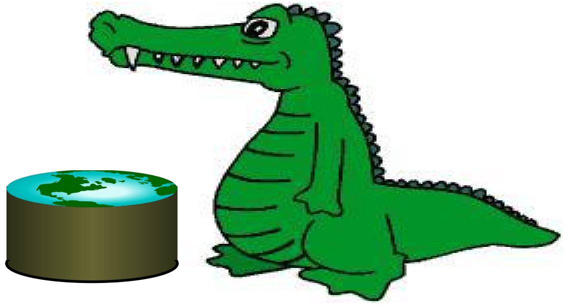
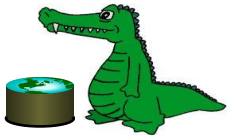
Exploratory Data Analysis

(Python, Pandas & matplotlib)

Xiaofeng Zhou





Goal

Learn how to use more advanced data processing tools

- Python
- Pandas: Data Processing
- Matplotlib: Visualization

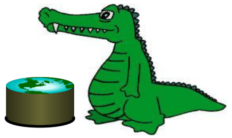
Setup

Go to course web site, find the ipython notebook file, put it in your VM and use Ipython notebook to view it.

Pandas

Main data structures:

- Series: one-dimensional collections of any data type.
- DataFrames: two-dimensional data structures similar to a database table.



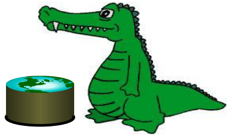
Pandas

- Import libraries

```
import pylab
import pandas as pd
```

- Create DataFrame

```
df = pd.DataFrame({
    'a': [1, 2, 3, 4],
    'b': [ 'w', 'x', 'y', 'z'] })
```



The Basics - explore

- Detailed information about schema

```
df.info()
```

- Check first / last few rows

```
df.head(n)

df.tail(n)
```

- Any range

```
df[1:3]
```

The Basics - describe

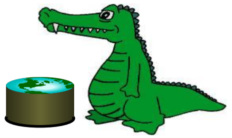
- df.describe()

	a
count	4
mean	2.5
std	1.29
min	1
max	4

Import Dataset

- Dataset can be downloaded from link in the Ipython Notebook

```
log_df = pd.read_csv(
    # Path
    "/home/datascience/wc_day6_1_sample.csv",
    # Column Headers
    names=[“ClientID”,”Date”,”Time”,”URL”, ”ResponseCode”,“Size”],
    # Non-Value
    na_values=['-'])
```



Row & Column Filtering

- Row filters (selection from RA)

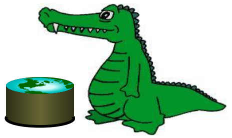
```
is_may1st = log_df['Date'] == '01/May/1998’'
may1_df = log_df[is_may1st]
```

Or

```
may1_df = log_df[log_df['Date'] == '01/May/1998']
```

- Column filters (selection from RA)

```
url_codes = log_df[['URL', 'ResponseCode']]
```



Grouping

- Form groups

```
grouped = log_df.groupby('ResponseCode')

grouped.groups.keys()

grouped.get_group(200)
```

- Resturns a **DataFrameGroupBy** object

  -- Much like a dictionary: Keys are grouping values that maps to a DataFrame with all objects in that group

- Operations for each group

```
grouped.describe()
grouped.size()
grouped.sum(), grouped.mean(), grouped.median()
```

Visualization - Pie Chart

- Show the percentage of each ResponseCode in a Pie Chart:

```
%matplotlib inline
# show the percentage of each response code
import matplotlib.pyplot as plt
grouped.size().plot(kind='pie', legend=True)
```

Visualization - Bar Chart

- Show the percentage of each ResponseCode in a Bar Chart:

```
grouped.size().plot(kind='bar')
```

Visualization - Line Plot

- Show the # request over each hour of the day

```
ax = hour_grouped.size().plot()
ax.set_ylabel("# Requests")
ax.set_xlabel("Hour of the day")
ax.set_title("# Request changes in a day")
```

Visualization - Two Line Plot

- Show the # request & size of traffic over each hour of the day

```
fig, ax1 = plt.subplots()
ax2 = ax1.twinx()
x = hour_grouped.size().index

ax1.plot(x, hour_grouped.size(), 'g-')
ax2.plot(x, hour_grouped['Size'].sum(), 'r-')

ax1.set_xlabel('Hour of the day')
ax1.set_ylabel('# Requests', color='g')
ax2.set_ylabel('Size of traffic handled', color='r')
```

Q&A

- Lab 2 In-Class Quiz.
- Homework.