Class 12 - Pandas

[w200] MIDS Python Course Summer 2018

Agenda

Course Schedule

Project 2 - Project Team Coordination (10 minutes)

Data Exploration and Analysis

Project 2 - Team Discussions (as time allows)

Schedule

Class 10 - Working with Text and Binary Data

Class 11 - NumPy

Class 12 - Data Analysis with Pandas

Class 13 - More Data Analysis with Pandas

Class 14 - Group Work, Code Testing and Final Project Showcase

https://docs.google.com/spreadsheets/d/11DxadnNwyFaJIPYLUJSPUlNGCtTenBCR4yaR1CbFBKg

Schedule | Where we're going - projects/exams

Live Session 11 - Discuss Final Project

Live Session 12 - Proposal Finalized

Live Session 13 - Final Exam Distributed

Live Session 14 - Final Exam Due, Projects Due, Final Project Showcase

https://docs.google.com/spreadsheets/d/11DxadnNwyFaJIPYLUJSPUlNGCtTenBCR4yaR1CbFBKg

Schedule

HW assignment

You have a fun Pandas assignment this week!

Assignment Review | Numpy

How did the homework go this week?

What was the hardest? What was easiest?

Agenda

Course Schedule

Project 2 - Project Team Coordination (10 minutes)

Data Exploration and Analysis

Project 2 - Team Discussions (as time allows)

Project 2 | Grading

1-2 page proposal: 10%
8 page paper: 60%
Final class presentation: 30%

Confidential peer review and task performance itemization

Instructions:
https://github.com/MIDS-INFO-W18/assignments_upstream_summer18/tree/master/project_2

Teams:
https://docs.google.com/spreadsheets/d/1MjMaUMcLagaP4XAfAE4qpnzB7OSLjGK8uULpJF2Lhsw

Project Breakouts

Talk with your team about your project

1. Create your team repository!
2. What do you hope to get out of the project?
3. When would you like to meet next?
4. Discussion

Agenda

Course Schedule

Project 2 - Project Team Coordination (10 minutes)

Data Exploration and Analysis

Project 2 - Team Discussions (as time allows)

Data Analysis | Two Thoughts

Data Exploration Basics

Data Analysis Basics

Data Analysis | Exploration vs. Analysis

Data Exploration

- Used to ensure data integrity
- Develop questions based on the variables you have
- Try to break things

Data Analysis

- Seeks to answer a research question or hypothesis
- May involve complex math, modeling, statistics
- More likely to combine multiple dataset
- Collapse and group data in various ways

Some functions are useful for both exploration and analysis!

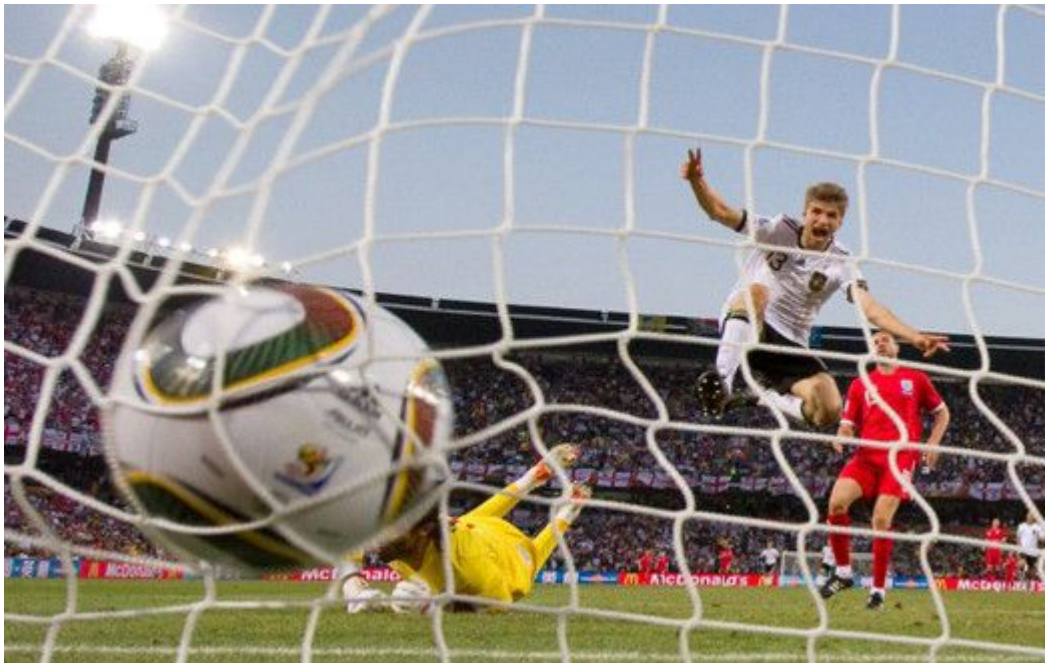
Data Analysis | Data Exploration Basics

Real World Data Is Messy.

No, really. It’s very messy.

If anything can be wrong in your data, it probably is.

