

法律声明

- 本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。



关注 小象学院

第一讲



工作环境准备及 数据分析基础

--Robin

目录

- 课程介绍
- 工作环境准备
- Python进阶技巧
- 科学计算库NumPy
- 实战案例1-1：中国五大城市PM2.5数据分析（1）

目录

- 课程介绍
- 工作环境准备
- Python进阶技巧
- 科学计算库NumPy
- 实战案例1-1：中国五大城市PM2.5数据分析（1）

课程介绍

面向人群：

1. 想了解和学习典型的**数据分析流程**和实践的学习者
2. 想接触和学习**非结构化数据**(文本、图像、时间序列等)分析的学习者
3. 想学习数据分析中**常用建模知识**的相关从业人员
4. 尚不会**使用Python处理数据分析**的从业者
5. 想转行**从事数据分析师行业**的学习者
6. 想**使用Python实现机器学习或深度学习**的工程师

课程介绍

课程目标：

1. 熟悉数据分析的流程，包括数据采集、处理、可视化、数据建模等
2. 掌握Python语言作为数据分析工具，从而有能力驾驭不同领域数据分析实践
3. 掌握非结构化数据的处理与分析
4. 快速积累多个业务领域数据分析项目经验，包括文本数据、图像数据及时间序列
5. 掌握使用Python实现基于机器学习及深度学习的数据分析和预测
6. 掌握数据分析中常用的建模知识

课程介绍

《Python人工智能》课程安排				
	标题	内容	实战案例	上课时间
第一课	工作环境准备及数据分析基础	1. 课程介绍 2. 工作环境准备 3. Python进阶技巧 4. 科学计算库NumPy	1-1. 中国五大城市PM2.5数据分析 (1)	2018/03/03 15:00-17:00
第二课	Pandas进阶及统计分析	1. 基本数据对象及操作 2. 数据清洗 3. 数据合并及分组 4. 透视表	1-2. 中国五大城市PM2.5数据分析 (2)	2018/03/04 15:00-17:00
第三课	数据展示及可视化	1. 数据可视化的重要性--Anscombe's quartet 2. 基本图表的绘制及应用场景 3. 数据分析常用图表的绘制 4. Pandas及Seaborn制图 5. 其他常用的可视化工具 --D3.js, ECharts	2. YouTube视频趋势分析	2018/03/10 15:00-17:00
第四课	Python机器学习(1)	1. 机器学习基本概念与流程 2. Python机器学习库scikit-learn 3. 机器学习常用算法介绍及演示(1) -- KNN, 线性回归, 逻辑回归, SVM, 决策树	3-1. 手机价格预测 (1)	2018/03/11 15:00-17:00
第五课	Python机器学习(2)	1. 模型评价指标及模型选择 2. 集成学习 -- Bagging, Boosting, Stacking, 集成规则 3. Boosting框架Xgboost	3-2. 手机价格预测 (2)	2018/03/17 15:00-17:00
第六课	图像数据处理及分析	1. 计算机视觉库OpenCV 2. 图像数据基本概念及操作 3. 常用的图像特征描述 4. 常用的聚类算法	4-1. 时尚商品图片分类(Fashion-MNIST) (1)	2018/03/18 15:00-17:00
第七课	神经网络及深度学习CNN	1. 神经网络 2. 深度学习 3. TensorFlow框架学习及使用 4. TensorFlow实现卷积神经网络 (CNN)	4-2. 时尚商品图片分类(Fashion-MNIST) (2)	2018/03/24 15:00-17:00
第八课	时间序列分析及深度学习RNN	1. 时间序列基础 2. 时间序列基本操作 3. 循环神经网络RNN 4. Keras框架学习及使用	5. 比特币价格分析	2018/03/25 15:00-17:00
第九课	文本数据分析	1. 自然语言处理及NLTK 2. 文本数据处理 3. “词袋”模型 4. 朴素贝叶斯	6. 垃圾短信检测	2018/03/31 15:00-17:00

