

Oracle Code Tokyo 2019

5/17, 2019

変化する時代に Pythonを選ぶ意味

辻 真吾 (@tsjshg)

www.tsjshg.info

Pythonエンジニア育成推進協会 PythonED

- ❖ Python人材育成のため2016年6月に設立
 - ❖ 認定スクール、試験の実施
- ❖ Python3エンジニア認定基礎試験
 - ❖ 教科書 Pythonチュートリアル
- ❖ Python3エンジニア認定データ分析試験（今年秋にβ試験の予定）
 - ❖ Pythonによるあたらしいデータ分析の教科書（翔泳社）



代表理事 吉政さん



顧問理事 寺田さん



監事 佐藤さん



試験問題監修



たかのりさん



辻

Start Python Club

- ❖ 阿久津さんと2人で2015年に開始
- ❖ 月1回虎ノ門C&Rグループ本社ビルで「みんなのPython勉強会」を開催
- ❖ 次回は6月12日(水) 19:00
- ❖ startpython.connpass.com
- ❖ 4,600人を越えるメンバー



The screenshot shows a webpage from Gihyo.jp. The header includes the Gihyo.jp logo and navigation links: デベロッパ, アドミニストレータ, WEB+デザイン, and ライフスタイル. The breadcrumb trail is: gihyo.jp » NEWS & REPORT » インタビュー » “人とのつながり”がもたらすコミュニティの力 — 「Start Python Club」3周年記念インタビュー. The main title of the article is “人とのつながり”がもたらすコミュニティの力 — 「Start Python Club」3周年記念インタビュー. Below the title are social media sharing buttons for Twitter (ツイート), a list button (リスト), a button showing 4 B! (B! 4), and a button showing 0 likes (いいね! 0). The date is 2018年5月7日, and the author is 阿久津剛史, 辻真吾, with the interviewer being 高屋卓也. There are tags for Python, Start Python Club, and コミュニティ. A pagination bar shows 1, 2, 3, 4, with 1 being the active page, and a link to the next page (次のページへ). The article text begins with: Start Python Clubは、初心者から上級者まで幅広いレベルの方々に、PythonとITを学ぶ場を提供することをミッションとして、2015年5月の設立以来、月1回の開催を基本に3年にわたって活動を継続しています。2018年3月にはイベントサイトConnpass上のグループのメンバーが3,500名を超え、日本国内のPythonコミュニティとして大きく成長しています。今回は共同設立者の阿久津剛史氏と辻真吾氏にStart Python Clubのこれまでの活動やこれからの抱負を聞いてみました。

勤め先のイベント

- ❖ 5月31日(金)・6月1日(土)
- ❖ 年に1度のキャンパス一般公開
- ❖ 各研究室の研究紹介のほか、
一般向けの講演会や中高生も
楽しめるクイズ大会など
- ❖ 駒場IIキャンパスなのでご注意
- ❖ www.komaba-oh.jp

NEW X WEEKEND



東京大学
生産技術研究所



東京大学
先端科学技術研究センター

東大駒場リサーチ キャンパス公開2019

5月31日（金）、6月1日（土）



もくじ

- ❖ Pythonのきほん
- ❖ データサイエンスとPython
- ❖ 数学のはなし

“There are only two kinds of languages: the ones people complain about and the ones nobody uses.”

–Bjarne Stroustrup (creator of the C++ programming language.)

完璧なプログラミング言語は存在しないよ。
実際に使われている言語は、常になにかしら文句を言われているでしょ。

Python

- ❖ 1989年のクリスマスにGuido van Rossum氏によって始まったプロジェクト
- ❖ オープンソースで開発されるマルチパラダイムプログラミング言語
- ❖ 最新版は3.7.3
 - ❖ 2.7は2019年いっぱいでおしまい
- ❖ Python Software Foundation (PFS) が知的財産などを管理

```
1 import this
```

The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!

