

# Python Data Analysis and Plotting

José Antonio Perdiguero López

Málaga Python: PyDay 2016

September 14, 2016

## Introduction

What is Data Analysis?

Why Python?

Python Data Analysis Stack

## Pandas

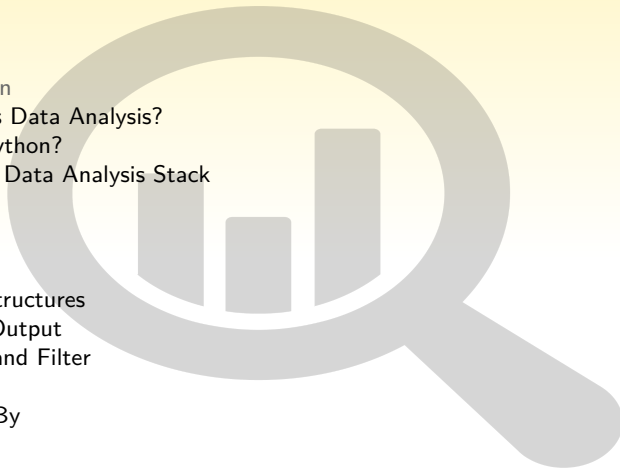
Data Structures

Input/Output

Select and Filter

Merge

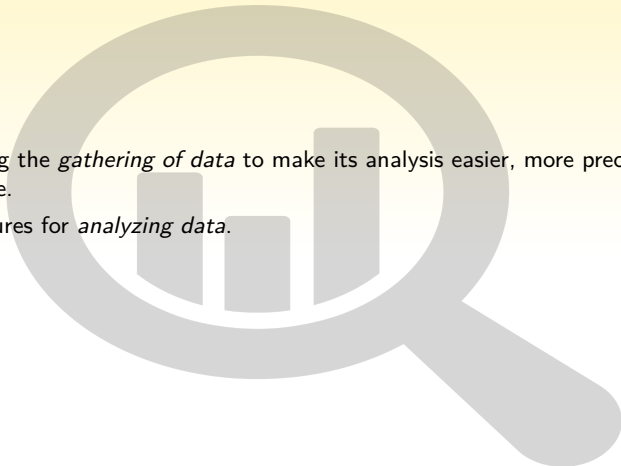
Group By



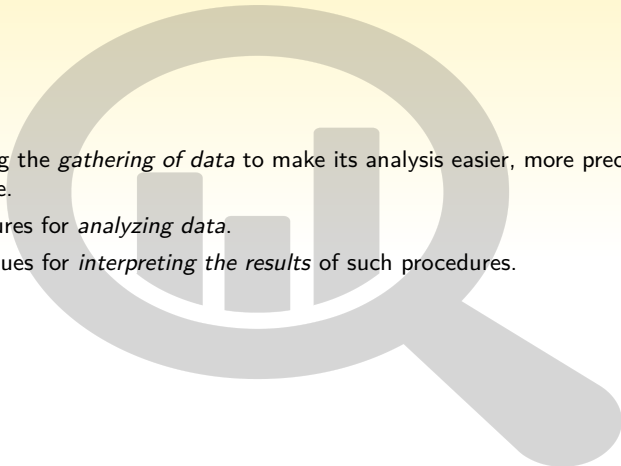
# What is Data Analysis?

- Planning the *gathering of data* to make its analysis easier, more precise or more accurate.

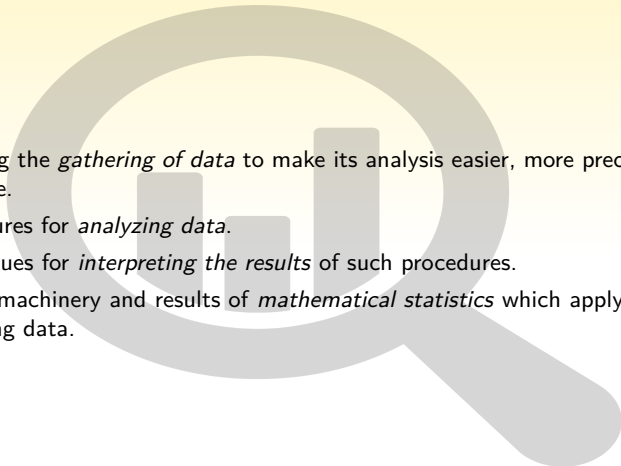
# What is Data Analysis?

- 
- Planning the *gathering of data* to make its analysis easier, more precise or more accurate.
  - Procedures for *analyzing data*.

