

Анализ данных **и** Машинное обучение в гидрологии

...

Неделя 2

План

Лекция

- жизнерадостная презентация с
Виноградовских чтений "Язык
программирования Python в
гидрологическом моделировании"
- рефлексия

Практическое занятие

- git и github
- получение удаленного репозитория
- знакомство со средой разработки
(Jupyter notebook)
- Первые шаги в python

Язык программирования Python в гидрологическом моделировании*

Георгий Айзель

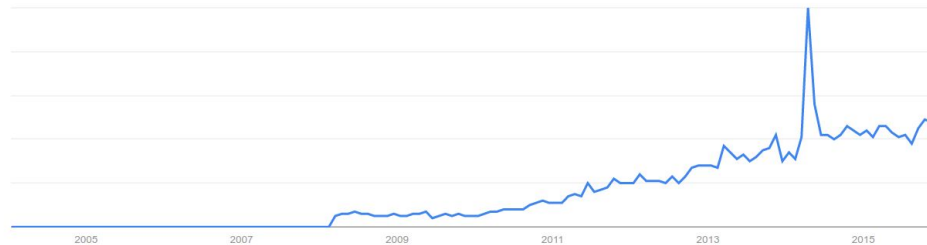
Институт водных проблем РАН

В мире

“Стартап”

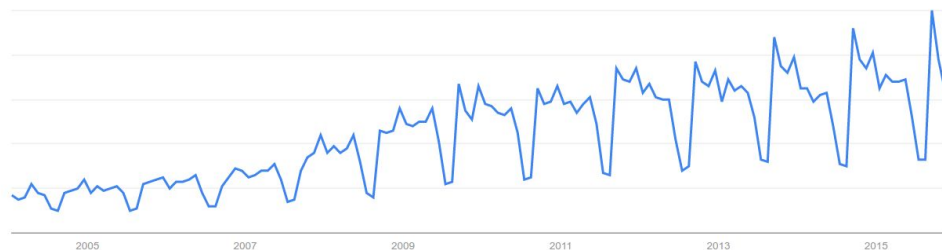
Динамика популярности ?

☐ Поиск по новостям ☐ Прогноз ?



Динамика популярности ?

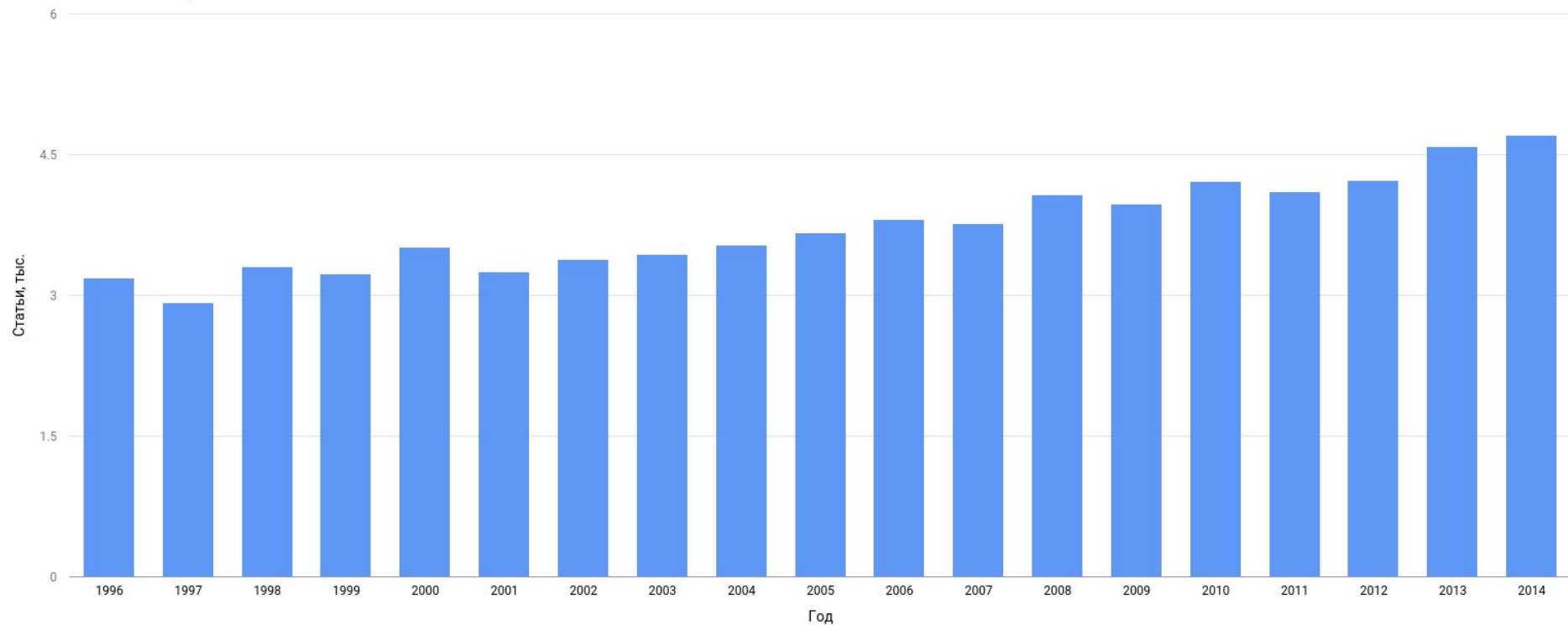
☐ Поиск по новостям ☐ Прогноз ?



