# Data Science Tools

Dr. Sethavidh Gertphol

# Outline

* IPython and Jupyter
* NumPy and SciPy
* Pandas
* scikit-learn
* Matplotlib and Seaborn
* Anaconda
* R

# IPython

* a shell for Python interactive programming
* interfaces with OS
  * file system
  * executing another program
* inline plot and other GUI
* a kernel for Jupyter

# spyder is an IDE for IPython

Spyder (Python 3.6)

/Users/akepooh/Google Drive KU/Documents/Classes/DataScience/DataSets

Editor - /Users/akepooh/Google Drive KU/Documents/Classes/DataScience/DataSets/practice1.py

practice1.py

scripts/code area

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Aug 31 22:46:06 2018

@author: akepooh
"""

import pandas as pd

medals = pd.read_csv('olympics.csv')
len(medals)
len(medals.columns)
cols = ['Country', 'SGames','SGold','SSilver','SBronze','STotal', 'WGames',
        'WGold','WSilver','WBronze','WTotal','AllGames', 'AllGold','AllSilver',
        'AllBronze','AllTotal']
medals.columns = cols
col_total = medals.iloc[146]
medals = medals[:146]
medals['Code'] = medals.Country.str.split('(').str.get(1).str.split(')').str.get(0)
medals['Country'] = medals.Country.str.split('(').str.get(0).str.strip()
medals.Code.is_unique
medals = medals.set_index('Code', verify_integrity=True)
medals.SGames.max()
medals.Country[medals.SGames == medals.SGames.max()]
medals.Country[medals.WGames == medals.WGames.max()].size
c = ['THA','SCG','INA','PHI','VIE','MAS']
medals.loc[c,['STotal','WTotal']]
medals.WTotal.sum()
medals.loc[medals.Country.str.startswith('A'),'Country':'STotal']
medals.loc[medals.WTotal > medals.STotal, ['Country','STotal','WTotal']]
medals[(medals.SGold > medals.SSilver) & (medals.SSilver > medals.SBronze)]
len(medals[(medals.STotal==0) | (medals.WTotal == 0)])
medals['score'] = (medals['SGold']+medals['WGold'])*3 + (
        medals.SSilver+medals.WSilver)*2 + medals.SBronze + medals.WBronze
medals['perGame'] = (medals.SGold*3+medals.SSilver*2+medals.SBronze)/medals.SGames
Wscore = (medals.WGold*3+medals.WSilver*2+medals.WBronze)/medals.WGames
Wscore = Wscore.fillna(0)
medals.perGame = medals.perGame + Wscore
top10_score = medals.sort_values('score',ascending=False)[:10]
top10perGame = medals.sort_values('perGame', ascending=False)[:10]
AllGermany = medals[medals.Country.str.contains('Germany')].sum()
AllGermany.SGames -= 5
AllGermany.WGames -= 6
AllGermany.loc['Country'] = 'All Germany'
medals = medals.drop(medals[medals.Country.str.contains('Germany')].index)
AllGermany.name='GER'
medals = medals.append(AllGermany)
medals['score'] = (medals['SGold']+medals['WGold'])*3 + (
        medals.SSilver+medals.WSilver)*2 + medals.SBronze + medals.WBronze
medals['perGame'] = (medals.SGold*3+medals.SSilver*2+medals.SBronze)/medals.SGames
Wscore = (medals.WGold*3+medals.WSilver*2+medals.WBronze)/medals.WGames
```

Variable explorer

| Name | Type | Size | Value |
|---|---|---|---|
| col_total | Series | (16,) | Series object of pandas.core.series module |
| cols | list | 16 | ['Country', 'SGames', 'SGold', 'SSilver', 'SBronze', 'STotal', 'WGames ... |
| medals | DataFrame | (147, 16) | Column names: Country, SGames, SGold, SSilver, SBronze, STotal, WGames ... |

variables list

Variable explorer | File explorer | Help | Profiler | Static code analysis

IPython console

Console 1/A

interactive shell

```
In [3]: len(medals)
Out[3]: 147

In [4]: len(medals.columns)
Out[4]: 16

In [5]: cols = ['Country', 'SGames','SGold','SSilver','SBronze','STotal', 'WGames',
  File "<ipython-input-5-99a77ecafbce>", line 1
    cols = ['Country', 'SGames','SGold','SSilver','SBronze','STotal', 'WGames',
                                                                             ^
SyntaxError: unexpected EOF while parsing


In [6]:

In [6]: cols = ['Country', 'SGames','SGold','SSilver','SBronze','STotal', 'WGames',
   ...:         'WGold','WSilver','WBronze','WTotal','AllGames',
'AllGold','AllSilver',
   ...:         'AllBronze','AllTotal']
   ...:

In [7]: medals.columns = cols

In [8]: col_total = medals.iloc[146]