Your Mindset:



Data Analysis | How to Explore?

Simple commands - but think deeply!

- value_counts()
- describe()
- max(), min(), isnull()
- basic plots,
  - histogram, scatter plot, bar chart

Data Analysis | How to validate?

1. Dataset documentation
2. Research (or even simple Google searches)
3. Cross-validation within your data
    a. E.g., if the variable “hospital_los” means “length of stay”, we should be able to compare it to variables “admit_date” and “discharge_date”
4. Cross-validation outside of your data
    a. E.g., We expect power to cost the most on the hottest days of the year (demand is highest). Let’s find the maximum power cost in our dataset, and compare it to the daily temperature to ensure it makes sense.

How much data validation is “enough” ?

Data Analysis | Exploration Demo

Demo

Pandas | Pseudocoding

How do I think about pseudocoding for data analysis?

Pandas | Analysis Design

You can think about an analysis as a **series of dataset transformations**

You might filter **out rows based on conditions**

You might **create new columns**

You might **aggregate** or **collapse by groups**

You might **join two datasets together**

Data Analysis | Analysis Breakout

Demo and Breakout

Boolean Selection

```
data = pd.Series([1,2,4,6,0,85,45,7,53,321,4,32,2355,6])

data[data < 10]

data[(data < 10) & (data > 5)]   # with keywords

data[data < 10][data > 5]        # chained
```

Using booleans as counter

```
(data > 5).sum()   # True = 1, False = 0
```

Slice and mutate | by index

slicing and manipulating

```
d5 = data.head().copy()    #  make a copy for safety

d6 = data.head(70).copy()    # specify the size of the head (ceiling e.g up to 70)

data[0:10:2]                 # standard slicing [start:end(exclusive):step]

data[0] = 10000              # value replacement

d5_6=pd.concat([d5, d6])
```

Check content | quick test for series

any and **all** tests for series

```
data[data < 10].any()          # true

data[data < 10].all()          # false

# alternative syntax

(data > 10000).any()           # false

(data > 0).all()               # false
```

Stats | quick descriptive stats for series

len(data)

data.mean(), mode(), median(), count(), std(), unique(),

data.value_counts() # super useful

data.shape # note this is an attribute not a method

Most of these are called with describe()

data.describe()

Index | lookup methods

```
Data[10] # By single index names

data[[10,20]]    # By multiple index names

# if the index is not numeric pandas interprets numbers as row number

d5.iloc[[0,3]]                                  # lookup by index location “dict style []”

data.index = range(100,len(data)+100)    # set  new index names

data.loc[[100,103]]                             # we can use .loc to call by index names

data.iget([0,3]), data.ix([0,3])            # DEPRECIATED
```

Reindexing and combining | Fill in values

```
data.index = New_index                          # overwrite the existing index

new_data = data.reindex([0,2,15,21])            # slice out rows and use their indexes,
missing values get NaN

data.reindex([0,2,15,21], fill_value=0)         # specify the missing value
```

Missing data | Fill in values

```
combo.reindex([0,2,15,21], fill_value=0)        # set fill value

new_combo.fillna(0)                             # fill in NaN values
```

forward and backward fill - guess at missing values

```
new_combo.ffill()                    # take values before

new_combo.bfill()                    # take values after

new_combo.interpolate()              # fills missing values with linear interpolation
```

Mapping methods |

```
s1=pd.Series(['1','3'])
s1=s1.astype(int)                    # why do we need this? What does it do?
s1.map(lambda x: x ** 2)             # pass a series to a lambda function
s1.map({‘1’:2,’2’:3,’3’:12})         # simple mapping with a dictionary
```

DataFrames | create and mutate

```
pd.DataFrame([upcase, lcase])                          # make a DataFrame from series

pd.DataFrame({'lowercase':lcase, 'uppercase':upcase})  # can pass col names

letters.columns = ['LowerCase','UpperCase']            # set cols explicitly

letters.index = lcase                                  # change the indexes

letters.sort('Number'), letters.sort()                 # sort by column, or by index

letters[['LowerCase','UpperCase']]                     # slice columns by name
```

DataFrames | groupby

Viewing your data by a category can yield critical insights

```
In [114]: df.groupby('Day_of_week').mean()
```

Out[114]:

	Miles	Minutes	Min_per_mile
Day_of_week			
Friday	2.786000	24.308333	7.747657
Monday	2.607143	22.243333	7.463291
Saturday	3.246429	46.708333	8.184961
Sunday	2.422727	19.762500	7.463840
Thursday	6.315000	84.530000	8.039543
Tuesday	2.428182	21.770833	7.659706
Wednesday	3.315000	28.021429	7.829348

Agenda

Course Schedule

Project 2 - Project Team Coordination (10 minutes)

Data Exploration and Analysis

Project 2 - Team Discussions (as time allows)

Project Breakouts

With the time left in the course, it is critical you start your project this week

1. Pick a time to meet next
2. Exchange contact information
3. Establish a communication plan. Email, text, both?
4. Give everyone “something to do” before you meet again.