“Наука”

Мы больше публикуемся

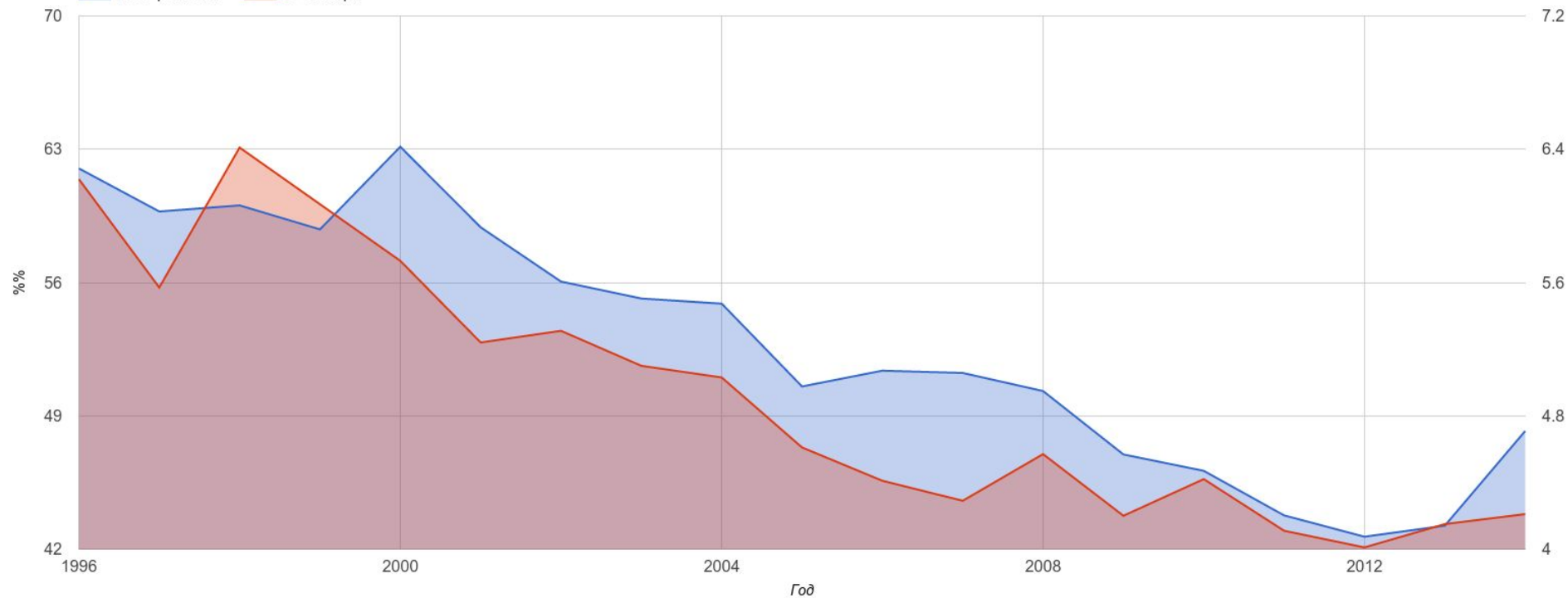
Тематика: науки о Земле



Но другие публикуются быстрее

Сравнительный анализ количества статей

— % от региона — % от мира



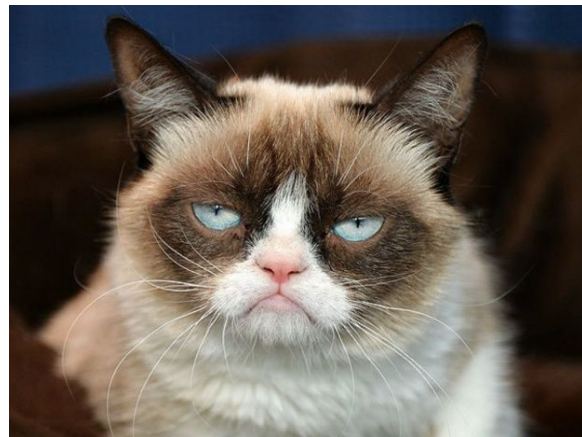
ЕСЛИ МЫ ЛУЧШЕ ВСЕХ



**ТО ПОЧЕМУ ТАК МНОГО ТЕХ
КТО ЛУЧШЕ НАС?**

Причины?

1. Недостаточное финансирование;