Pythonのダウンロードとインストール

[Python](#)[PSF](#)[Docs](#)[PyPI](#)[Jobs](#)[Community](#)[Donate](#)[GO](#)[Socialize](#)[About](#)[Downloads](#)[Documentation](#)[Community](#)[Success Stories](#)[News](#)[Events](#)

Download the latest version for Mac OS X

[Download Python 3.7.3](#)

Looking for Python with a different OS? Python for [Windows](#),
[Linux/UNIX](#), [Mac OS X](#), [Other](#)

Want to help test development versions of Python? [Pre-releases](#),
[Docker images](#)

Looking for Python 2.7? See below for specific releases



pip コマンドを使ったパッケージ管理

Find, install and publish Python packages
with the Python Package Index



Or [browse projects](#)

179,823 projects

1,319,180 releases

1,896,948 files

329,941 users

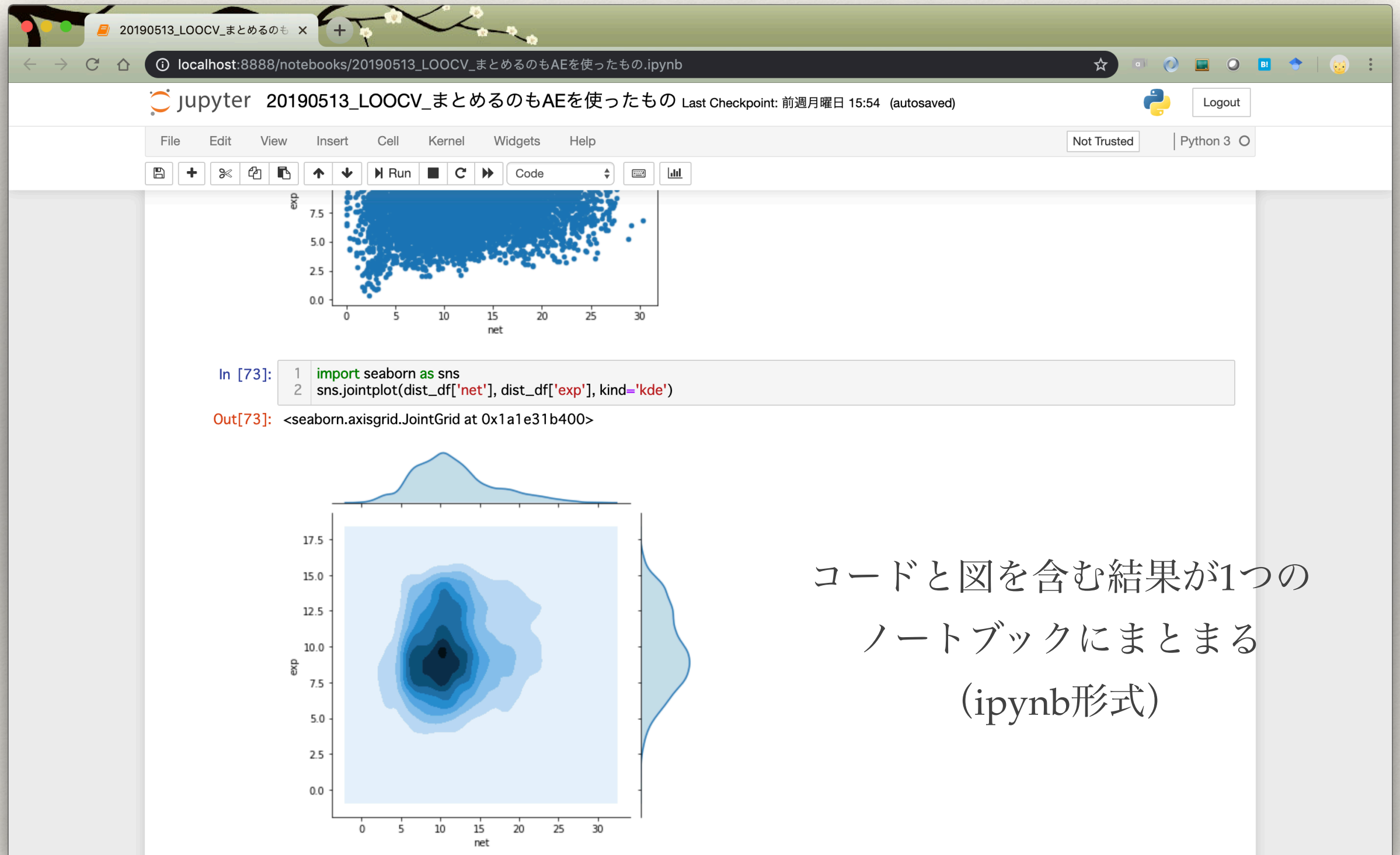


The Python Package Index (PyPI) is a repository of software for the Python programming language.