数据处理
及分析

机器学习
及建模

多个领域
建模实践

疑问

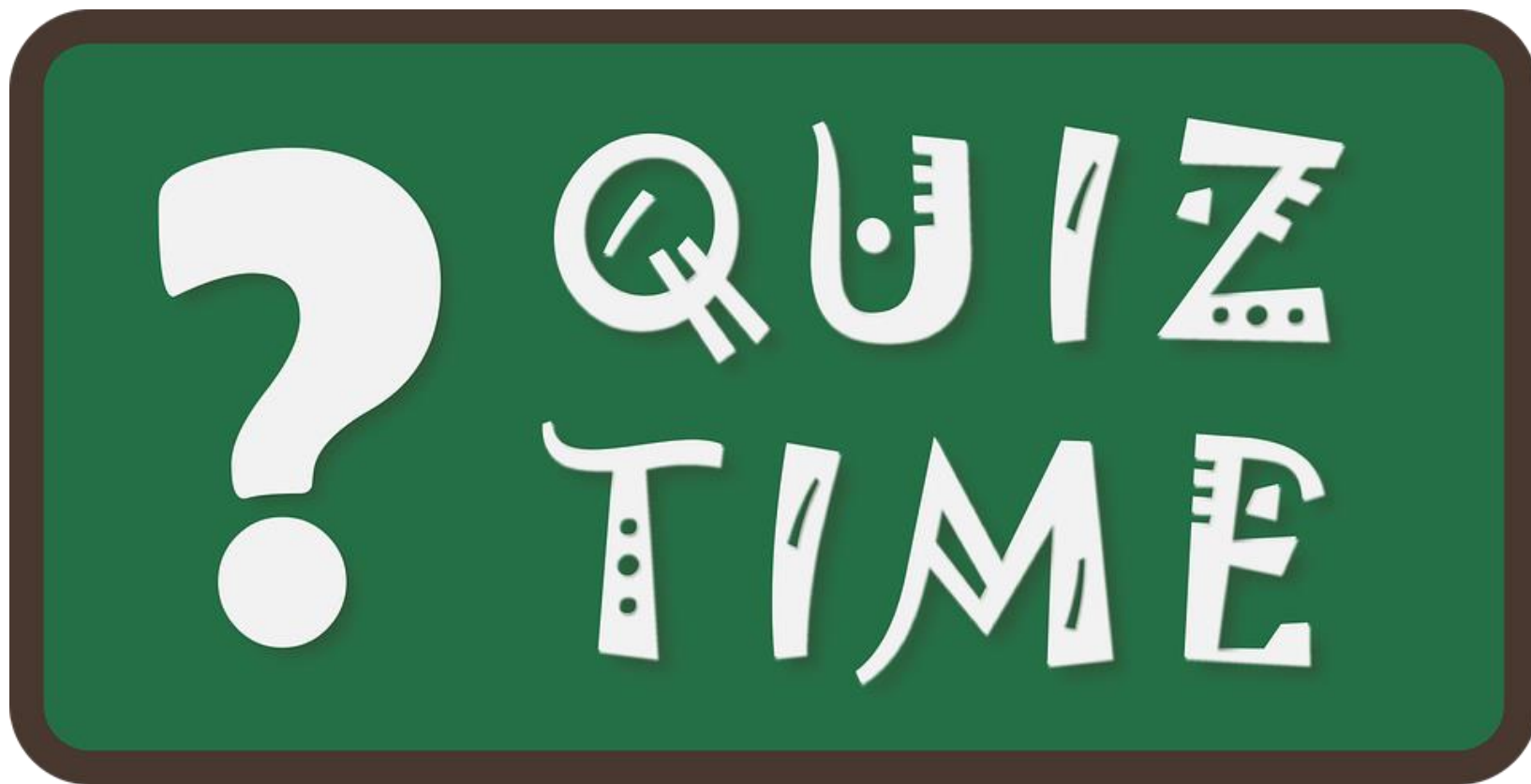
□ 问题答疑：<http://www.xxwenda.com/>

■ 可邀请老师或者其他人回复问题

小象问答邀请 @Robin_TY 回答问题



随堂测验



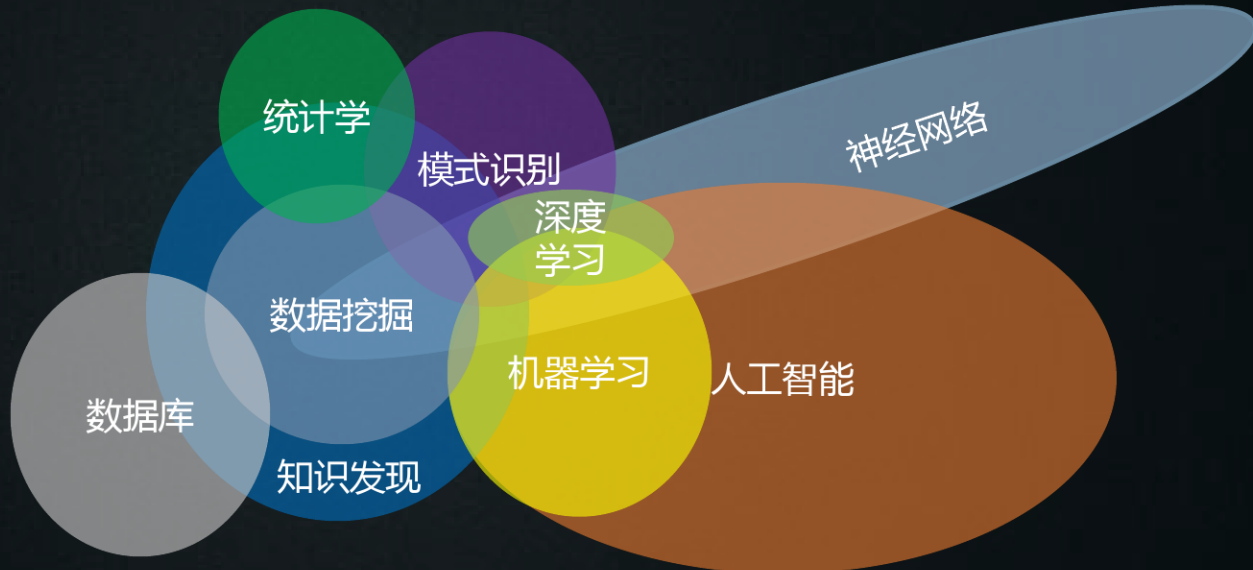
人工智能

PART ONE AI 知识图谱

一图看懂人工智能大家庭

- » 什么是人工智能？它和神经网络、机器学习、深度学习、数据挖掘这类热门词汇有什么关系？撇开复杂的概念和高冷的定义，一图看懂人工智能相关领域的错综复杂的关系。
- » 由图可见，人工智能、机器学习、深度学习并非是层层包含的关系，而最近火热的神经网络也只是与人工智能有交叉而非人工智能的实现方式或者子集。