In [9]:
```

IPython console | History log

Permissions: RW | End-of-lines: LF | Encoding: UTF-8 | Line: 17 | Column: 22 | Memory: 65 %

# Jupyter Notebook

* web application that creates and shares documents that
    * has live code
    * has narrative text (in Markdown)
    * has visualization
* can view, run, and share codes through the web
* was part of IPython, but spinned out

Prepare KU ds camp training o | Google Drive KU/Documents/C | 04-02-AggGroup | 07-01-LinearRegression

localhost:8888/notebooks/Google%20Drive%20KU/Documents/Classes/DataScience/Slides/04-02-AggGrou...

Apps | KU Work | Asst Prof | KU Research | Arduino | Big Data | Kid | Full-Stack | High-Resolution a... | ทัวร์ชมฟาร์ม ไฮโดรโ... | archive

jupyter 04-02-AggGroup Last Checkpoint: 09/13/2018 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run | Markdown

web-based interface

markdown (text/bullets)

## Pivot Table

Pivot Table is used to create a new table where cells are summarized from another table.

The row (index) and column of the pivot table can be chosen from columns of a another table.

aggfunc parameter is the name of function to summarized data

https://nikgrozev.com/2015/07/01/reshaping-in-pandas-pivot-pivot-table-stack-and-unstack-explained-with-pictures/

inserted picture

| ix | Item | CType | USD | EU |
|---|---|---|---|---|
| 0 | Item0 | Gold | 1 | 1 |
| 1 | Item0 | Bronze | 2 | 2 |
| 2 | Item0 | Gold | 3 | 3 |
| 3 | Item1 | Silver | 4 | 4 |

| ix=Item | Bronze | Gold | Silver |
|---|---|---|---|
| Item0 | 2 | 2 = mean(1,3) | NaN |
| Item1 | NaN | NaN | 4 |

d.pivot_table(index='Item', columns='CType', values='USD', aggfunc=np.mean)

code

```
In [8]: pt = players.pivot_table(index='team',columns='position',values='passes',aggfunc='mean').head(5)
        pt
```

result

Out[8]:

| position | defender | forward | goalkeeper | midfielder |
|---|---|---|---|---|
| team | | | | |
| Algeria | 102.000000 | 10.000000 | 23.5 | 68.222222 |
| Argentina | 138.166667 | 112.666667 | 47.0 | 151.714286 |
| Australia | 78.750000 | 30.000000 | 51.0 | 63.090909 |
| Brazil | 190.000000 | 73.000000 | 69.0 | 111.750000 |
| Cameroon | 77.375000 | 41.000000 | 54.0 | 77.666667 |

# NumPy

Numeric

* NumPy is a python library for scientific computing

lav ← nd array

* add supports for **large multi-dimensional array object**

ข้อมูลต่อประเภทเดียวกัน (ปังดับ)

* includes operations like sorting array, linear algebra and Fourier transform

* basis for other scientific packages

# SciPy

Science

→ ต่อยอด numpy ก่อน
(มีอะไรเพิ่มจาก numpy)

* provides many more numerical operations on NumPy array such as
  * linear programming, optimization
  * linear algebra
  * statistics
  * signal processing
  * etc.
* used for scientific computing and much more

# Pandas

* provides high-performance **data structure and data analysis**
  * Series and DataFrame objects
  * data manipulation on those objects
  * parallel operations for performance
* some compare Pandas to a command-line Excel
* basis for data science works

# scikit-learn

* Python package for **Machine Learning**
* basic ML algorithms
  * SVM, random forest, KNN classification
  * ridge/lasso regression
  * K-means, spectral clustering
* also
  * dimensionality reduction
  * model selection

# Matplotlib

* Python package for **plotting data and creating charts**
* charts can be output to IPython or Jupyter Notebook
* can be used as quick, exploratory plot
* or can be fine-tune to create publication quality charts

# Seaborn

การพล็อตกราฟ → เอา plt ไปแก้ตอนได้!

* Python library for quickly plotting charts
* Use matplotlib underneath
* has default settings and standard interface to matplotlib
* users can create chart quickly with **single call** to seaborn

# Anaconda

* a Distribution of Python and packages
  * has everything we talked about in one distro
* also a package manager that allows searching and installing of new packages
* also an environment manager that allows multiple configurations of Python to exist in the same system

Anaconda Navigator

ANACONDA NAVIGATOR

Sign in to Anaconda Cloud

Home

Environments

Learning

Community

Documentation

Developer Blog

Applications on base (root) Channels Refresh

**JupyterLab** 0.35.3
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
Launch

**Notebook** 5.7.2
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Launch

**Qt Console** 4.3.1
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
Launch

**Spyder** 3.3.2
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Launch

**Glueviz** 0.13.3
Multidimensional data visualization across files. Explore relationships within and among related datasets.
Install

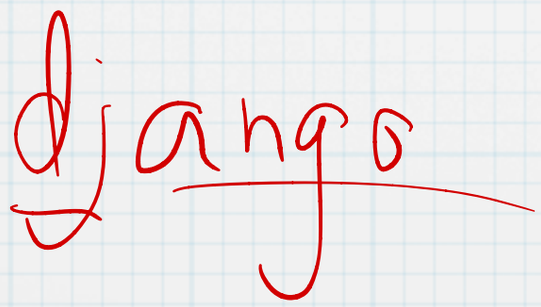
**Orange 3** 3.17.0
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.
Install

**RStudio** 1.1.456
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.
Install

**VS Code** 1.30.1
Streamlined code editor with support for development operations like debugging, task running and version control.
Install

# R

* a language for statistical computing
* has GUI frontend (RStudio)
* package management (CRAN)
* has huge package selections



# Summary

Data Gathering

Data Cleaning

Data Transform

Data Visualization

Data Modeling

Python

Pandas

Matplotlib

scipy/ scikit-learn

R / ggplot2