GENERAL ASSEMBLY

INTRO TO DATA SCIENCE

PANDAS

LAST TIME:

- LIBRARIES
- NUMPY

QUESTIONS?

I. PANDAS
II. PANDAS
III. PANDAS

EXERCISES: PANDAS

PANDAS

PANDAS is a library that wraps around numpy to work in a data environment more akin to R.

PANDAS is a library that wraps around numpy to work in a data environment more akin to R.

PANDAS greatly improves the data environment with one primary new data type: data frames!

R

```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> data(iris)
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

PANDAS

```
Python 2.7.6 |Anaconda 1.7.0 (x86_64)| (default, Jan 10 2014, 11:23:1
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "help", "copyright", "credits" or "license" for more information
>>> import pandas as pd
>>> iris = pd.read_csv('https://raw.github.com/pydata/pandas/master/p
>>> iris.head()
   SepalLength  SepalWidth  PetalLength  PetalWidth         Name
0          5.1         3.5          1.4         0.2  Iris-setosa
1          4.9         3.0          1.4         0.2  Iris-setosa
2          4.7         3.2          1.3         0.2  Iris-setosa
3          4.6         3.1          1.5         0.2  Iris-setosa
4          5.0         3.6          1.4         0.2  Iris-setosa
```

PANDAS is primarily used for:

Object Creation
Viewing Data
Selecting Data
Finding missing Data
Grouping
Reshaping
Plotting

II. CREATING, VIEWING, SELECTING

PANDAS objects are primarily created two ways:

`pandas.read_csv`
Reads in a csv file that has been formatted

`pandas.DataFrame`
Creates a data frame based on a numpy matrix

Examples:

```
import pandas as pd
import numpy as np
iris = pd.read_csv(‘data/iris.csv’)

normdf = pd.DataFrame(np.random.randn(6,4),
    index=dates,columns=list('ABCD'))
```

PANDAS can be easily manipulated with slicing (like other python objects)

```
iris.head() # Dumps the top/head of the frame
iris.head(5)# Set a ‘head’ to print
iris[:5]    # Same
iris[45:56] # print from 45 to 56
```

PANDAS can also provide information about the data unit as a whole.

```
iris.dtypes
iris.describe()
```

PANDAS can also subset data frames:

```
iris['SepalLength']
iris.SepalLength
```

```
iris[iris.SepalLength > np.mean(iris.SepalLength)].SepalWidth
```

III. PIVOTS, GROUPING, RESHAPING

As noted earlier, PANDAS works with built in numpy functionality, but it also has that functionality built in.

```
np.mean(iris.SepalLength)
iris.SepalLength.mean()
```

The real power comes from being able to aggregate data and find much more functional information using numpy functions alongside PANDAS.

```
pd.pivot_table(iris,values=[‘PetalWidth’],
      cols=['Name']).mean()

iris_group = iris.groupby([‘Name’])
iris_group.mean()
```

It's also relatively simple to begin creating new data that is functioned off of your current data set using apply.

Using apply and writing functions to grok your data is absolutely essential to fully utilize your understanding of data and the utilization of machine learning tools.

```
def short_or_long(x):
    if x > 4:
        return 'long'
    else:
        return 'short'
```

```
def short_or_long(x):
      return 'long' if x > 4 else ‘short'

iris['PetalLengthSummary'] =
      iris.PetalLength.apply(short_or_long)
```

LAB: PANDAS