» 人工智能相关领域关系图



P1

数据来源：SAS，2014 and PwC，2016

头条 今日头条
你关心的 才是头条

人工智能

PART ONE AI 知识图谱

人工智能产业结构图



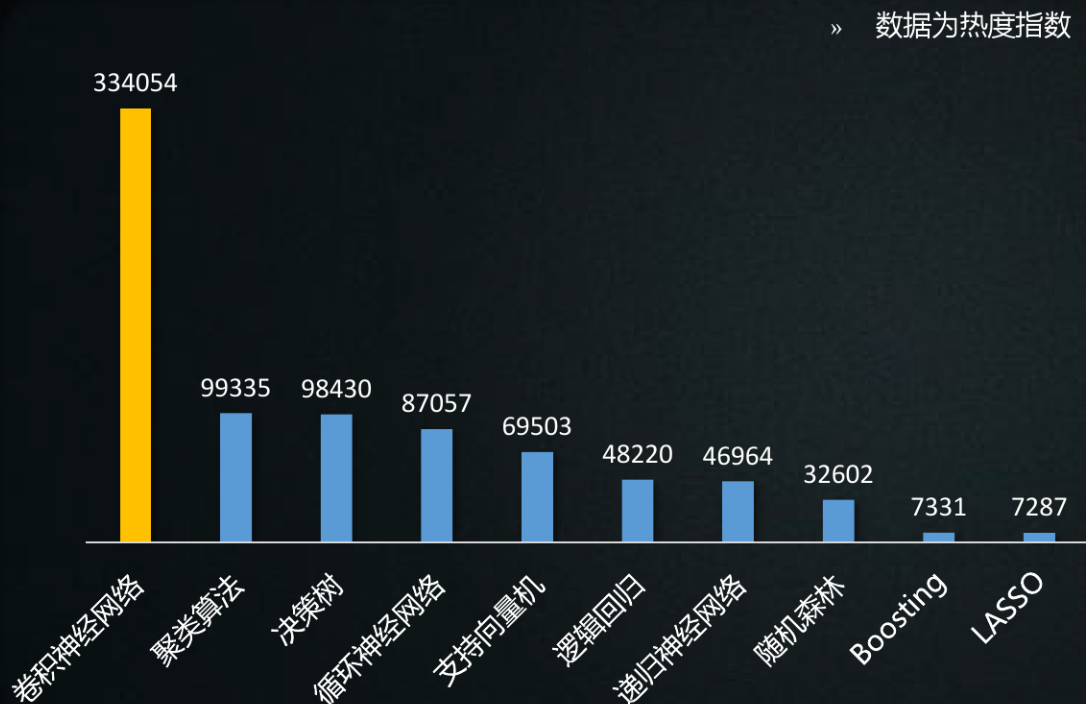
P6

头条 今日头条
你关心的 才是头条

人工智能

PART TWO AI 公司影响力

人工智能十大热门算法

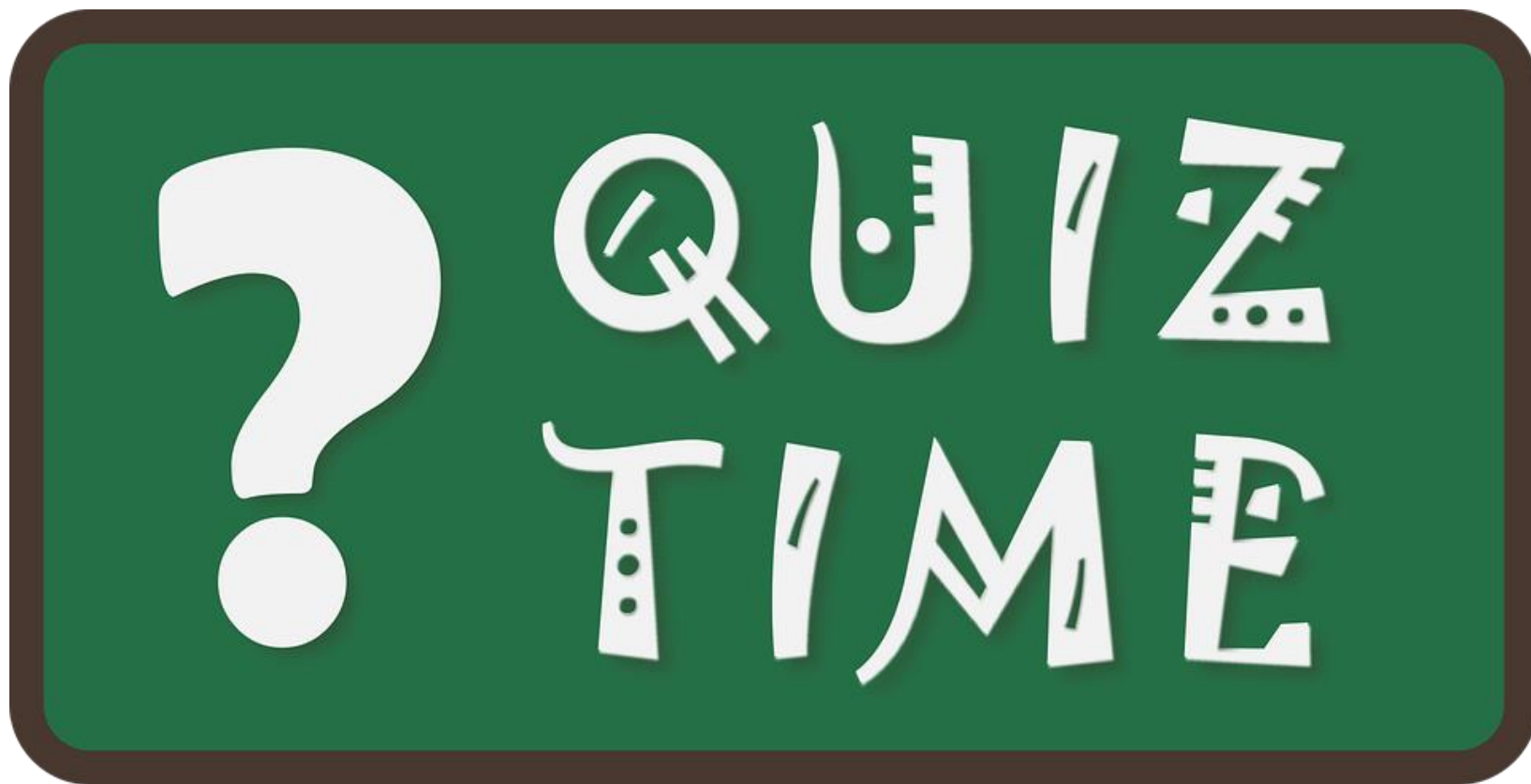


- » 人工智能的核心技术就是算法。
- » 排在第一的卷积神经网络，即CNN，是一种强大的图像识别任务处理模型，它将输入的图像通过卷积层抽象化。
- » 这项算法因为在谷歌AlphaGo对战李世石比赛中所应用，而名声大振。AlphaGo 的胜利证明了卷积神经网络的强大和通用性。

— P10 —

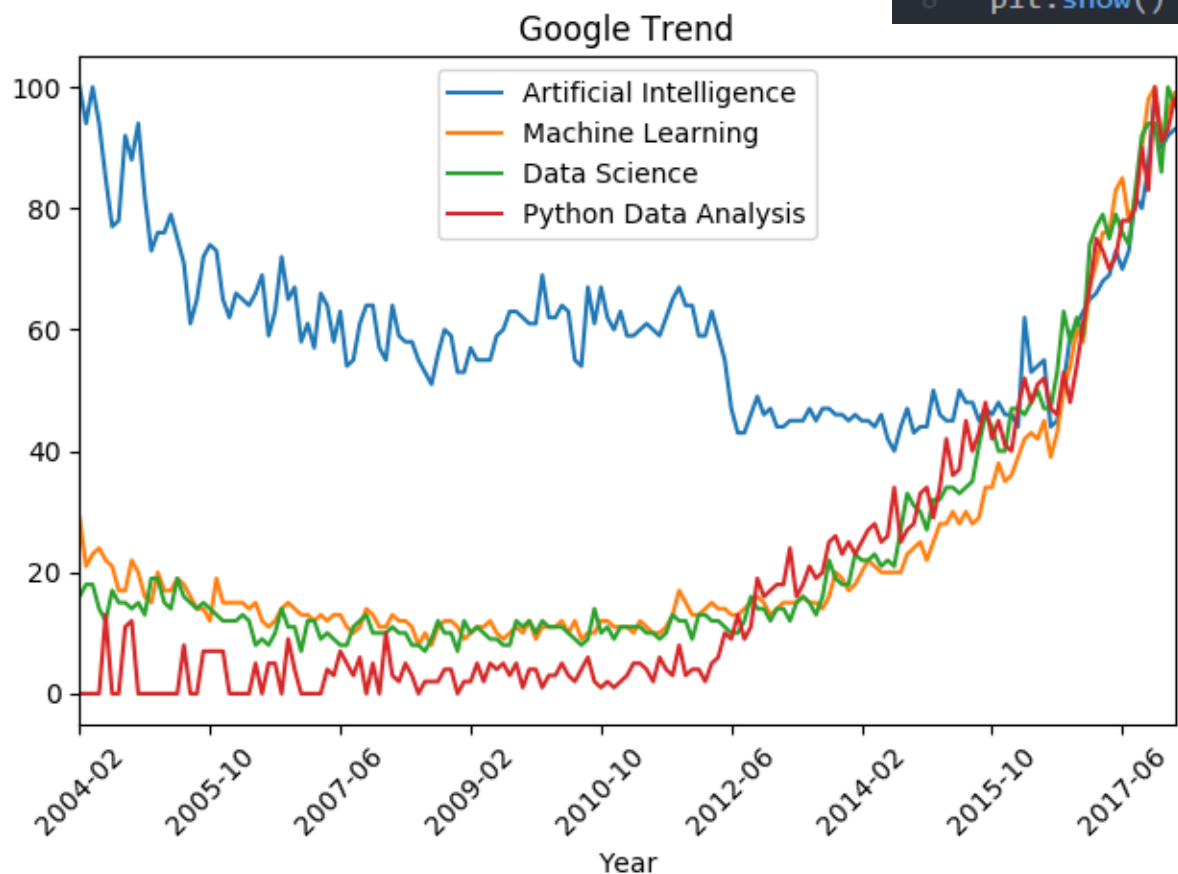


随堂测验



从Python数据分析入手

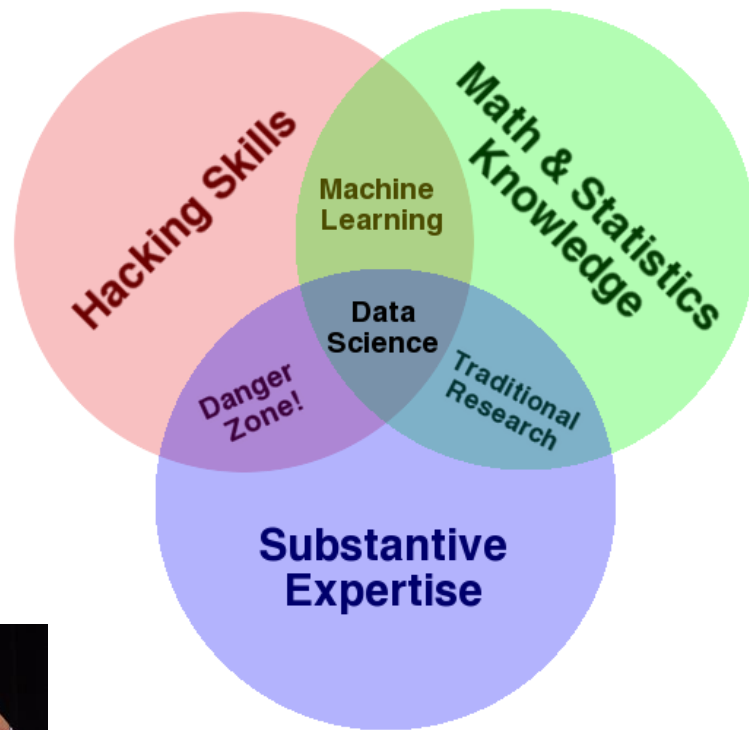
```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 df = pd.read_csv('./trend.csv', index_col='Year')
5 df.plot(title='Google Trend', rot=45)
6 plt.tight_layout()
7 plt.savefig('./google_trend_plot.png')
8 plt.show()
```



数据科学

Drew Conway认为数据科学包括：

- 黑客技术
 - 如编程能力、
 - 向量化操作和算法思想
- 数学和统计知识
 - 如常见的分布、最小二乘法
- 实质性的专业知识



<http://drewconway.com/>

数据科学

数据科学涉及到的操作 by David Donoho

1. 数据探索与准备

- 数据操作、清洗等

2. 数据展现形式与转化

- 不同格式的数据操作，表格型、图像、文本等

3. 关于数据的计算

- 通过编程语言（Python或R）计算分析数据

4. 数据建模

- 预测、聚类等机器学习模型

5. 数据可视化与展示

- 绘图、交互式、动画等

6. 数据科学涉及到的学科知识



<https://statweb.stanford.edu/~donoho/>

50 Years of Data Science

数据科学

何谓数据分析

- 用适当的**统计分析方法**对收集来的**大量数据**进行分析，提取**有用信息**和形成**结论**对数据加以**详细研究**和**概括总结**的过程



数据分析的目的

- 从数据中挖掘规律、验证猜想、进行预测

数据科学

原图地址：[点击下载](#)