# What is Data Analysis?

- 
- Planning the *gathering of data* to make its analysis easier, more precise or more accurate.
  - Procedures for *analyzing data*.
  - Techniques for *interpreting the results* of such procedures.

# What is Data Analysis?

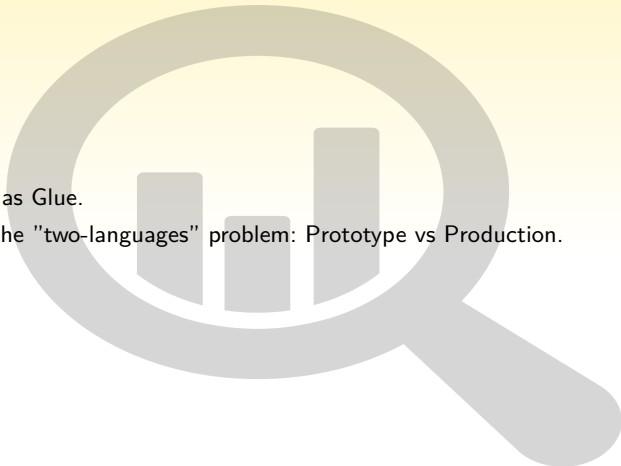
- 
- Planning the *gathering of data* to make its analysis easier, more precise or more accurate.
  - Procedures for *analyzing data*.
  - Techniques for *interpreting the results* of such procedures.
  - All the machinery and results of *mathematical statistics* which apply to analyzing data.

# Why Python?

- Python as Glue.

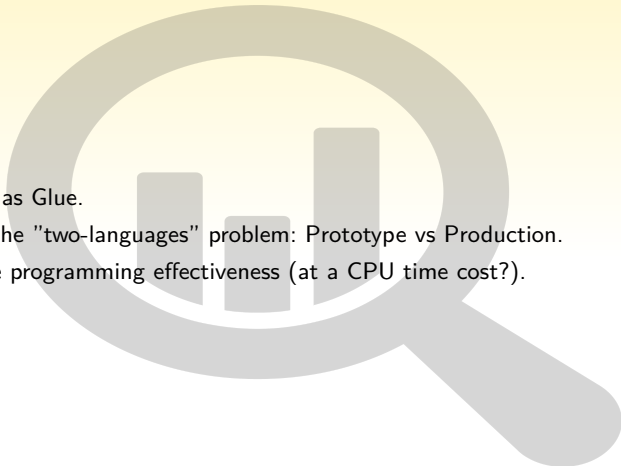


# Why Python?

- 
- Python as Glue.
  - Solves the "two-languages" problem: Prototype vs Production.



# Why Python?


- 
- Python as Glue.
  - Solves the "two-languages" problem: Prototype vs Production.
  - Increase programming effectiveness (at a CPU time cost?).

# Python Data Analysis Stack

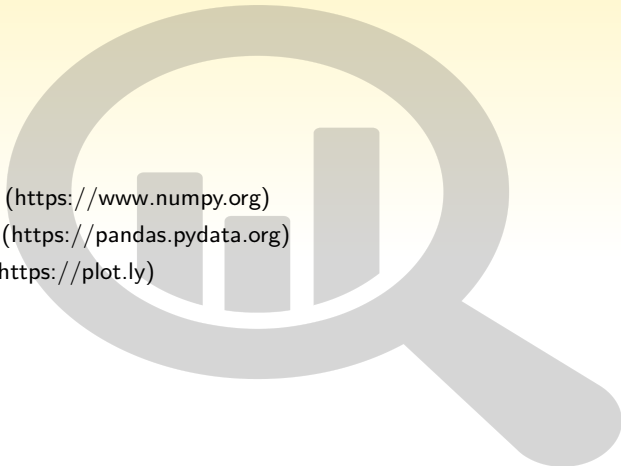
- NumPy (<https://www.numpy.org>)



# Python Data Analysis Stack

- 
- NumPy (<https://www.numpy.org>)
  - Pandas (<https://pandas.pydata.org>)

# Python Data Analysis Stack

- 
- NumPy (<https://www.numpy.org>)
  - Pandas (<https://pandas.pydata.org>)
  - Plotly (<https://plot.ly>)

# Data Structures

## Series

One-dimensional labeled array capable of holding any data type.

## DataFrame

Two-dimensional labeled data structure with columns of potentially different types.

