Week 6
Introduction to Pandas

Melody Huang

What is Pandas?

- Python’s response to R’s DataFrame object
- Combines some functionalities from DataFrames in R, as well as the dplyr library (SQL-like join functions)
- Allows for writing data into CSV and text files (also can write data frames into Excel, SQL data bases, and HDF5, which is commonly used in big data)
- Handling NA’s
- Also commonly used for time series

Basic Syntax

- Create a Pandas Data Frame from scratch:

```
series1 = pd.Series([1,3,5,np.nan,6,8])
#Results in one column
```

- To create a multiple column data frame:

```
df1 = pd.DataFrame({"column1":[3,6,1,7],
                    "column2":[1,7,2,7]})
```

Analogies to R’s Data Frame:

R	Pandas
head(df)	df.head()
tail(df)	df.tail()
summary(df)	df.describe()
df$column1	df[‘column1’]
df[3,]	df.iloc[2]
na.omit(df)	df.dropna(how='any')

Some Nice Additional Features

- Allows for you to easily shift your series as needed
- Example:

```
series1 = pd.Series([1,3,5,np.nan,6,8])
```

1	3	5	NA	6	8

```
print(series1.shift(1))
```

NA	1	3	5	NA	6

Additional Features (cont.)

- Can easily join data frames together using merge:

df1

studentID	name
23095	Jill
10956	Heather
24096	Brad

df2

studentID	grade
23095	A
10956	B
24096	A-

```
pd.merge(df1, df2, on=‘studentID’)
```

Additional Features (cont.)

- Adding rows: `df1.append(df2)`
- Adding columns: `pd.concat(df1, df2)`