2. Сокращение числа исследователей;
3. Отсутствие экспериментальных полевых работ;
4. Уныние;
5. вот это всё.



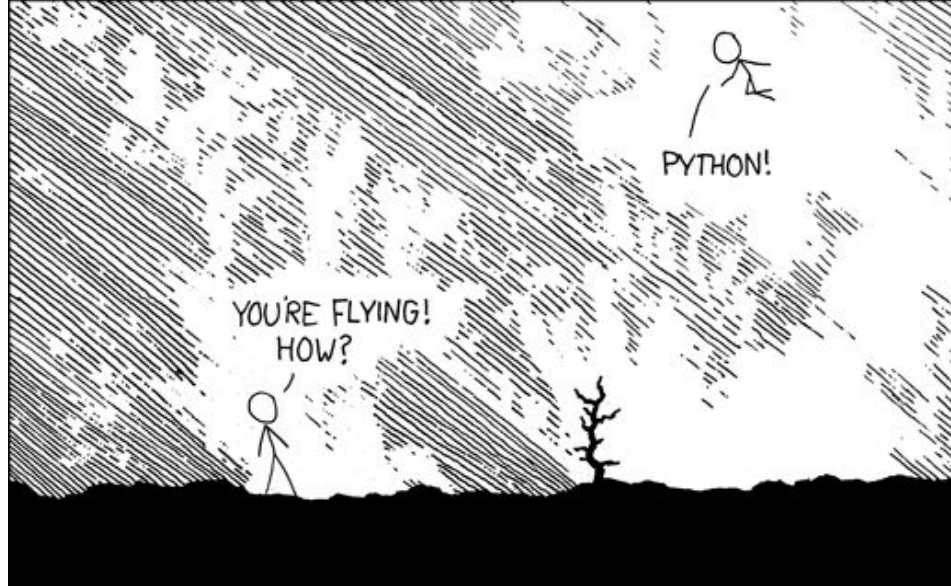
Причина (ИМХО) -- медленная проверка гипотез



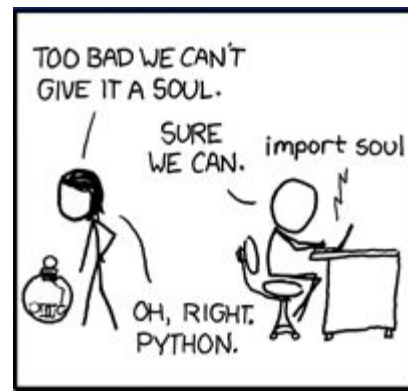
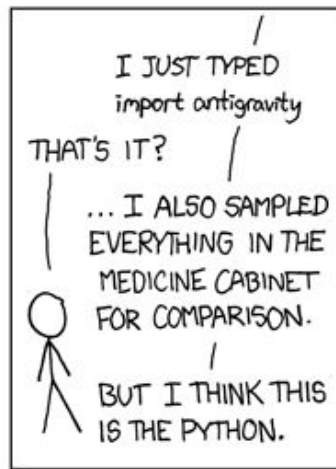
Проверка гипотез



Как сделать быстрее,
выше, сильнее?