目录

- 课程介绍
- 工作环境准备
- Python进阶技巧
- 科学计算库NumPy
- 实战案例1-1：中国五大城市PM2.5数据分析（1）

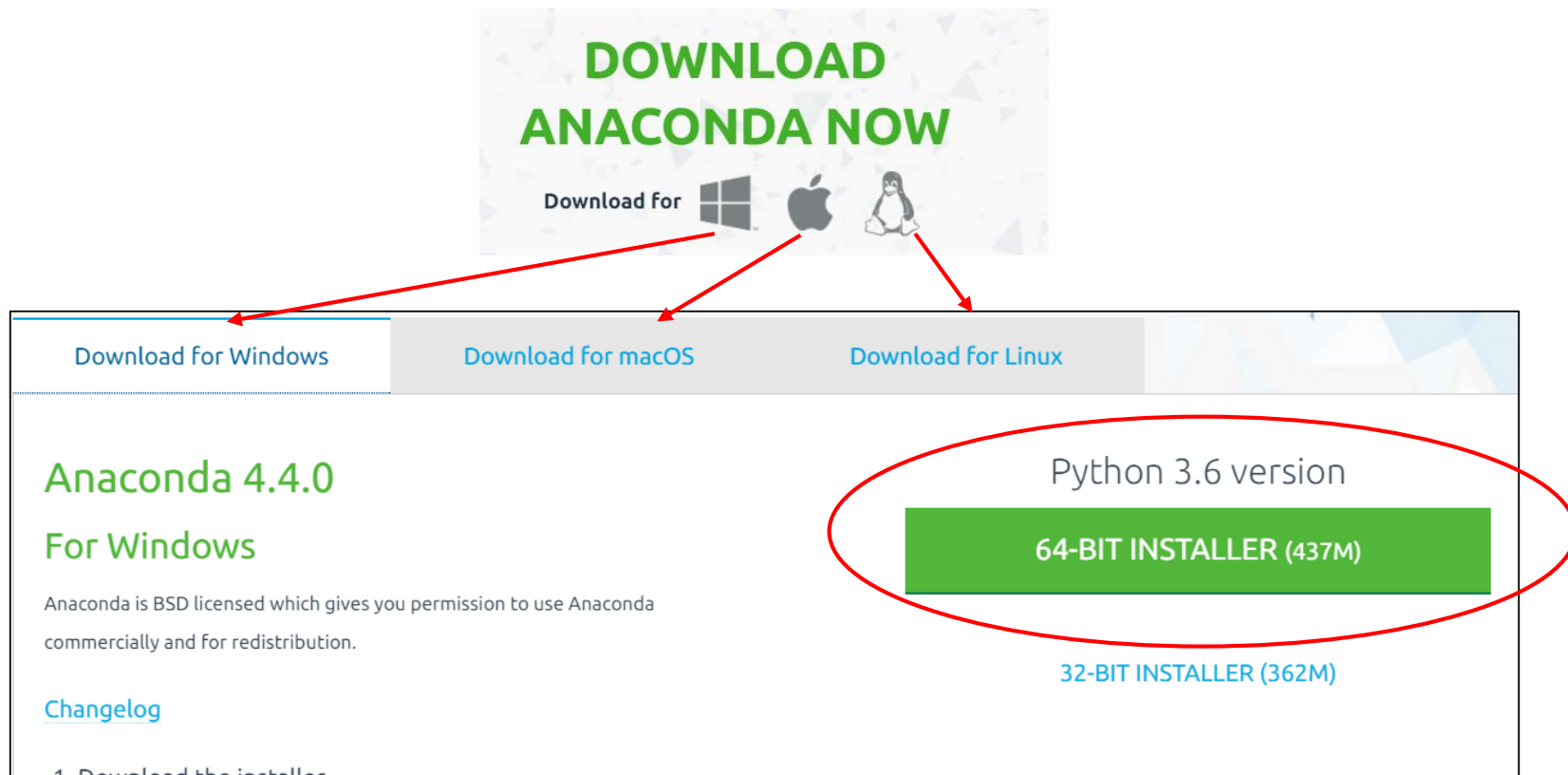
工作环境准备

安装Anaconda

- Anaconda是Python的一个科学计算发行版，内置了数百个Python经常会使用的库，也包括做机器学习或数据挖掘的库，如Scikit-learn、NumPy、SciPy和Pandas等，其中可能有一些是TensorFlow的依赖库
- Anaconda提供了一个编译好的环境可以直接安装
- Anaconda自动集成了最新版的MKL（Math Kernel Library）库，加速矩阵运算和线性代数运算
- Anaconda <https://www.continuum.io/downloads>
- 根据操作系统下载对应版本的64位的Python3.x版



工作环境准备

安装Anaconda



The image shows the Anaconda download page. At the top, a green banner says "DOWNLOAD ANACONDA NOW" with icons for Windows, macOS, and Linux. Below this, three tabs are visible: "Download for Windows", "Download for macOS", and "Download for Linux". The "Download for Windows" tab is selected. On the left, it says "Anaconda 4.4.0 For Windows" and provides a brief description of the BSD license. On the right, there are two green buttons: "64-BIT INSTALLER (437M)" and "32-BIT INSTALLER (362M)". A red oval highlights the "64-BIT INSTALLER (437M)" button. Red arrows point from the "Download for Windows", "Download for macOS", and "Download for Linux" tabs to the "64-BIT INSTALLER (437M)" button.

DOWNLOAD ANACONDA NOW

Download for   

[Download for Windows](#) [Download for macOS](#) [Download for Linux](#)

Anaconda 4.4.0
For Windows

Anaconda is BSD licensed which gives you permission to use Anaconda commercially and for redistribution.

[Changelog](#)

1. Download the installer

Python 3.6 version

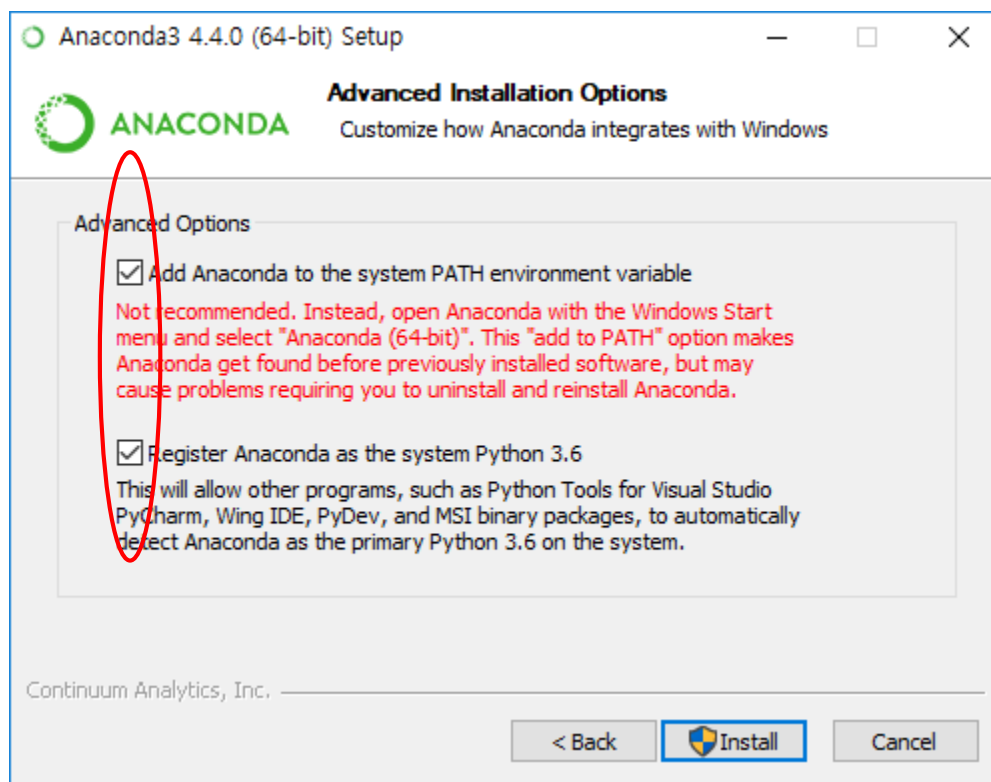
64-BIT INSTALLER (437M)