# Input

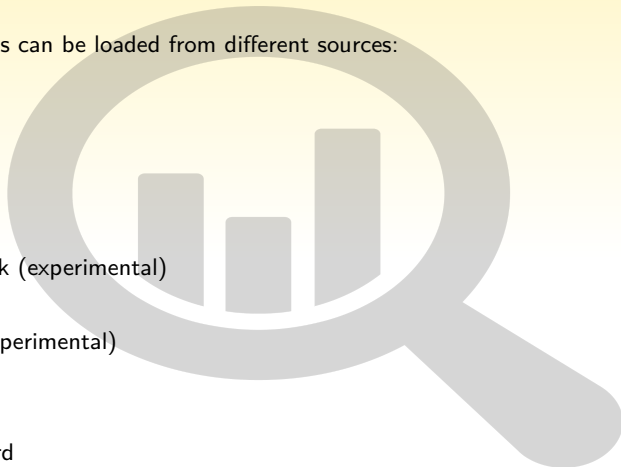
Pandas DataFrame accepts many different kinds of input:

- Dict of 1D ndarrays, lists, dicts, or Series.
- 2-D numpy.ndarray.
- Structured or record ndarray.
- A Series.
- Another DataFrame.

# Read

DataFrames can be loaded from different sources:

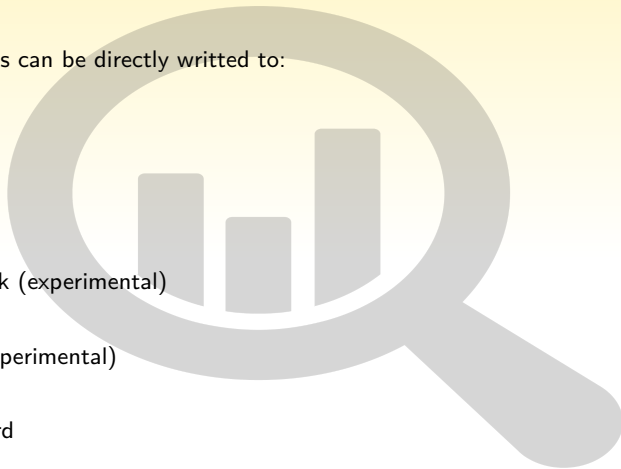
- csv
- excel
- hdf
- sql
- json
- msgpack (experimental)
- html
- gbq (experimental)
- stata
- sas
- clipboard
- pickle



# Write

DataFrames can be directly writted to:

- csv
- excel
- hdf
- sql
- json
- msgpack (experimental)
- html
- gbq (experimental)
- stata
- clipboard
- pickle





# Select and Filter

## Select

Select rows, columns or cells using python indexing notation:

```
# Get rows from 5 to 10
df.ix[5:10]

# Get column Foo
df.ix[:, 'Foo']

# Get rows 1, 3 and 7; and columns Foo and Bar
df.ix[[1, 3, 7], ['Foo', 'Bar']]
```

## Filter

Apply filters to DataFrames using python expressions:

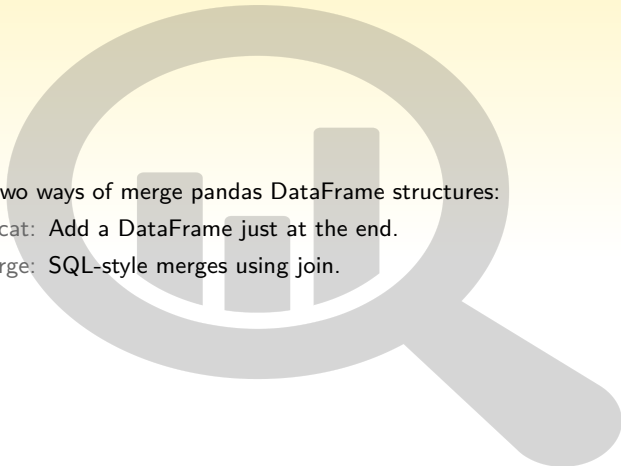
```
# Get rows whose value of Foo column is positive
df[df['Foo'] > 0]
```

## Select and Filter

Select rows using filters and index:

```
# Get Bar column for those rows whose value of Foo column is positive
df.ix[df['Foo'] > 0, 'Bar']
```

# Merge Structures



There are two ways of merge pandas DataFrame structures:

Concat: Add a DataFrame just at the end.

Merge: SQL-style merges using join.

# Group By

The process of *Group By* involves the steps:

- Split the data into groups based on some criteria.
- Apply a function to each group independently.
- Combine the results into a data structure.