Почему Python?



Forming →

Norming →

Storming →

Performing

Forming (получение данных)

- txt
- csv
- netcdf
- sql
- xml
- web api

1. import ...
2. path =
3. connection =
4. data = parse(path)

profit!

Norming (предварительная обработка данных)

- сортировка
- группировка
- заполнение пропусков
- удаление выбросов
- создание новых переменных
- нормирование

```
1. import numpy as np
2. import pandas as pd
3. from sklearn import
   Preprocessing
4. library.method()

profit!
```

Storming (моделирование, анализ)

- классификация
- кластеризация
- регрессия
- распознавание образов
- моделирование
- прогнозирование

```
1. from sklearn import SVR
2. model = SVR()
3. model.fit(X, y)
4. metrics(model)
5. model.predict(y)

profit!
```


Performing (представление результатов)

- научная графика
 - ❑ Matplotlib, Seaborn
- воспроизводимые результаты
 - ❑ Jupyter notebook, Docker, Git(hub)
- переиспользование кода
 - ❑ OOP, Gist
- создание веб-приложений
 - ❑ Flask

Ой, а я начал учить R,
мне теперь придется
все бросить и учить
Python?

Нет.

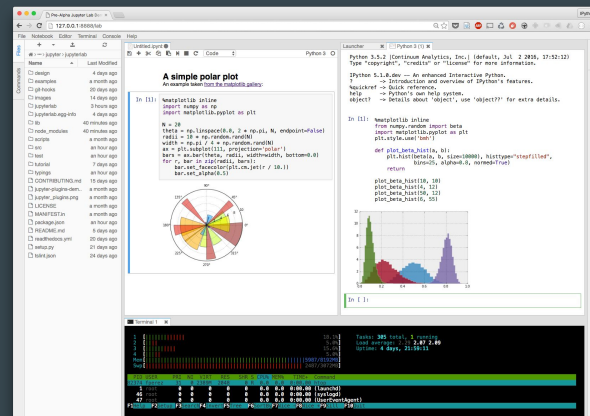
R тоже клёвый.