32-BIT INSTALLER (362M)

工作环境准备

安装Anaconda

确认勾选将Python添加到系统环境变量



工作环境准备

Python包管理

- 安装: `pip install xxx`, `conda install xxx`
- 卸载: `pip uninstall xxx`, `conda uninstall xxx`
- 升级: `pip install --upgrade xxx`, `conda update xxx`
- 详细用法: <https://pip.pypa.io/en/stable/reference/>

Python虚拟环境

- Virtualenv: <https://virtualenv.pypa.io/en/stable/userguide/>
- conda 虚拟环境: <https://conda.io/docs/using/envs.html>

多版本Python管理

- conda管理: <https://conda.io/docs/py2or3.html>

工作环境准备

1. 命令行输入python

```
Python 3.5.2 |Anaconda custom (64-bit)| (default, Jul 5 2016, 11:41:13)
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello Python')
Hello Python
```

2. 命令行输入ipython

```
Python 3.5.2 |Anaconda custom (64-bit)| (default, Jul 5 2016, 11:41:13)
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0 -- An enhanced Interactive Python.
?                -> Introduction and overview of IPython's features.
%quickref        -> Quick reference.
help             -> Python's own help system.
object?         -> Details about 'object', use 'object??' for extra details.

In [1]: print('Hello Python')
Hello Python

In [2]:
```


工作环境准备

IDE

- Jupyter notebook
 1. Anaconda自带，无需单独安装
 2. 记录思考过程，实时查看运行过程
 3. 基于web的在线编辑器（本地）
 4. .ipynb文件分享
 5. 可交互式
 6. 记录历史运行结果
 7. 支持Markdown, Latex
- IPython
 1. Anaconda自带，无需单独安装
 2. Python的交互式命令行 Shell

