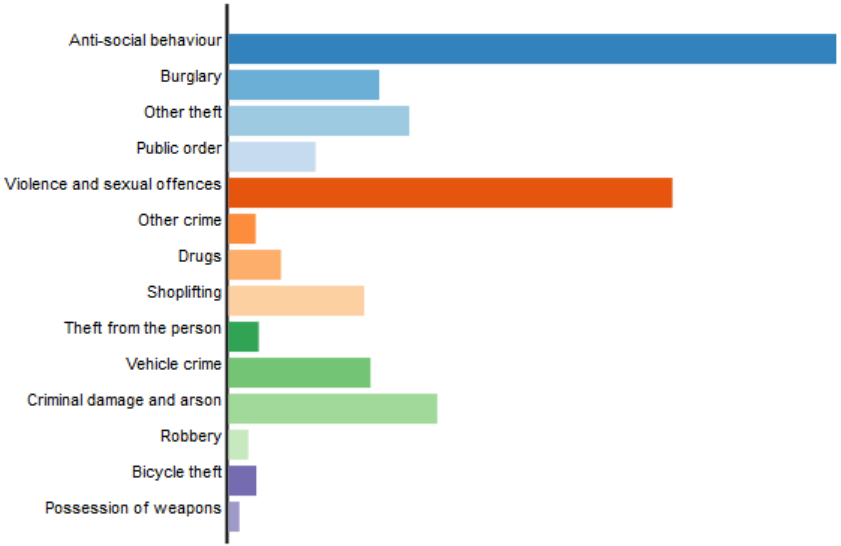
Crimes & Patterns

...statistics are only a starting point. Each case is unique and has to be figured out on its own specific terms.

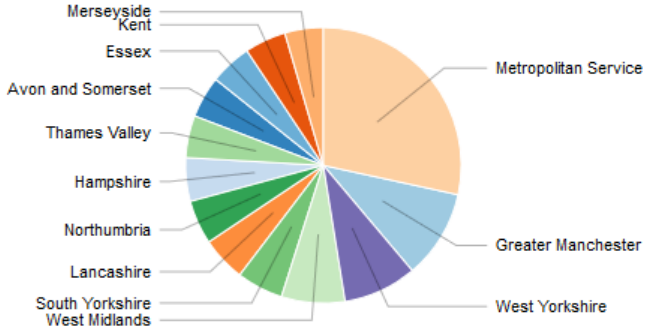
John Douglas, Mark Olshaker in *Law & Disorder* —

Statistics | Predictions | Similarities | About

Crimes by type



Crimes by place



crimesandpatterns.net

Crimes & Patterns

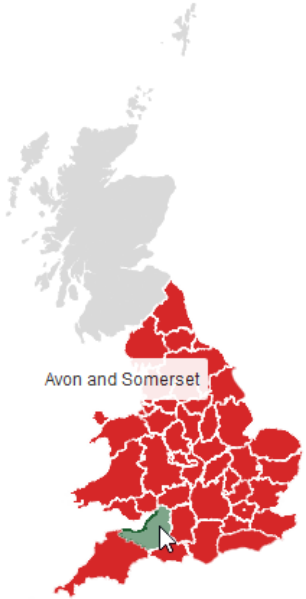
...statistics are only a starting point. Each case is unique and has to be figured out on its own specific terms.