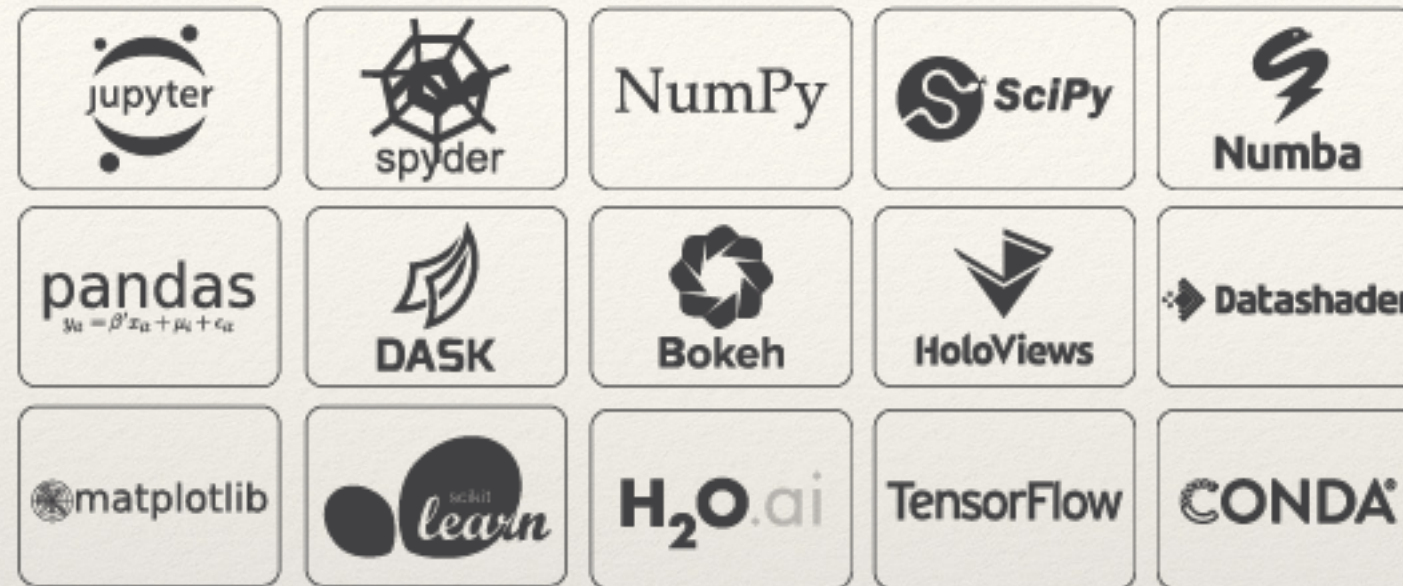
PyPI helps you find and install software developed and shared by the Python community. [Learn about installing packages.](#)

Package authors use PyPI to distribute their software. [Learn how to package your Python code for PyPI.](#)

Jupyter環境の普及



ライブラリの一括導入



 **ANACONDA.** Products

Anaconda Distribution

The World's Most Popular Python/R Data Science Platform

[Download](#)

Pythonのいろいろな実装



Javaによる実装。2.7で開発が止まってしまっている模様



.NET環境での実装

2.7.9が2018年10月にリリースされている



早さを追求した実装

JITコンパイラ技術などが使われているらしい



この後、紹介があります。

これらと区別するために、Cによる標準の実装をCPythonと呼ぶこともあります

Pythonの特徴

シンプルな文法

Python に switch や case 文がないのはなぜですか？

if... elif... elif... else の繰り返しで簡単に同じことができます。switch 文の構文に関する提案がいくつかありましたが、範囲判定をするべきか、あるいはどのようにするべきかについての合意は (まだ) 得られていません。現在の状況の完全な詳細は [PEP 275](#) を参照してください。

