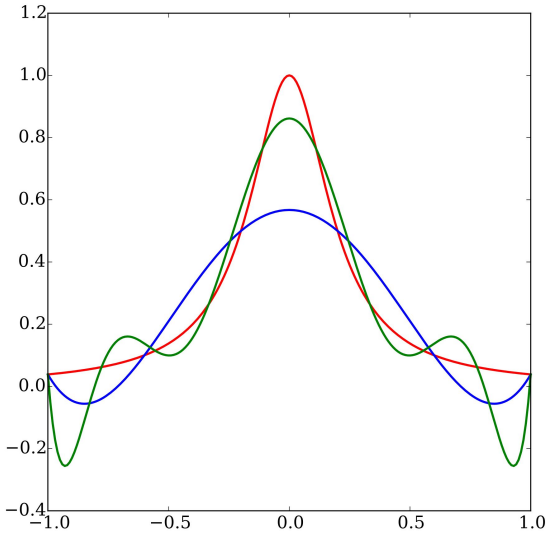
Numpy & Pandas: An Introduction

Python For Data

Python and Data: How They Work Together

- Core Python language is not good for data analysis
  - Wasn't built for it originally
  - Modern data analysis requires data structures that aren't handled by native Python
- But Python is everyone's favorite language.....
- Scipy ecosystem was built to make data-intensive computing feasible using Python



Numpy: Where It All Starts

Numpy: replacing Matlab

- Numpy is the most foundational of the libraries in the Scipy ecosystem
- Allows you to perform computations on large amounts of data at one time
- Fundamental data structure is the 2D-array

```
np.where(b > 5, 'hello', False)
```

```
array([['False', 'False', 'False',
        ['False', 'False', 'hello',
        ['hello', 'hello', 'hello',
        ['hello', 'hello', 'hello',
```


Numpy: Where It All Starts

- Built from the ground up for fast performance
- Is only Python by appearance: compiles to C
- Library is very well developed, has 500+ different methods built into it
- Is the base of other data libraries built on top of it

```
np.where(b > 5, 'hello', False)
```

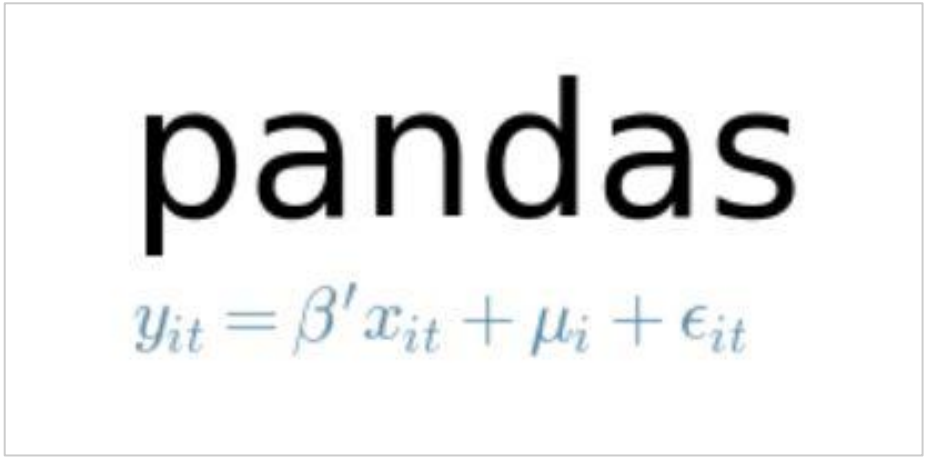
```
array([['False', 'False', 'False',
        ['False', 'False', 'hello',
        ['hello', 'hello', 'hello',
        ['hello', 'hello', 'hello',
```


Pandas

Python For Data

Pandas: An Introduction

- Primary tool for dealing with outside data
- Uses a data structure called a dataframe to do its work.
- Works like a massive Excel spreadsheet that you can dynamically program



Pandas: Connecting to Data

- Pandas connects to following:
  - CSV
  - Excel
  - SQL
  - JSON
  - STATA
  - SAS
  - HTML
  - others
- All use the same syntax
  pd.read_data_type_here()

```
df = pd.read_excel(r'C
df.head()
```

	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

Pandas: Connecting to Data

- Have the following options when reading in data:
  - What columns to include
  - What rows to include or exclude
  - Specific data types of different columns
  - Specify columns that represent time
  - How to infer time

```
df = pd.read_excel(r'C
df.head()
```

	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

Selecting & Querying Data in Pandas

Python For Data

Pandas: Selecting Data

- Can select data via rows and columns
- Can grab values both using labels and index positions
- Usually happens with two primary commands:
  - **.loc:** grab data by its labels
  - **.iloc:** grab data by its index position

```
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

Solo Exercise: Activity #2

Complete Section #1 In Your Lab

Pandas: Selecting Data

- Can use Pandas to query data similar to what you'd use in SQL

- Allows you to dynamically select different columns with complex conditions

```
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

Solo Exercise: Activity #3

Complete Section #2 In Your Lab