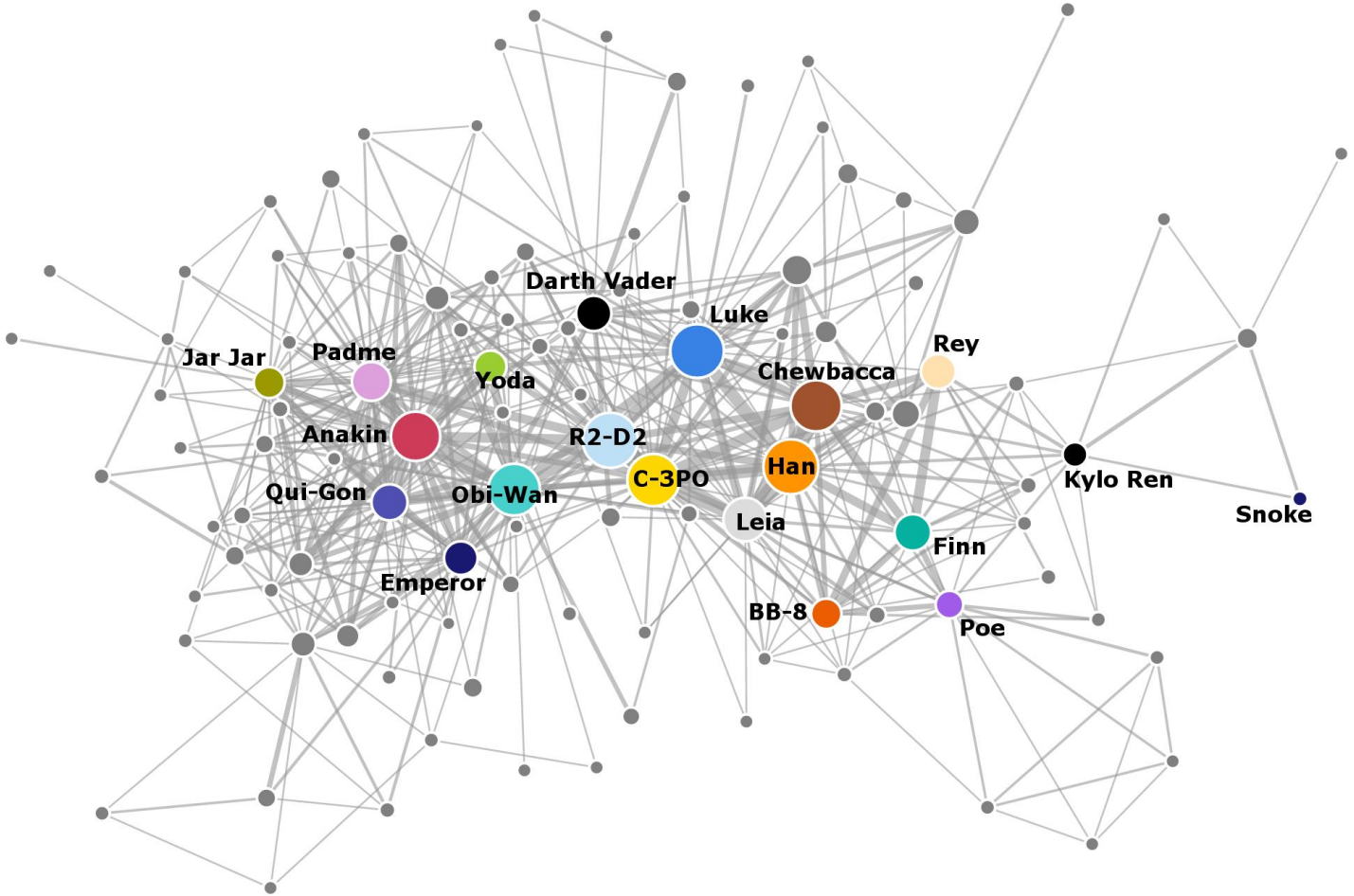
John Douglas, Mark Olshaker in *Law & Disorder* —