非常に大きな数の選択肢から選ぶとき、値を呼び出す関数に対応づける辞書を作れます。例えば:

```
def function_1(...):
    ...

functions = {'a': function_1,
            'b': function_2,
            'c': self.method_1, ...}

func = functions[value]
func()
```

オブジェクトのメソッドを呼び出すには、さらに単純に [getattr\(\)](#) 組み込み関数で特定の名前のメソッドを検索できます:

```
def visit_a(self, ...):
    ...
...

def dispatch(self, value):
    method_name = 'visit_' + str(value)
    method = getattr(self, method_name)
    method()
```


インデントの強制

 bisect.py ×

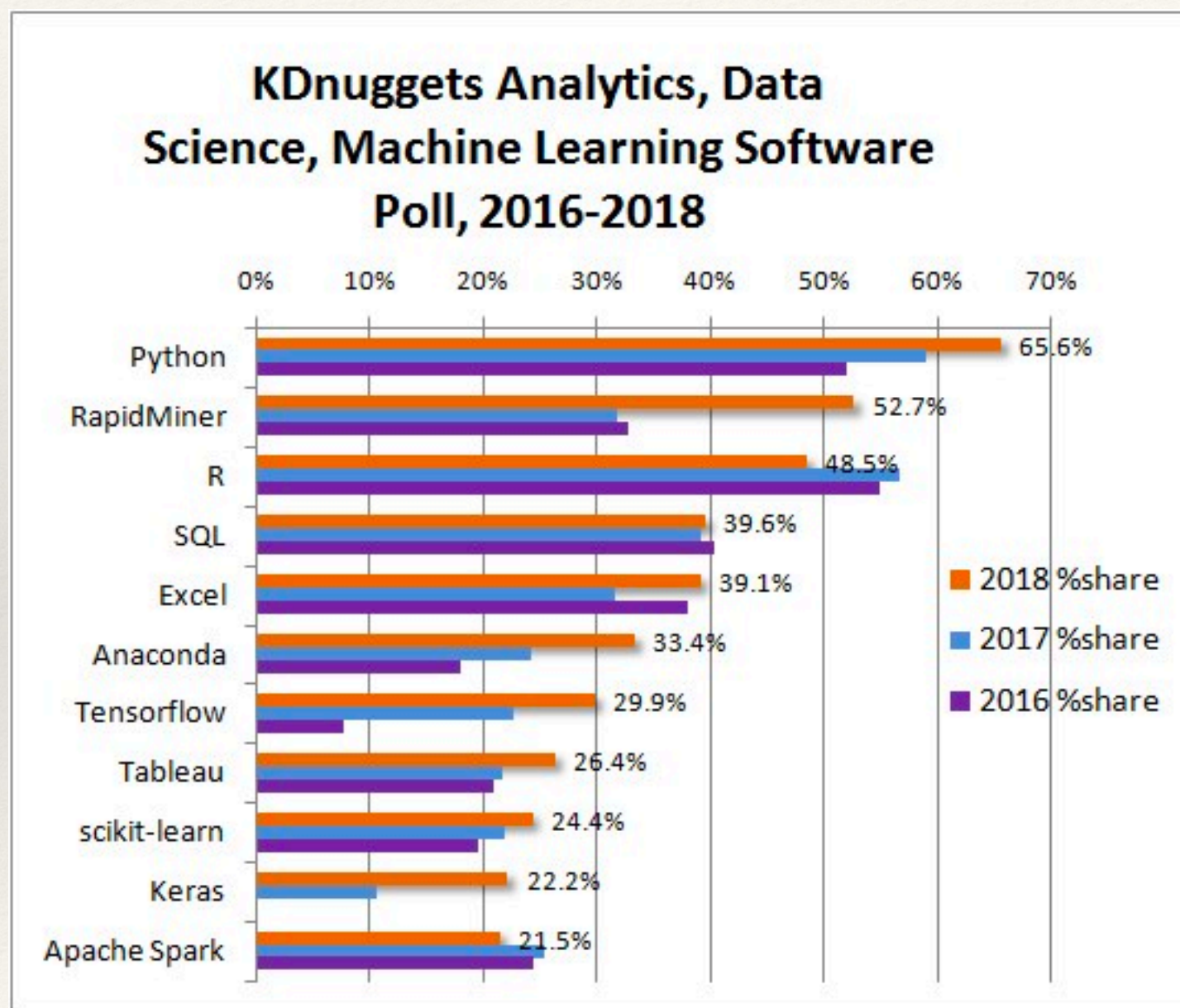
```
1  """Bisection algorithms."""
2
3  def insort_right(a, x, lo=0, hi=None):
4      """Insert item x in list a, and keep it sorted assuming a is sorted.
5
6      If x is already in a, insert it to the right of the rightmost x.
7
8      Optional args lo (default 0) and hi (default len(a)) bound the
9      slice of a to be searched.
10     """
11
12     if lo < 0:
13         raise ValueError('lo must be non-negative')
14     if hi is None:
15         hi = len(a)
16     while lo < hi:
17         mid = (lo+hi)//2
18         if x < a[mid]: hi = mid
19         else: lo = mid+1
20     a.insert(lo, x)
21
```

