DAT Class 8

Horizontal, Directional Data Extraction

Numpy & Pandas: An Introduction

Overview

Give students a brief but coherent overview of Numpy & Pandas, plus the role they serve in the python ecosystem

Important Note

Be prepared to handle issues with installation here.

Learning Objectives

In this lesson, students will:

- Get their Jupyter notebook up and running
- Introduce students to IPython computing environment
- Introduce students to numpy arrays & pandas dataframes

Duration

45 min - 1 hour

Suggested Agenda

Time	Activity	Purpose
0:00 - 0:15	**Anaconda Download + Installation**	Getting everyone's install correct, introduction to Jupyter notebook
0:16 - 0:30	**Numpy Introduction**	Discuss students the important details of what makes numpy unique, its role in data science ecosystem
0:31 - 0:45	**Pandas Introduction**	What it does, how it interacts with numpy, what dataframes are.
0:46 - 0:50	**Summary**	Finish this portion of the lesson by having students

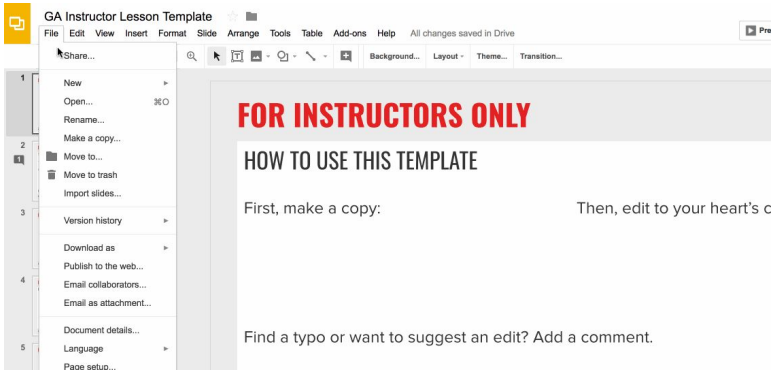
Materials and Preparation

- Time spent on this lesson might be highly variable depending on how difficult a time people have with getting their operating environment up and running.
- Besides getting everyone's Jupyter Notebook up and running, focus on making sure everyone has a broad conception about what makes numpy & pandas unique, and how they interact with one another
- Wrap this lesson up by using pd.read_csv() to load in a dataframe

Preparation (Continued)

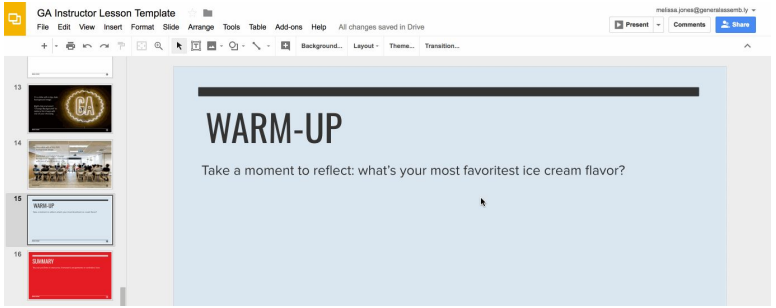
Ready to customize this lesson?

First things first, make a copy of this presentation. You'll then be able to edit to your 💗's content.



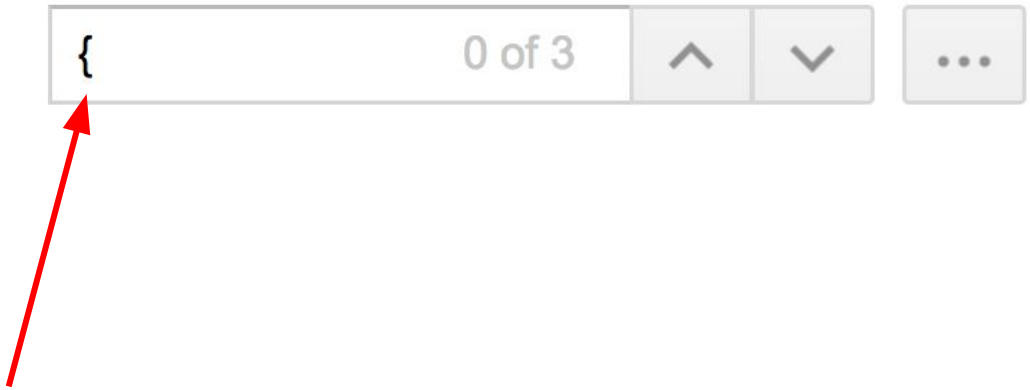
Find a typo or want to suggest an edit?

Add a comment. Our **Product Advisory Board** will review and incorporate your suggestions.



Preparation (Continued)

Remember to press Command F and do a search for instances of brackets in the deck. All instances of brackets denote places where instructors must fill in information/customize prior to presenting in class.



Differentiation and Extensions

- This is a suggestion for a way in which instructors might change up this lesson to meet the needs of more advanced or less advanced students.
- This is another suggestion.
- This is another suggestion.
- Be sure to link to any supplemental/optional materials on this slide.