Statistics | **Predictions** | Similarities | About

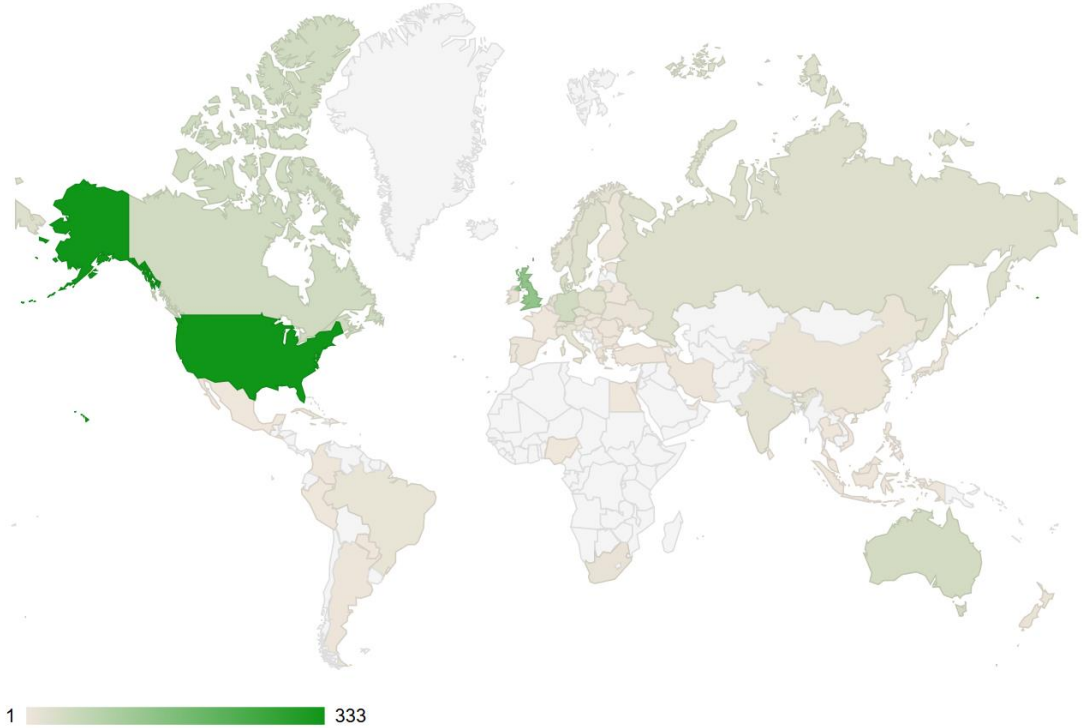
Will suspect (not) be found?

Bicycle theft




Darth Vader
Luke
Rey
Jar Jar
Padme
Yoda
Chewbacca
Anakin
R2-D2
Han
C-3PO
Kylo Ren
Qui-Gon
Obi-Wan
Leia
Snoke
Finn
Emperor
BB-8
Poe

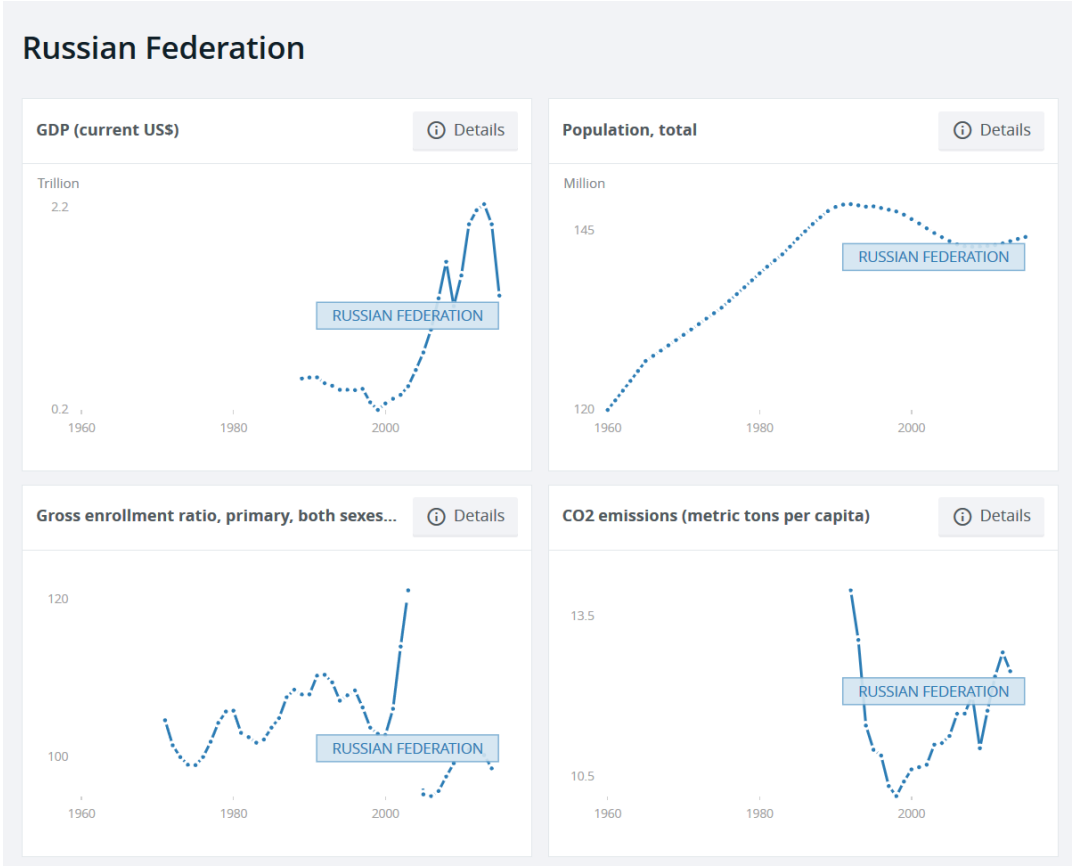
Где живут F# разработчики?