Pythonが教えてくれること

- ❖ シンプルな文法でも十分プログラミングできる
- ❖ インデントを強制するとコードが読みやすくなる
 - ❖ とくに他の誰かが書いたコード
 - ❖ 過去の自分もほとんど他人

Pythonとデータサイエンス

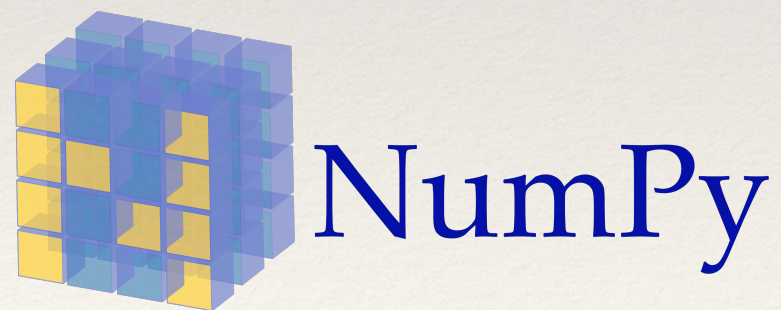
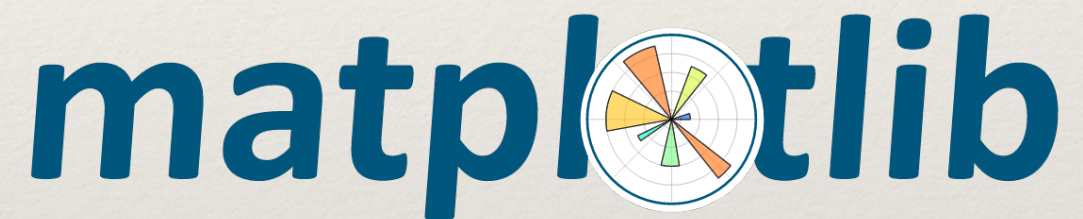
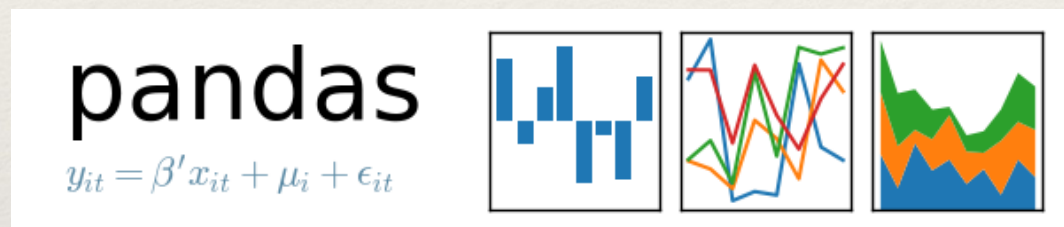
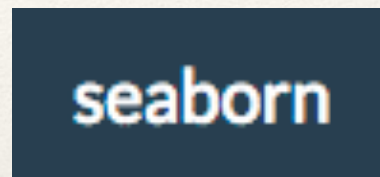
データサイエンスで最も利用されている言語に