工作环境准备

- IDE -- 没有最好的，只有**最适合自己的**（以下选一个就可以）



PyCharm社区版，部分免费，可满足不涉及web的开发，适合大多数开发者

<https://www.jetbrains.com/pycharm/download/>

Eclipse + PyDev，完全免费，适合熟悉Eclipse或Java的开发者

1. Eclipse, <https://eclipse.org/downloads/>

2. PyDev插件, <https://marketplace.eclipse.org/content/pydev-python-ide-eclipse>

Spyder，完全免费，适合熟悉Matlab的开发者

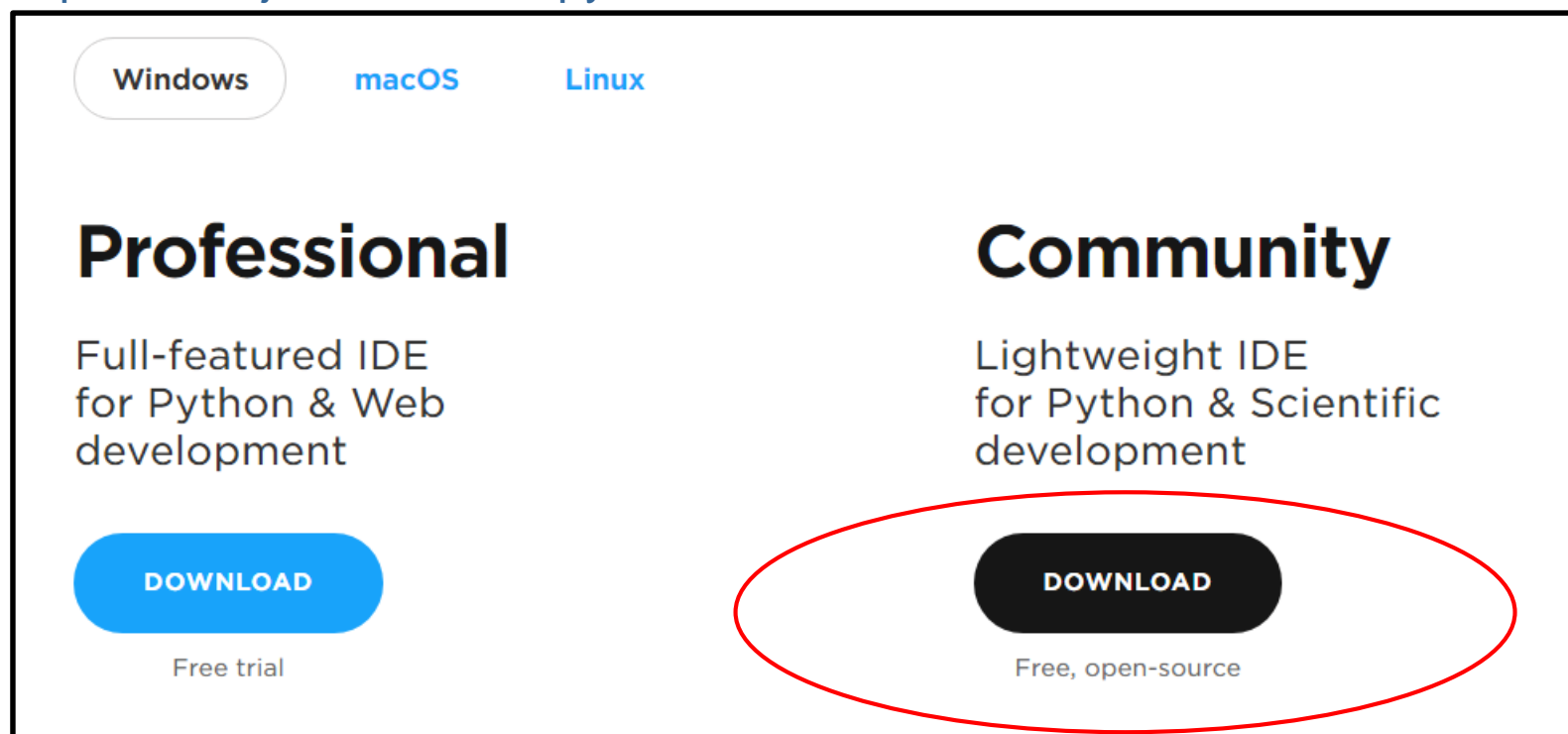
<https://github.com/spyder-ide/spyder>

工作环境准备

PyCharm配置

- 下载

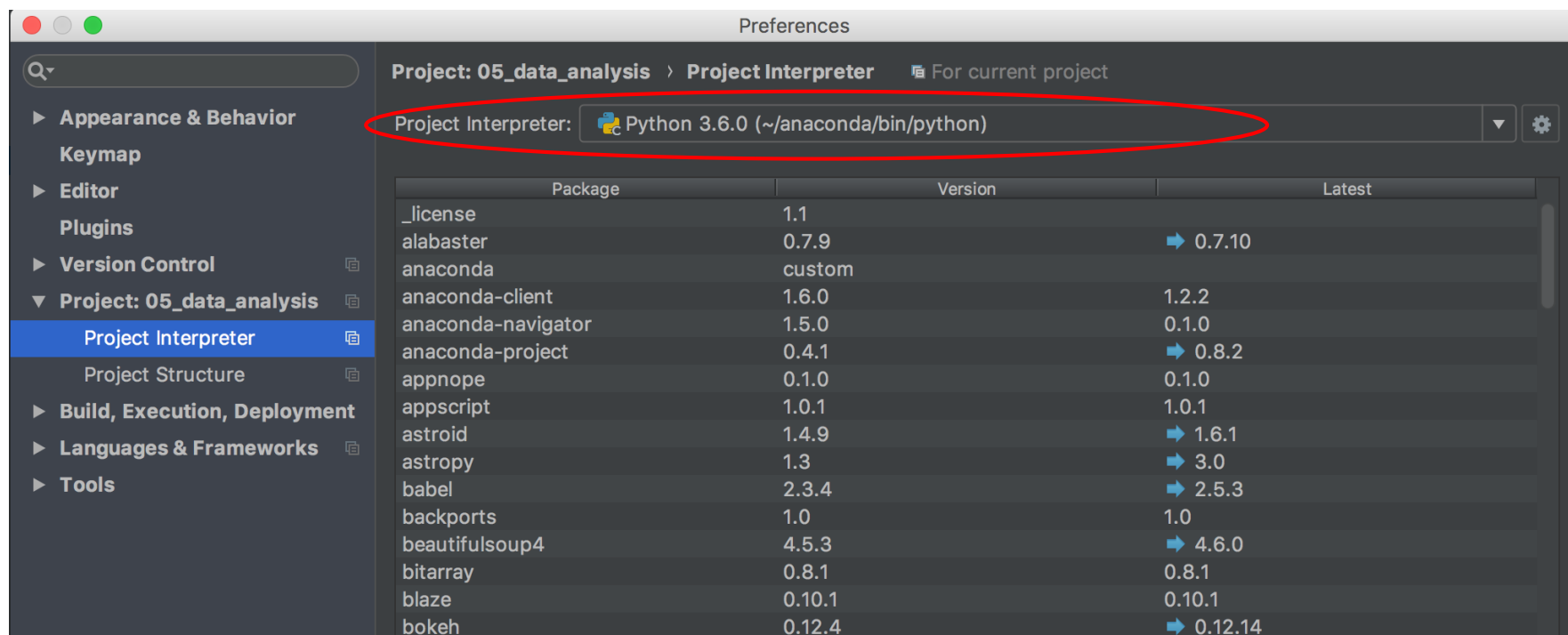
<https://www.jetbrains.com/pycharm/>



工作环境准备

PyCharm配置

- 新建项目，选择Python的安装路径



目录

- 课程介绍
- 工作环境准备
- Python进阶技巧
- 科学计算库NumPy
- 实战案例1-1：中国五大城市PM2.5数据分析（1）

Python进阶技巧

- 条件表达式

```
if x > 0:  
    y = math.log(x)  
else:  
    y = float('nan')
```



```
y = math.log(x) if x > 0 else float('nan')
```

- 列表式推导式

[exp **for** item **in** collection **if** condition]

```
l1 = []  
for i in range(1000):  
    if i % 2 == 0:  
        l1.append(i)
```



```
[i for i in range(1000) if i % 2 == 0]
```

Python进阶技巧

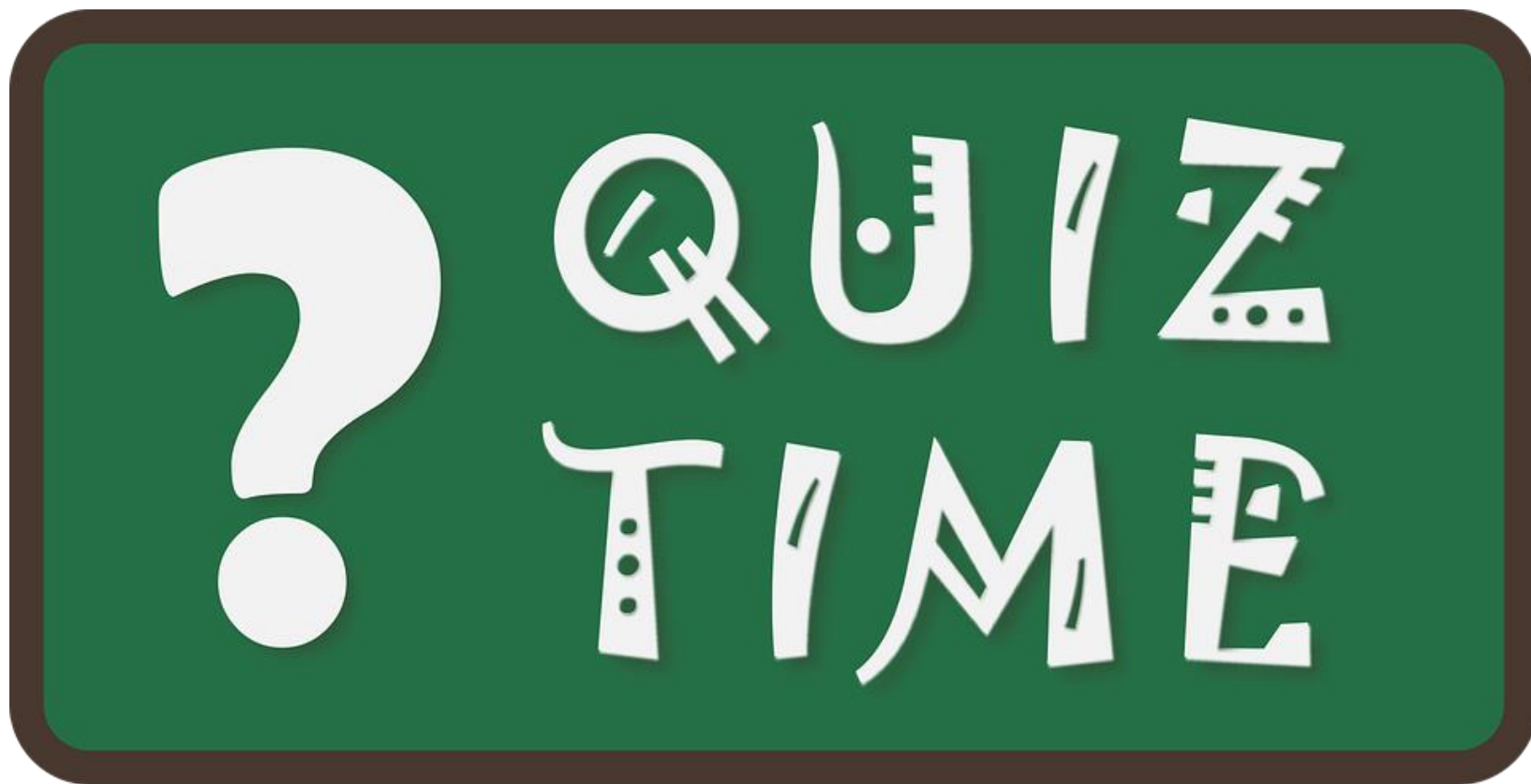
- Python常用容器类型

lect01_eg01.ipynb

List Vs Set Vs Dictionary Vs Tuple

Lists	Sets	Dictionaries	Tuples
List = [10, 12, 15]	Set = {1, 23, 34} Print(set) -> {1, 23, 24} Set = {1, 1} print(set) -> {1}	Dict = {"Ram": 26, "mary": 24}	Words = ("spam", "eggs") Or Words = "spam", "eggs"
Access: print(list[0])	Print(set). Set elements can't be indexed.	print(dict["ram"])	Print(words[0])
Can contains duplicate elements	Can't contain duplicate elements. Faster compared to Lists	Can't contain duplicate keys, but can contain duplicate values	Can contains duplicate elements. Faster compared to Lists
List[0] = 100	set.add(7)	Dict["Ram"] = 27	Words[0] = "care" -> TypeError
Mutable	Mutable	Mutable	Immutable - Values can't be changed once assigned
List = []	Set = set()	Dict = {}	Words = ()
Slicing can be done print(list[1:2]) -> [12]	Slicing: Not done.	Slicing: Not done	Slicing can also be done on tuples
<u>Usage:</u> Use lists if you have a collection of data that doesn't need random access. Use lists when you need a simple, iterable collection that is modified frequently.	<u>Usage:</u> - Membership testing and the elimination of duplicate entries. - when you need uniqueness for the elements.	<u>Usage:</u> - When you need a logical association b/w key:value pair. - when you need fast lookup for your data, based on a custom key. - when your data is being constantly modified.	<u>Usage:</u> Use tuples when your data cannot change. A tuple is used in combination with a dictionary, for example, a tuple might represent a key, because its immutable.