Meet Deedle

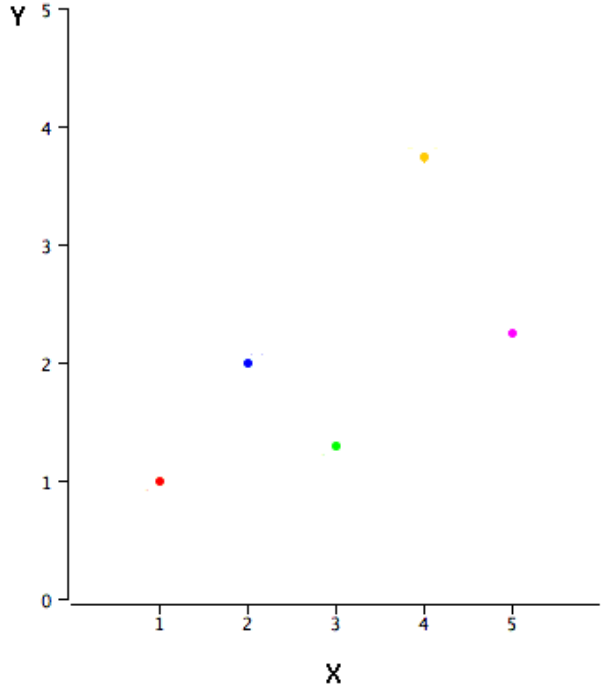
bluemountaincapital.github.io/Deedle

databank.worldbank.org

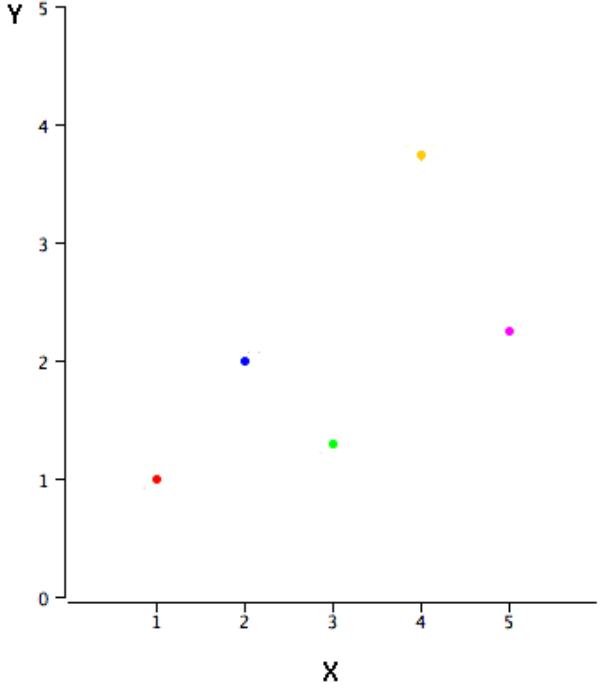


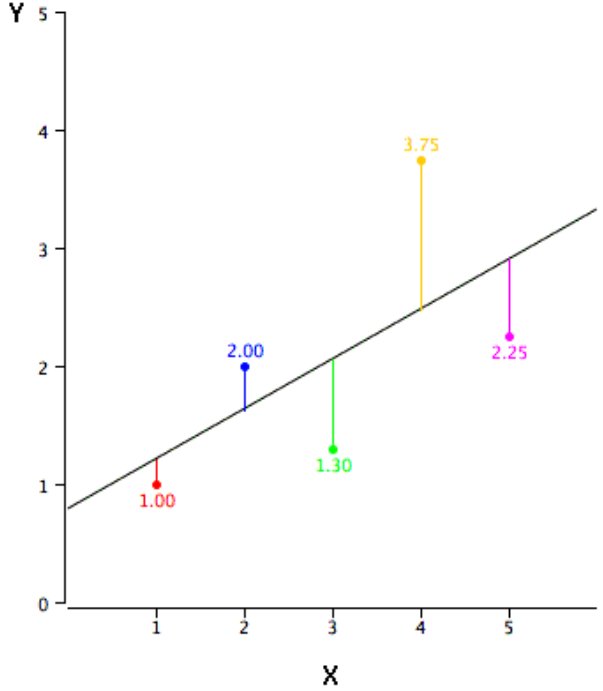
Линейная регрессия – метод восстановления зависимости между данными

Линейная регрессия – метод восстановления зависимости между данными



Линейная регрессия – метод восстановления зависимости между данными

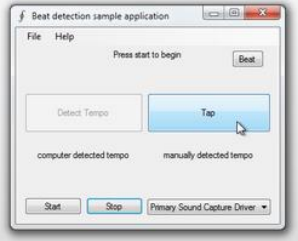




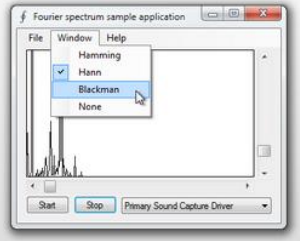
Meet
Accord Framework

accord-framework.net

Audio beat detector

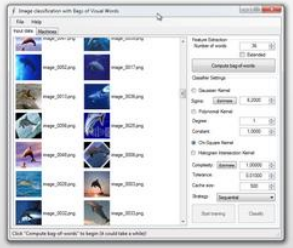


Spectrum analyzer (Fourier)



Wave Recorder

Classification (Bag-of-Words and SVM)



Corners detection (FAST)

Corners detection (Harris)

Corners detection (SURF)

Image stitching (FREAK)

Image stitching (Harris)

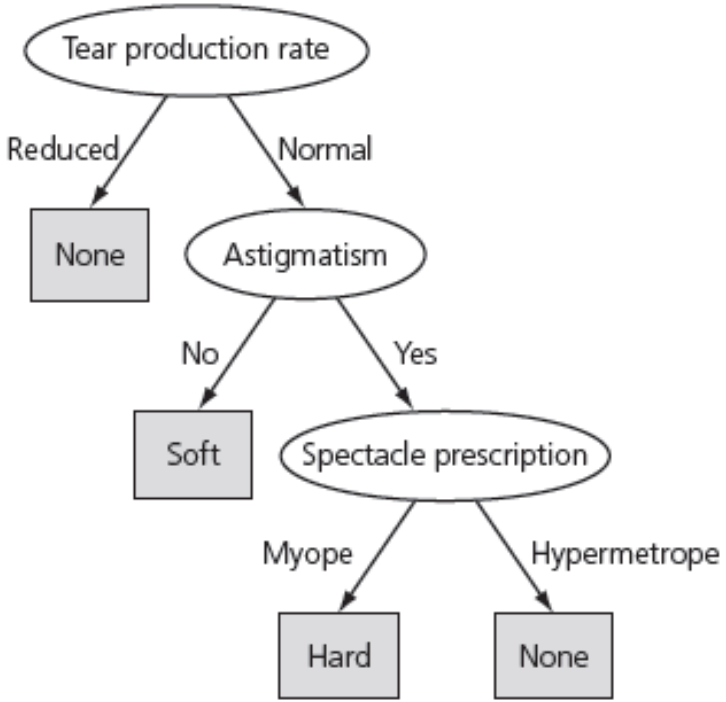
```
public LenseType ChooseLenseType(EyeData eye) {
    if ( eye.Tears == Tears.Reduced ) {
        return LenseType.None;
    }
    else {
        return LenseType.Soft;
    }
}
```

```
public LenseType ChooseLenseType( EyeData eye ) {
    if( eye.Tears == Tears.Reduced ) {
        return LenseType.None;
    }
    else if ( eye.IsAstigmatic ) {
        return LenseType.Soft;
    }
    else {
        return LenseType.Hard;
    }
}
```

```
public LenseType ChooseLenseType( EyeData eye ) {
    if( eye.Tears == Tears.Reduced ) {
        return LenseType.None;
    }
    else if( eye.IsAstigmatic ) {
        if ( eye.Age != Age.Presbyopic ) {
            return LenseType.Hard;
        }
        else {
            return LenseType.None;
        }
    }
    else {
        return LenseType.Hard;
    }
}
```

```
public LenseType ChooseLenseType( EyeData eye ) {
    if( eye.Tears == Tears.Reduced ) {
        return LenseType.None;
    }
    else if( eye.IsAstigmatic ) {
        if( eye.Age != Age.Presbyopic ) {
            return LenseType.Hard;
        }
        else {
            if ( eye.Prescription == Prescription.Myope ) {
                return LenseType.None;
            }
            else {
                return LenseType.Soft;
            }
        }
    }
    else {
        if ( eye.Prescription == Prescription.Myope ) {
            return LenseType.Hard;
        }
        else {
            if ( eye.Age != Age.Young ) {
                return LenseType.None;
            }
            else {
                return LenseType.Hard;
            }
        }
    }
}
```