Software	2018 % share	% change 2018 vs 2017
Python	65.6%	11%
RapidMiner	52.7%	65%
R	48.5%	-14%
SQL	39.6%	1%
Excel	39.1%	24%
Anaconda	33.4%	37%
Tensorflow	29.9%	32%
Tableau	26.4%	21%
scikit-learn	24.4%	11%
Keras	22.2%	108%

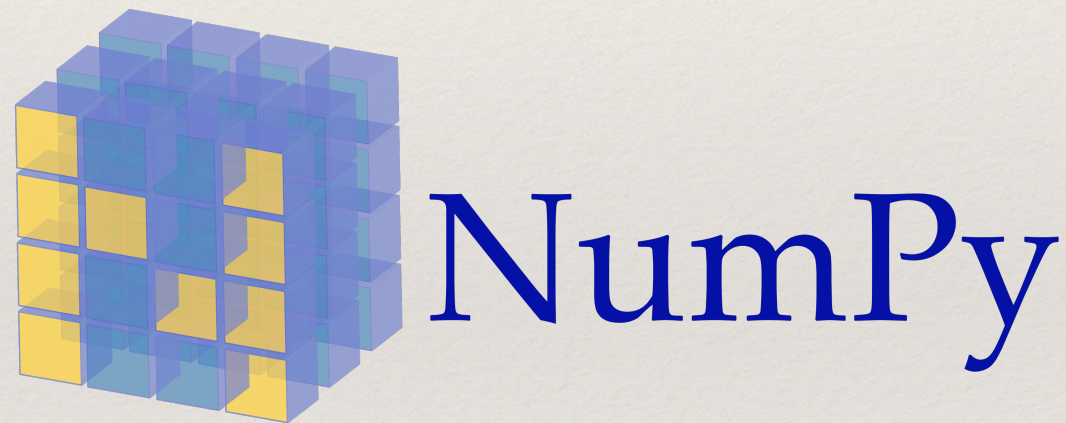
重力波の観測（2017）やブラックホールの可視化（2019）などでも活用されている

Pythonはglue (のり) 言語



数値計算の高速化

- ❖ Pythonはインタプリタ言語なので速くはない
- ❖ ベクトルや行列の計算を高速化するNumPy



LAPACK(Linear Algebra PACKage)

BLAS (Basic Linear Algebra Subprograms)

PythonはもともとCなどで書かれたネイティブなライブラリとの連携が容易

Pythonで数学

Timeline

Countries

Results

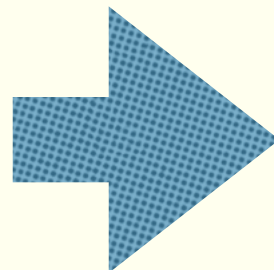
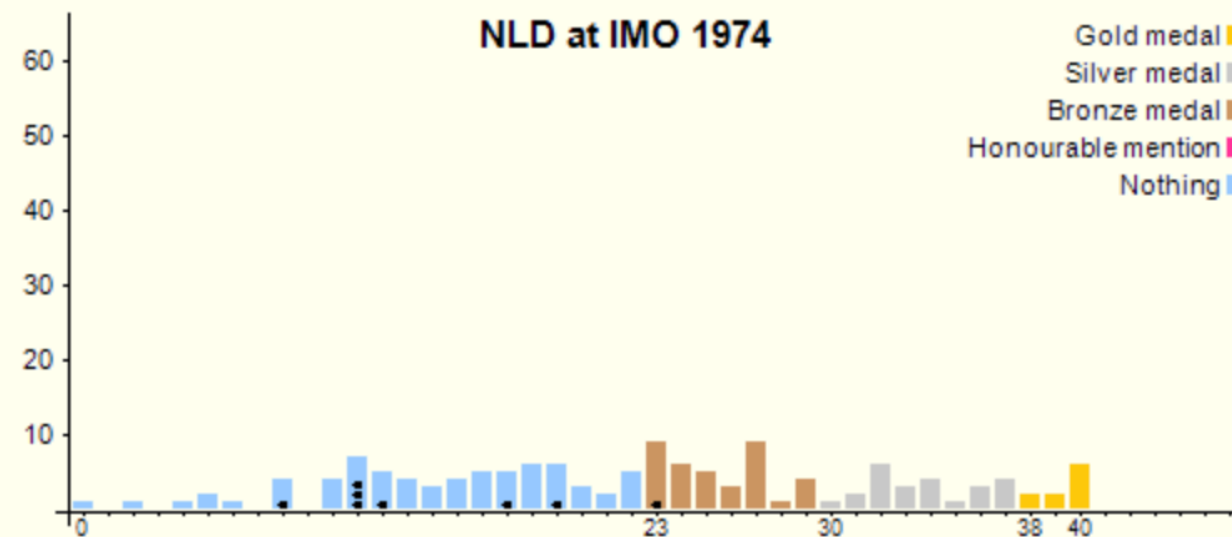
Search

