

Always First: Importing Libraries

Remember that Python has only 68 built-in functions! We will rely on libraries for advanced functionality within Python:

Python:	<code>import pandas as pd</code> <code>df = pd.read_csv('file.csv')</code>
Description:	Imports the pandas library into our notebook, allowing use of <code>pd.</code> functions, and reads the CSV <code>file.csv</code> from the same directory as the notebook storing it in <code>df</code> .

Illini Football: Every game played (1892-2018)

Our next dataset explores every football game played by The Fighting Illini. Over 125+ years, the Illini have played nearly 1,300 games! Here are just a few rows/columns of this dataset:

Season	Date	Opponent	Illini\ Score	Opponent\ Score	Result	Location	Note	...
...
2018	10/6/2018	Rutgers	38	17	W	@		...
2018	10/13/2018	Purdue	7	46	L	vs.	Homecoming	...
2018	10/20/2018	Wisconsin	20	49	L	@		...
...

Multiple GroupBy

We can always GroupBy multiple columns, creating a more detailed analysis of our data.

Python:	<code>df.groupby(['Opponent', 'Result'])</code>
Description:	Returns a GroupBy object that groups data by the Opponent and then by the Result.

What happens if we swap the order?

Python:	<code>df.groupby(['Result', 'Opponent'])</code>
Description:	...what is different?

Functions

In Python we define a function that runs an algorithm that we define! The ability to create functions will greatly expand the data analysis we can do beyond the limitations of the pandas library.

Python:	<pre>def winning(row): if row['Result'] == 'W': return 1 else: return 0</pre>
Description:	

Puzzle #1: What does this function do?

Python:	<pre>def losing(row): if row['Result'] == 'L': return 1 else: return 0</pre>
Description:	

Puzzle #2: What does this function do?

Python:	<pre>def unknown(row): if row['Opponent'] == 'Indiana': return 1 if row['Opponent'] == 'Iowa': return 1 if row['Opponent'] == 'Maryland': return 1 if row['Opponent'] == 'Michigan': return 1 if row['Opponent'] == 'Michigan State': return 1 if row['Opponent'] == 'Minnesota': return 1 if row['Opponent'] == 'Nebraska': return 1 if row['Opponent'] == 'Ohio State': return 1 if row['Opponent'] == 'Penn State': return 1 if row['Opponent'] == 'Purdue': return 1 if row['Opponent'] == 'Wisconsin': return 1 return 0</pre>
Description:	

Using Functions

Within a DataFrame, we need to **apply** the function to run it across every row.

Python:	<code>df.apply(functionName, axis=1)</code>
----------------	---

- We must include **axis=1** in order to apply the function to **every row**.
 - By default (axis=0), the function is applied to every column (not usually what we will care about).
-

Puzzle #3: Describe Each Line of Python code:

Python + Description:	<pre>df['Win'] = df.apply(winning, axis=1) df['Loss'] = df.apply(losing, axis=1) group = df.groupby(by='Opponent') result = group.agg(sum) result[['Opponent', 'Win', 'Loss']]</pre>
------------------------------	--

Data Cleaning

In Lab this week, you began to see data cleaning. We have several ways Python can help us with Data Cleaning:

Python:	<code>df.unique('Result')</code>
Description:	
Output:	

Replacing Values

Python:	<code>df.replace('Orig', 'New')</code>
Description:	

Example:

Python:	<code>df.replace('T', 'L')</code>
Description:	
What Happens?	