Decision Tree



- А в этом вашем F# столько же крутых статистических пакетов, сколько и в R?

- А в этом вашем F# столько же крутых статистических пакетов, сколько и в R?

Нет. Но мы не стесняемся их тырить, и делаем это умело

Проблемы межъязыкового взаимодействия

- Это же надо писать код на другом языке!

Проблемы межъязыкового взаимодействия

- Это же надо писать код на другом языке!
- Блиииин, сериализация!

Проблемы межъязыкового взаимодействия

- Это же надо писать код на другом языке!
- Блиииин, сериализация!
- И взаимодействие между процессами…

Проблемы межъязыкового взаимодействия

- Это же надо писать код на другом языке!
- Блиииин, сериализация!
- И взаимодействие между процессами…

Да ну его нафиг!

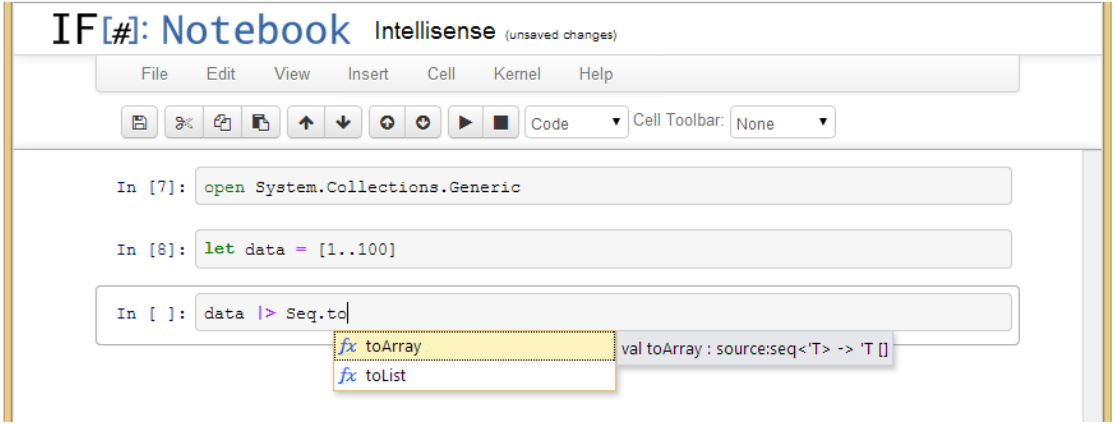
Meet
R Provider

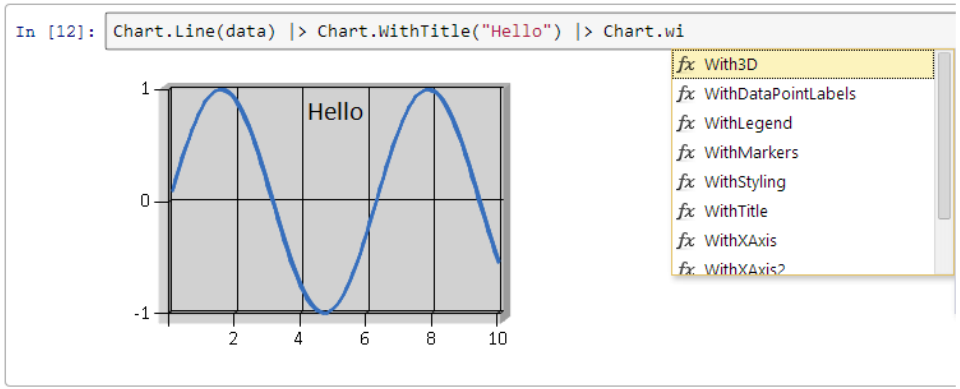
bluemountaincapital.github.io/FSharpRProvider

```
let widgets = [ 3; 8; 12; 15; 19; 18; 18; 20; ]
let sprockets = [ 5; 4; 6; 7; 12; 9; 5; 6; ]

R.plot(widgets)

R.plot(widgets, sprockets)
```


jupyter

github.com/fsprojects/IfSharp





Data science tools Professional, powerful & cross-platform

FsLab is a collection of libraries for data-science. It provides a rapid development environment that lets you write advanced analysis with a few lines of production-quality code.

Access data using **type providers**, explore it using a efficient **data-frame and time-series** library and through the **R language** integration and use rich **visualization and reporting**.

Download Getting started

Cross-platform

FsLab runs on Mac, Linux and Windows. You can use it with Emacs, Xamarin Studio, MonoDevelop, Visual Studio or other F#-enabled editor.

Production-ready

Once you're done with the initial data analysis, you can turn your code into production-quality implementation without rewriting it.

Powered by F#

FsLab lets you use a simple but powerful language that analysts, data scientists and software developers all understand.

Open-source