Problems

Hall of fame

About IMO

Links and Resources

◀ **NETHERLANDS** ▶◀ **16TH IMO 1974** ▶

Contestant [♀♂]	P1	P2	P3	P4	P5	P6	Total	Rank		Award
								Abs.	Rel.	
Guido van Rossum	4	6	0	6	7	0	23	63	55.40%	Bronze medal
Bernard ter Haar	5	4	0	5	5	0	19	82	41.73%	
Hantsje Zantema	5	1	0	6	5	0	17	94	33.09%	
Jacobus Cornelis Groeneveld	5	0	0	6	1	0	12	115	17.99%	
Eric Jan van der Marel	4	0	0	5	1	1	11	120	14.39%	
Antonie Wijnand Frederik Rutgers	5	0	0	6	0	0	11	120	14.39%	
Peter Christiaan Schmale	5	0	0	6	0	0	11	120	14.39%	
Jos van der Hijden	5	0	0	3	0	0	8	131	6.47%	
Team results	38	11	0	43	19	1	112	15	17.65%	B

Leader: **Ary van Tooren**Deputy leader: **Jan van de Craats**

Results may not be complete and may include mistakes. Please send relevant information to the webmaster: webmaster@imo-official.org.

Guido先生、1974年の国際数学オリンピックで銅メダル（ただの天才）

素数

- ❖ 1とその数自身でしか割り切ることができない数
 - ❖ 2, 3, 5, 7, 11, 13,...
- ❖ 無数にあることが証明できます

メルセンヌ数

- ❖ $2^n - 1$ の形で表現できる数
- ❖ ときどき素数になる
 - ❖ メルセンヌ素数



n=2203は664桁の素数

```
1 2**2203 - 1
```

```
1475979915214180235084898622737381736312066145333169775147771216478570297878078949
3774073370493892893827485075314964804772812648387602591918144633653302695404969612
0111343015690239609398909022625932693502528140961498349938822283144859860183431853
6230923772641390209490231836446899608210795482963763094236630945410832793769905399
9824571863229447296364188906233721717237421056364403682184596496329485386969058726
5048691443463745750728044182367681351785209934866084717257940842231667809767022401
1990280170474894487426924742108823536808485072502240519452587542875349976558572670
2296339625752126374778977855015526465226099888699140135404838098656812504194976866
97771007
```

Python3からメモリの許す限り大きな整数を扱うことができるようになっていきます

フェルマーの小定理

p が素数で p と a が互いに素（共通の約数が1だけ）
のとき次の式が成り立つ

$$a^{p-1} \equiv 1 \pmod{p}$$

やってみよう！

In [*]:

```
1 a = 2
2 p = 2**2203 - 1
3 a**(p-1) % p
```

答えは1だが、さすがに計算に時間がかかる・・・。

組み込み関数pow

冪乗を計算してくれるpow関数が3つ目の引数をとれる

In [2]:

```
1 a = 2  
2 p = pow(2, 2203) - 1  
3 pow(a, p-1, p)
```

Out[2]: 1

数学愛にあふれる小粋な実装

激しく変化する時代

- ❖ それでも変わらないものがあるはず
- ❖ 本質的なもの変わらない
 - ❖ のではないか？



Pythonが見せてくれる本質

- ❖ プログラミングスタイル
- ❖ C言語との連携
- ❖ 数学とアルゴリズム

人生迷子にならないように、Pythonをお供に生きていきましょー

ご静聴ありがとうございました