Прошел год

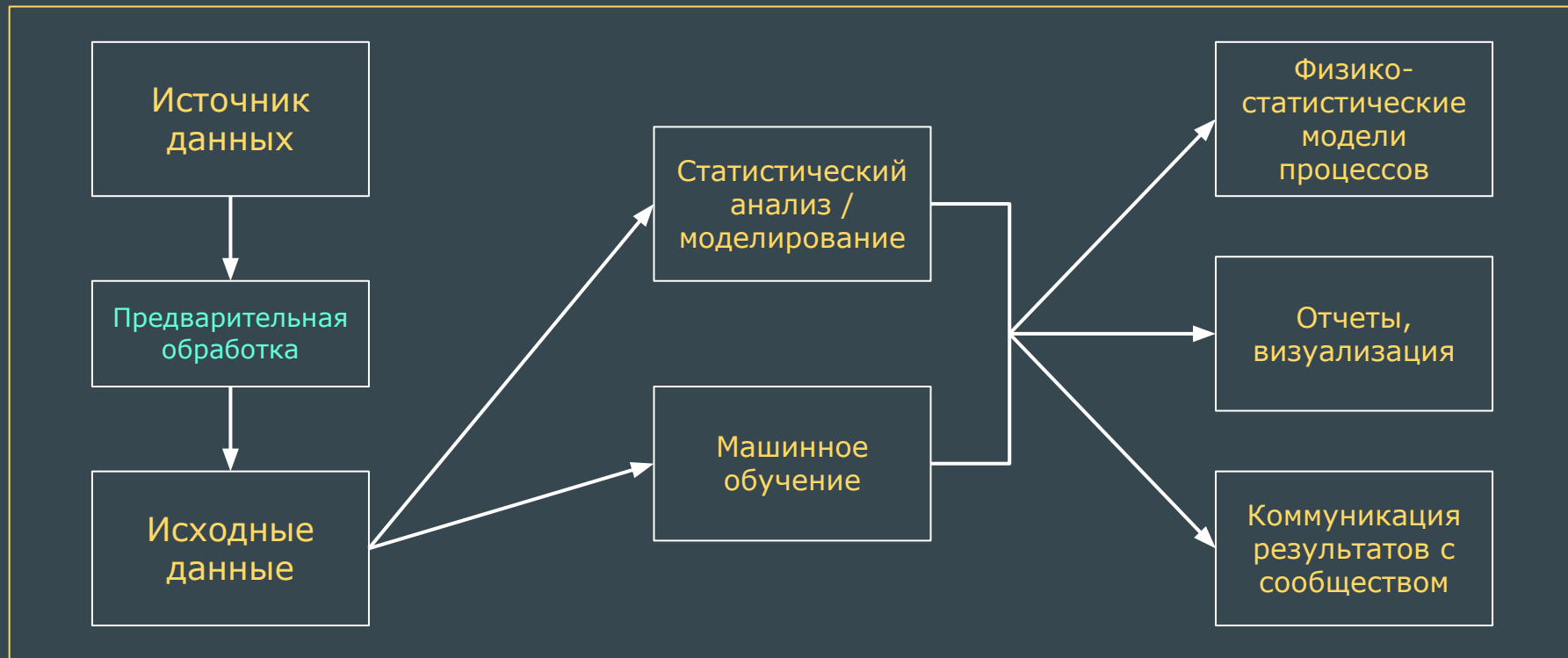
Итоги

- + Python - лучший выбор;
- + Большое количество специализированных библиотек (numpy, pandas, scipy);
- + Хорошие библиотеки написаны на C++ и имеют API для использования в python (tensorflow, scikit-learn);
- + Можно импортировать fortran код;
- + Высокая скорость разработки;
- + Читаемый код.

- выбор между 2.7 и 3.5;
- Библиотеки лишают вас творчества;
- [JupyterLab](#) все еще в alpha.



Среда воспроизводимых вычислений



Где научиться

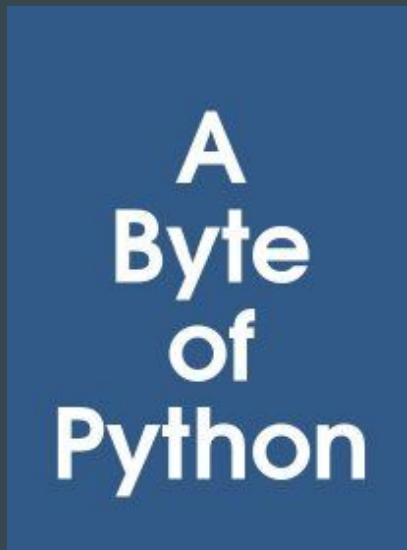
+ Coursera;

+ edX;

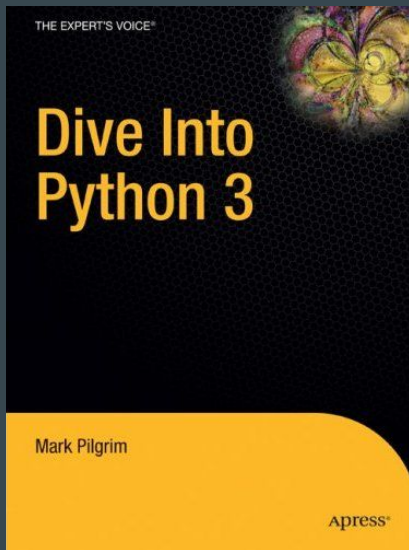
+ Udacity;

+ Stepik.

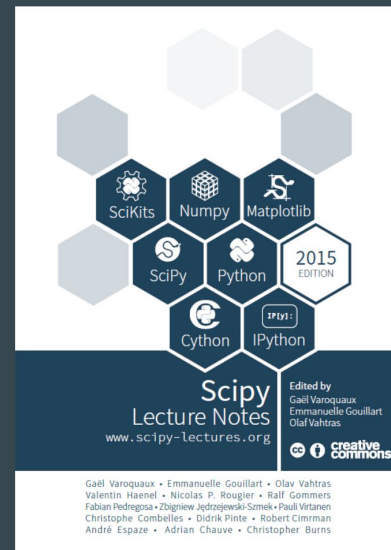
Что почитать/посмотреть



python.swaroopch.com



diveintopython.net



scipy-lectures.org

Важно

Вы можете помочь существенно улучшить этот курс!

- ayzelgv@gmail.com, hydrogo@yandex.ru
- vk.com/ayzelgv, facebook.com/ayzelgv
- ИВП РАН, кабинет 617