All libraries included in FsLab are licensed under OSI-approved licenses and have an active community of contributors on GitHub.

Interactive

FsLab encourages an interactive development style. This makes it easy to explore the data and experiment with different analyses.

Cloud-scale

F# code is type-inferred and efficiently compiled. It runs on multi-core and you can easily scale to the cloud with MBrace.

fslab.org

Libraries What is included in FsLab?

F# Data First-class data access

- Use language that understands external data sources
- Access JSON, XML, CSV, HTML and WorldBank
- Get auto-completion based on a sample document

F# Data homepage

Deedle Interactive data exploration

- Data frame and series library inspired by pandas
- Quickly explore data & build efficient compiled code
- Automatic alignment, missing data handling & more

Deedle in 10 minutes | Homepage

R Provider Professional stats packages

- Access over 6000 professional R packages
- Pass data frames and time-series to R easily
- Get auto-completion on R functions & parameters

R provider homepage

Math.NET Fast numerical calculations

- Linear algebra, statistics and probability
- Optimization, transformations, curve fitting
- Switch to parallel or native MKL implementation

Math.NET Numerics

XPlot Powerful HTML5 charting

- Create HTML5 charts with easy-to-use library
- Choose between Google Charts and Plot.ly
- Lines, bars, pies, bubbles, maps and much more!

XPlot homepage

F# Charting For desktop charting

- Rich, efficient charts for desktop apps
- Create animated live and incremental charts
- Easily integrates with .NET applications

F# Charting homepage

FsLab For integrated experience

- One easy to install package with all you need
- Data access, analysis and visualization
- Everything works well together, no versioning hell

FsLab templates | Visual Studio template

Literate F# Produce LaTeX and HTML reports

- Mix F# code and Markdown annotations in one file
- Embedded equations, tables, charts, R graphics and more
- Produce high-quality LaTeX, PDF or HTML outputs