随堂测验



Python进阶技巧

- Counter
 - 类似于数学中的多重集
 - `import collections`
 - `update()` 更新内容，注意是做“加法”，不是“替换”
 - 访问内容[key]
 - 注意和dict的区别：如果Counter中不存在key值，返回0；而dict会报错
 - `elements()` 方法，返回所有元素
 - `most_common()` 方法，返回前n多的数据

lect01_eg01.ipynb

Python进阶技巧

- `defaultdict`
 - 在Python中如果访问字典里不存在的键，会出现`KeyError`异常。有些时候，字典中每个键都存在默认值是很方便的
 - `defaultdict`是Python内建`dict`类的一个子类，第一个参数为`default_factory`属性提供初始值，默认为`None`。它覆盖一个方法并添加一个可写实例变量。它的其他功能与`dict`相同，但会为一个不存在的键提供默认值，从而避免`KeyError`异常。
- Python `map()` 函数
 - `map(function, sequence)`
 - 可用于数据清洗

`lect01_eg01.ipynb`

Python进阶技巧

匿名函数 lambda

- 简单的函数操作
- 返回值是func类型
- 可结合map()完成数据清洗操作

Python操作CSV数据文件

- import csv
- csv.DictReader()

lect01_eg01.ipynb

目录

- 课程介绍
- 工作环境准备
- Python进阶技巧
- **科学计算库NumPy**
- 实战案例1-1：中国五大城市PM2.5数据分析（1）

科学计算库NumPy

NumPy, Numerical Python

- 高性能科学计算和数据分析的基础包，提供多维数组对象
- ndarray，多维数组（矩阵），具有矢量运算能力，快速、节省空间
- 矩阵运算，无需循环，可完成类似Matlab中的矢量运算
- 线性代数、随机数生成
- `import numpy as np`

SciPy

- 在NumPy库的基础上增加了众多的数学、科学及工程常用的库函数
- 线性代数、常微分方程求解、信号处理、图像处理、稀疏矩阵等
- `import scipy as sp`

科学计算库NumPy

ndarray, N维数组对象（矩阵）

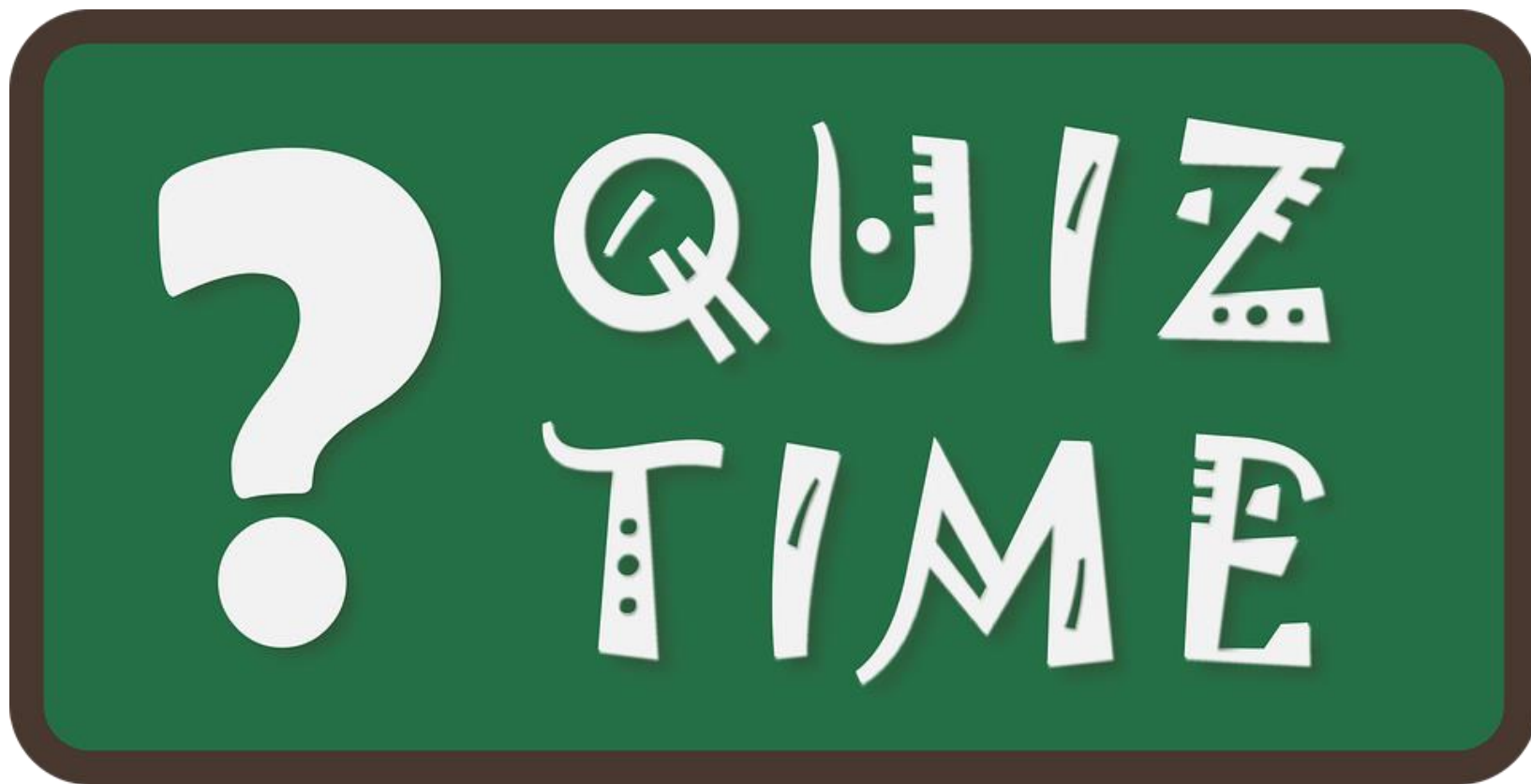
- ndim属性, 维度个数
- shape属性, 各维度大小
- dtype属性, 数据类型

lect02_eg02.ipynb

创建ndarray

- np.array(collection), collection为序列型对象(list), 嵌套序列(list of list)
- np.zeros, np.ones, np.empty 指定大小的全0或全1数组
 - 注意: 第一个参数是元组, 用来指定大小, 如(3,4)
 - empty不是总是返回全0, 有时返回的是未初始的随机值

随堂测验



科学计算库NumPy

创建ndarray (续)

- `np.arange()`类似`range()`

ndarray数据类型

- `dtype`, 类型名+位数, 如`float64`, `int32`
- 转换数组类型
 - `astype`

索引与切片

- 一维数组的索引与Python的列表索引功能相似
- 多维数组的索引
 - `arr[r1:r2, c1:c2]`
 - `arr[1,1]` 等价 `arr[1][1]`
 - `[:]` 代表某个维度的数据

`lect02_eg02.ipynb`

0,0	0,1	0,2
1,0	1,1	1,2
2,0	2,1	2,2

科学计算库NumPy

索引与切片 (续)

- 条件索引
 - **布尔值**多维数组 `arr[condition]` `condition`可以是多个条件组合
 - 注意，多个条件组合要使用 **& |**，而不是`and or`

0	1	2
3	4	5
6	7	8

T	F	F
F	T	F
F	F	T

0

4

8

[lect02_eg02.ipynb](#)

科学计算库NumPy

转置

- `arr.transpose()` 或 `arr.T`

数据叠加

- `vstack()`, `hstack()`

常用的统计方法

- `arr.mean()`, `arr.sum()`,
- `arr.max()`, `arr.min()`
- `arr.std()`, `arr.var()`
- `arr.argmax()`, `arr.argmin()`
- `arr.cumsum()`, `arr.cumprod()`
- 注意多维的话要**指定统计的维度**, 否则默认是全部维度上做统计

[lect02_eg02.ipynb](#)

