

2018 COMP90008

Workshop 1

Your Tutor

Winn Chow

winn.chow1@unimelb.edu.au

Github: <https://github.com/winnchow/2018-COMP20008>

Agenda

Python programming:

Run Anaconda 3 > Jupyter Notebook

Pandas

- Series
- DataFrames
- Groupby

Discussion questions (if time allows)

Jupyter Notebook Python 3

1. Anaconda 3 -> Anaconda Prompt
 - `cd <your working directory>`
 - `jupyter notebook`
2. Anaconda 3 -> Jupyter Notebook

Python Data Analysis Library (Pandas)

Official Website

- <https://pandas.pydata.org/>

Documentation

- <http://pandas.pydata.org/pandas-docs/stable/>

API reference

- <http://pandas.pydata.org/pandas-docs/stable/api.html>

10 Minutes to pandas

- <http://pandas.pydata.org/pandas-docs/stable/10min.html#min>

Pandas Cheat Sheet

- https://s3.amazonaws.com/assets.datacamp.com/blog_assets/PandasPythonForDataScience.pdf

Series

A array of values with labels (called index)

```
# defining the Series name  
co2_Emission.name = 'CO2 Emission'
```

```
# defining the name of the index  
co2_Emission.index.name = 'Year'
```

```
co2_Emission.values
```

```
array([15.45288167, 17.20060983, 17.86526004, 18.16087566, 18.20018196,  
       16.92095367, 16.86260095, 16.51938578, 16.34730205])
```

```
# verify the series object  
co2_Emission
```

	Year	Values
	1990	15.452882
	2000	17.200610
	2007	17.865260
	2008	18.160876
	2009	18.200182
	2010	16.920954
	2011	16.862601
	2012	16.519386
	2013	16.347302

Name: CO2 Emission dtype: float64

Name

Selection

.loc is primarily label based

.iloc is primarily integer
position based

```
# create a new series of the population
Aus_Population = {'1990':17065100, '2000':19153000, '2007':20827600,
                  '2008':21249200, '2009':21691700, '2010':22031750,
                  '2011':22340024, '2012':22728254, '2013':23117353}
population = pd.Series(Aus_Population)
```

```
# both the start and the stop are included
population.loc['1990':'2000']
```

```
1990    17065100
2000    19153000
dtype: int64
```

```
# the start is included but the stop is not
population.iloc[0:2]
```

```
1990    17065100
2000    19153000
dtype: int64
```

DataFrame

Two-dimensional
tabular data
structure contains
an ordered
collection of
columns

```
# create a DataFrame from a csv file
countries = pd.read_csv('countries.csv', encoding = 'ISO-8859-1')
```

```
countries.columns
```

```
Index(['Country', 'Region', 'IncomeGroup'], dtype='object')
```

```
countries.index
```

```
RangeIndex(start=0, stop=217, step=1)
```

```
# check the top 10 countries in the DataFrame
countries.head(10) # the default value is set to 5
```

Index

	Country	Region	IncomeGroup
0	Afghanistan	South Asia	Low income
1	Albania	Europe & Central Asia	Upper middle income
2	Algeria	Middle East & North Africa	Upper middle income
3	American Samoa	East Asia & Pacific	Upper middle income
4	Andorra	Europe & Central Asia	High income
5	Angola	Sub-Saharan Africa	Upper middle income
6	Antigua and Barbuda	Latin America & Caribbean	High income
7	Argentina	Latin America & Caribbean	Upper middle income
8	Armenia	Europe & Central Asia	Lower middle income
9	Aruba	Latin America & Caribbean	High income

Column
Name

Column

Reference

Python for data analysis [electronic resource] / Wes McKinney (the creator of Pandas)

- <https://eds-b-ebscohost-com.ezp.lib.unimelb.edu.au/eds/detail/detail?vid=4&sid=149fdfdf-6ea8-49ae-b82f-11dc053d9891%40pdc-v-sessmgr05&bdata=JnNpdGU9ZWZlWxpdmUmc2NvcGU9c2l0ZQ%3d%3d#AN=melb.b6087937&db=cat00006a>

Data Wrangling

Data Science

- https://cdn-images-1.medium.com/max/1000/1*mgXvzNcwfpnBawl6XTkVRg.png

Examples of Dirty Data

- <https://www.youtube.com/watch?v=Z7ffLdRsftg>

What is Big Data?

- https://www.sas.com/en_au/insights/big-data/what-is-big-data.html