Introduction | Embedding results

```
let inputs = [| 1..100 |]

let streamComputationTask =
    inputs
    |> CloudFlow.OfArray
    |> CloudFlow.map (fun num -> num * num)
    |> CloudFlow.map (fun num -> num % 10)
    |> CloudFlow.countBy id
    |> CloudFlow.toArray
    |> cluster.CreateProcess
```

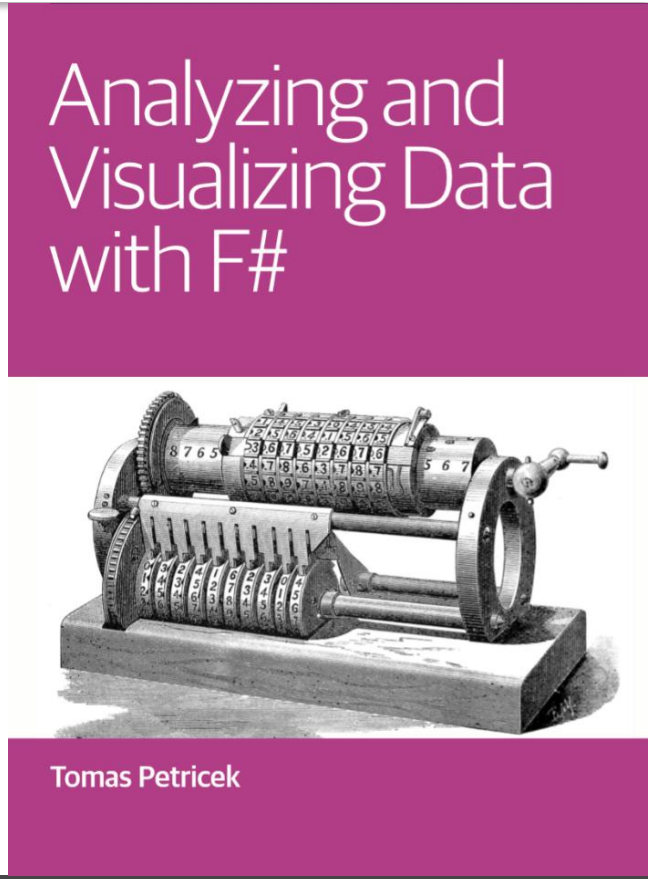
mbrace.io

Как это работает?

Tomas Petricek

Analyzing and Visualizing Data with F#

fslab.org/report

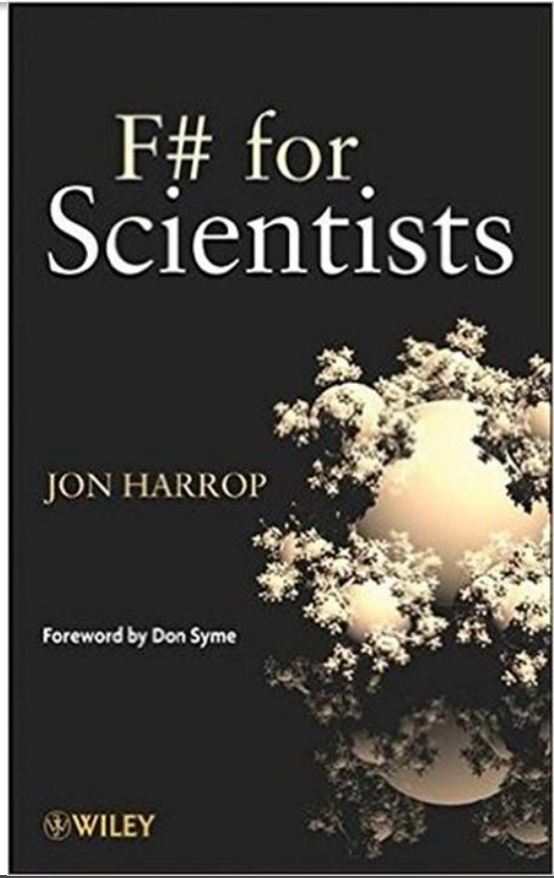


Как это работает?

Jon Harrop

F# for Scientists

amzn.com/0470242116



Data Scientist – Person who is exploring data

Data Scientist – Person who is always learning from data

Спасибо!

 roman_nevolin@epam.com

 nevoroman