科学计算库NumPy

array的拷贝操作

- `arr1 = arr2`
- `arr1`内数据的更改会影响`arr2`
- 建议使用`arr1 = arr2.copy()`

`arr.all()` 和 `arr.any()`

- `all()`, 全部满足条件
- `any()`, 至少有一个元素满足条件

`arr.unique()`

- 找到唯一值并返回排序结果

`lect02_eg02.ipynb`

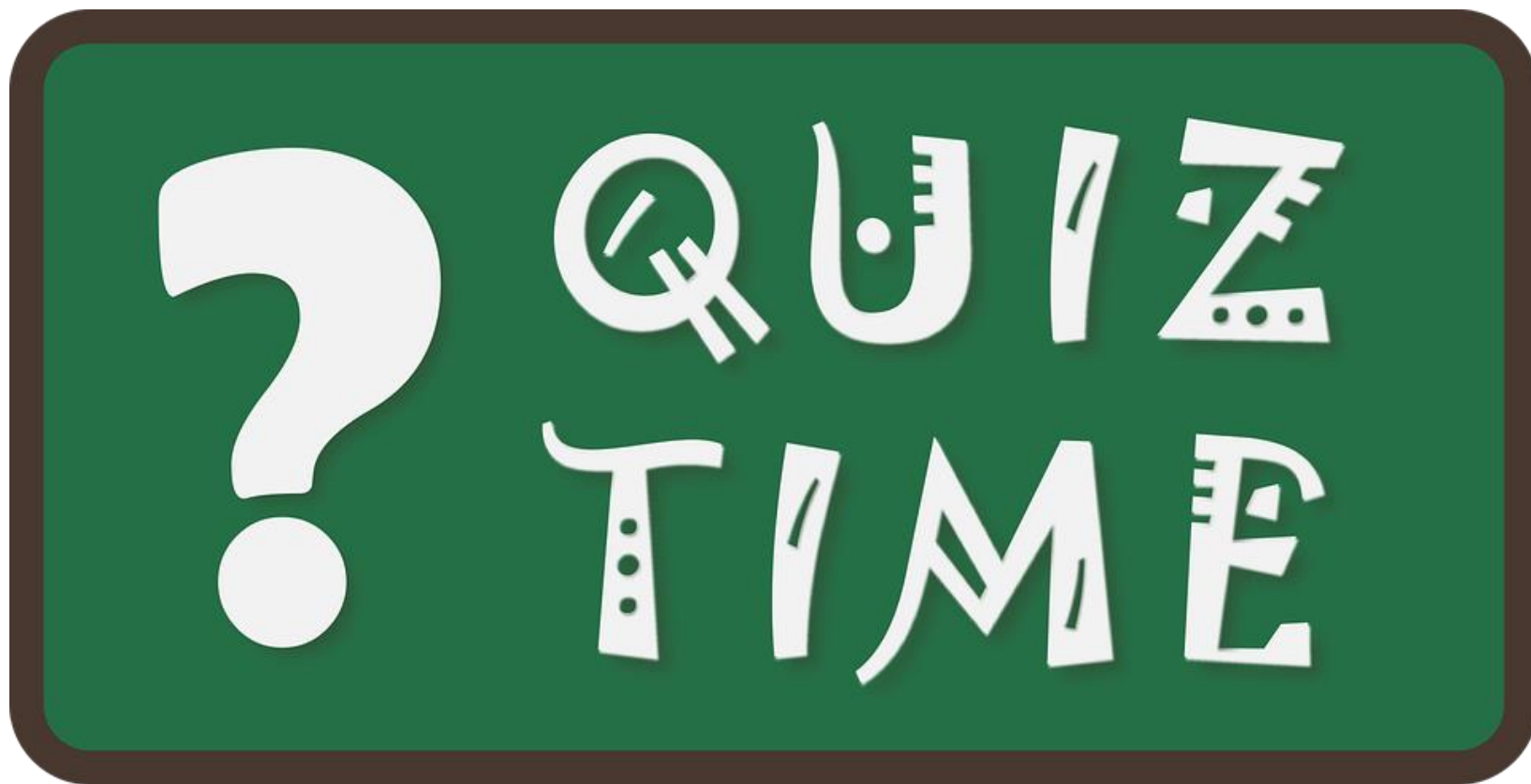
科学计算库NumPy

向量化 (vectorization)

- 获得执行速度更快、更紧凑的代码策略
- 基本思路：“一次”在一个复杂对象上进行操作，或者向其应用某个函数，而不是通过在对象的单个元素上循环来进行
- 在Python级别上，函数式编程工具map，filter和reduce提供了向量化的手段
- 在NumPy级别上，在ndarray对象上进行的循环由经过高度优化的代码负责，大部分代码用C语言编写，远快于纯Python
- 矢量间运算，相同大小的数组间的运算应用在元素上
- 矢量和标量运算，“广播” — 将标量“广播”到各个元素

lect02_eg02.ipynb

随堂测验



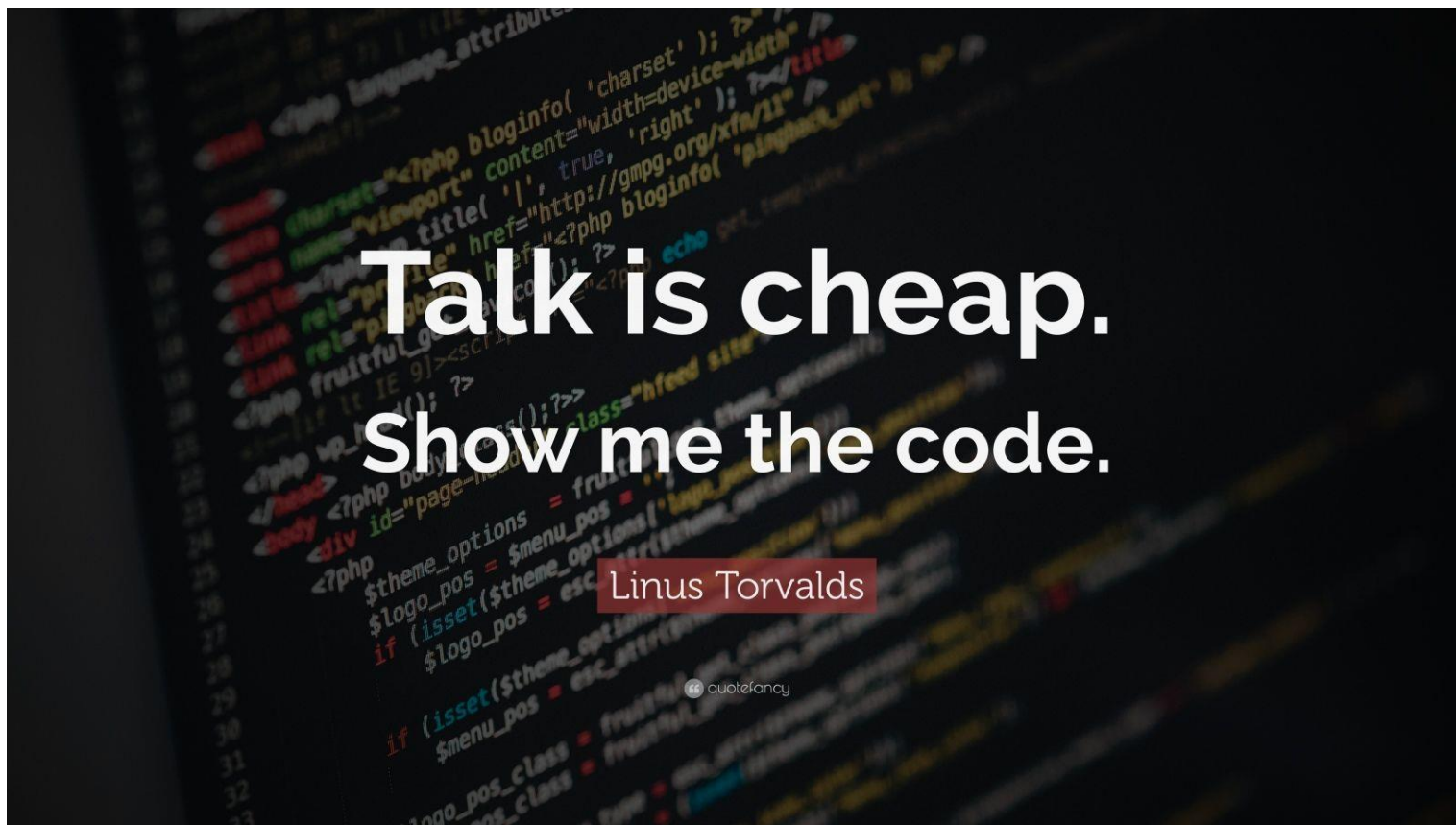
目录

- 课程介绍
- 工作环境准备
- Python进阶技巧
- 科学计算库NumPy
- 实战案例1-1：中国五大城市PM2.5数据分析（1）

实战案例 1-1

项目名称：中国五大城市PM2.5数据分析（1）

- 请参考相应的配套代码及案例讲解文档



疑问

□ 问题答疑：<http://www.xxwenda.com/>

■ 可邀请老师或者其他人回复问题

小象问答邀请 @Robin_TY 回答问题



联系我们

小象学院：互联网新技术在线教育领航者

— 微信公众号：**小象学院**