Pandas & Time

Extracting data from a date can be roughly be categorized in three different ways:

- **Grouped data extraction** - summary statistics over particular months, years, etc.
- **Horizontal data extraction** - looking at internal attributes of a particular date -- day of week, business day, etc.
- **Directional data extraction** - rolling averages, momentum, etc

```
In [14]: df.iloc[:, [0,1]]
```

Out[14]:

	Cust Id	Start Date
0	90621	2015-07-01
1	90621	2015-07-01
2	48771	2015-08-01
3	114161	2015-11-01
4	87151	2016-05-01
5	121021	2016-05-01
6	23821	2016-06-01
7	62871	2016-06-01
8	83041	2016-06-01
9	64271	2016-06-01
10	62551	2016-06-01

Horizontal Data Extraction

Looking at internal characteristics of a date:

- Is it a holiday? End of the month? Evening? Morning? Etc.
- Value can be derived without relation to previous dates
- Mostly derived from the datetime attribute in Pandas

```
In [14]: df.iloc[:, [0,1]]
```

Out[14]:

	Cust Id	Start Date
0	90621	2015-07-01
1	90621	2015-07-01
2	48771	2015-08-01
3	114161	2015-11-01
4	87151	2016-05-01
5	121021	2016-05-01
6	23821	2016-06-01
7	62871	2016-06-01
8	83041	2016-06-01
9	64271	2016-06-01
10	62551	2016-06-01

Discussion: What day has highest variance of prices?

For AAPL stock, what day of the week has the highest standard deviation in trading price?

(PS - I know we're not quite doing things in the best way right now, just bear with me).

Directional Date Extraction

Looking at a value today compared to characteristics from trailing dates

- How does value today compare with values from one week ago?
- What about average values of previous two weeks?
- Useful for:
  - Comparing rates of growth
  - Demand forecasting
  - Financial analysis

```
In [14]: df.iloc[:, [0,1]]
```

Out[14]:

	Cust Id	Start Date
0	90621	2015-07-01
1	90621	2015-07-01
2	48771	2015-08-01
3	114161	2015-11-01
4	87151	2016-05-01
5	121021	2016-05-01
6	23821	2016-06-01
7	62871	2016-06-01
8	83041	2016-06-01
9	64271	2016-06-01
10	62551	2016-06-01

Directional Date Extraction

Some useful methods:

- shift() - takes values from previous observations and inserts them into today's date
  - Can be both unscaled or a fixed frequency
- diff() - takes difference in values from observed value and previous one
- pct_change() - takes percent difference from previous value

```
In [14]: df.iloc[:, [0,1]]
```

Out[14]:

	Cust Id	Start Date
0	90621	2015-07-01
1	90621	2015-07-01
2	48771	2015-08-01
3	114161	2015-11-01
4	87151	2016-05-01
5	121021	2016-05-01
6	23821	2016-06-01
7	62871	2016-06-01
8	83041	2016-06-01
9	64271	2016-06-01
10	62551	2016-06-01

Directional Date Extraction

Some useful techniques to use with this:

- Compounded returns
- Average growth rates

```
In [14]: df.iloc[:, [0,1]]
```

Out[14]:

	Cust Id	Start Date
0	90621	2015-07-01
1	90621	2015-07-01
2	48771	2015-08-01
3	114161	2015-11-01
4	87151	2016-05-01
5	121021	2016-05-01
6	23821	2016-06-01
7	62871	2016-06-01
8	83041	2016-06-01
9	64271	2016-06-01
10	62551	2016-06-01

Discussion: What week had the highest growth?

Among all stocks in our dataset, what 3 weeks had the highest average week over week growth?

Discussion: What week had the highest growth?

For the year 2019, what's been the average monthly growth for AMZN?

Directional Date Extraction

Window Statistics

- Represent how a value has changed over a particular amount of time
- Useful for capturing how unusual a particular value is at this point in time, or what sort of 'momentum' a certain metric has
- Calculations can be both static and dynamic

```
In [14]: df.iloc[:, [0,1]]
```

Out[14]:

	Cust Id	Start Date
0	90621	2015-07-01
1	90621	2015-07-01
2	48771	2015-08-01
3	114161	2015-11-01
4	87151	2016-05-01
5	121021	2016-05-01
6	23821	2016-06-01
7	62871	2016-06-01
8	83041	2016-06-01
9	64271	2016-06-01
10	62551	2016-06-01

Discussion: Rolling Correlations

If we use percent difference between itself and its rolling two week average, what date represented the best time to purchase Microsoft stock?

Directional Date Extraction

Window Statistics

Two main methods:

- rolling() - used to calculate a variety of window statistics
- ewm() - calculates a weighted moving average.....useful for capturing growth dynamics at a particular point in time.
  - More heavily weights recent events
  - Alpha paremeter determines how much you weight recent values

```
In [14]: df.iloc[:, [0,1]]
```

Out[14]:

	Cust Id	Start Date
0	90621	2015-07-01
1	90621	2015-07-01
2	48771	2015-08-01
3	114161	2015-11-01
4	87151	2016-05-01
5	121021	2016-05-01
6	23821	2016-06-01
7	62871	2016-06-01
8	83041	2016-06-01
9	64271	2016-06-01
10	62551	2016-06-01

Thank You!