Department of Computer Engineering
University of Peradeniya

Data Mining and Machine Learning
Lab 03

July 5, 2017

1 Introduction

Pandas contains high-level data structures and manipulation tools designed to make data analysis fast and easy in Python. This is built on top of NumPy and makes it easy to use in NumPy-centric applications.

1.1 Importing Pandas

```
import pandas as pd
```

1.2 Series and Data Frames

Series and Data Frames are the primary objects which provided by Pandas. A Series object is the base data structure which operates similar to NumPy arrays with one more additional feature, i.e., indexing capabilities. When it is required more than one Series of data that is aligned by a common index pandas DataFrame can be employed.

1.2.1 Creating Series and Data Frame

There are numerous ways to construct Series and Data Frame.

```
# create a series with a list
s = pd.Series([2,4,1,-4,'home'], index=['a', 'b', 'c', 'd', 'e'])
# TODO What is the data type of s? Can it be changed?
# dtype=object

# create a Data Frame with a dictionary
data = {'population': [1.5, 1.7, 3.6, 2.4, 2.9],
 'state': ['Ohio', 'Ohio', 'Ohio', 'Nevada', 'Nevada'],
 'year': [2000, 2001, 2002, 2001, 2002]}
df = pd.DataFrame(data, columns=['year', 'state', 'population',
'debt'], index=['one', 'two', 'three', 'four', 'five'])
```

1.2.2 Accessing and Modifying

```
s[['a', 0]]
s.values[2:]
df[['population', 'state']]
df.population
df.ix[0:,]
df.ix[2:4:,2]
df.iloc[2:4:,2]
df.loc['one']
df.debt = 34.67
df.debt = [ df.ix[:,2][i]*5 for i in range(0, df.shape[0])]
df.head()
df.tail()
df.sample(n=5)
df['newColumn'] = pd.Series(np.random.randn(df.shape[0]),
index=df.index)
```

1.2.3 Loading Data from CSV File

```
df = pd.read_csv('sampleDataSet.csv')
df.shape
df = pd.read_csv('sampleDataSet.csv', names = ['a','b','c',
'd','e','f','g','h','i'])
df.shape
# TODO Comment on result of df.shape() with and without setting
# names
# TODO Try adding header=None parameter without setting names
# parameter
```

1.2.4 Dealing with Missing Data

```
df.isnull().a
df.isnull().sum()
df = df[df.isnull().a != True]
df.dropna(axis=0).isnull().sum()
df.dropna(axis=1)
df.dropna(axis=1, how='all')
df.dropna(axis=1, thresh=1)
df.drop('a', axis=1)
df.fillna(0)
df.replace(0, 5)
df.replace('.', np.nan)
df[np.random.rand(df.shape[0]) > 0.5] = 1.5
```

1.2.5 Applying Functions

Function can be written as a lambda expression or a ordinary function df

```
f = lambda df: df.max() - df.min()

def f(x):
        return x.max() - x.min()

df.ix[:,3:5].apply(f) # applying function element-wise
```

1.2.6 Group Operations

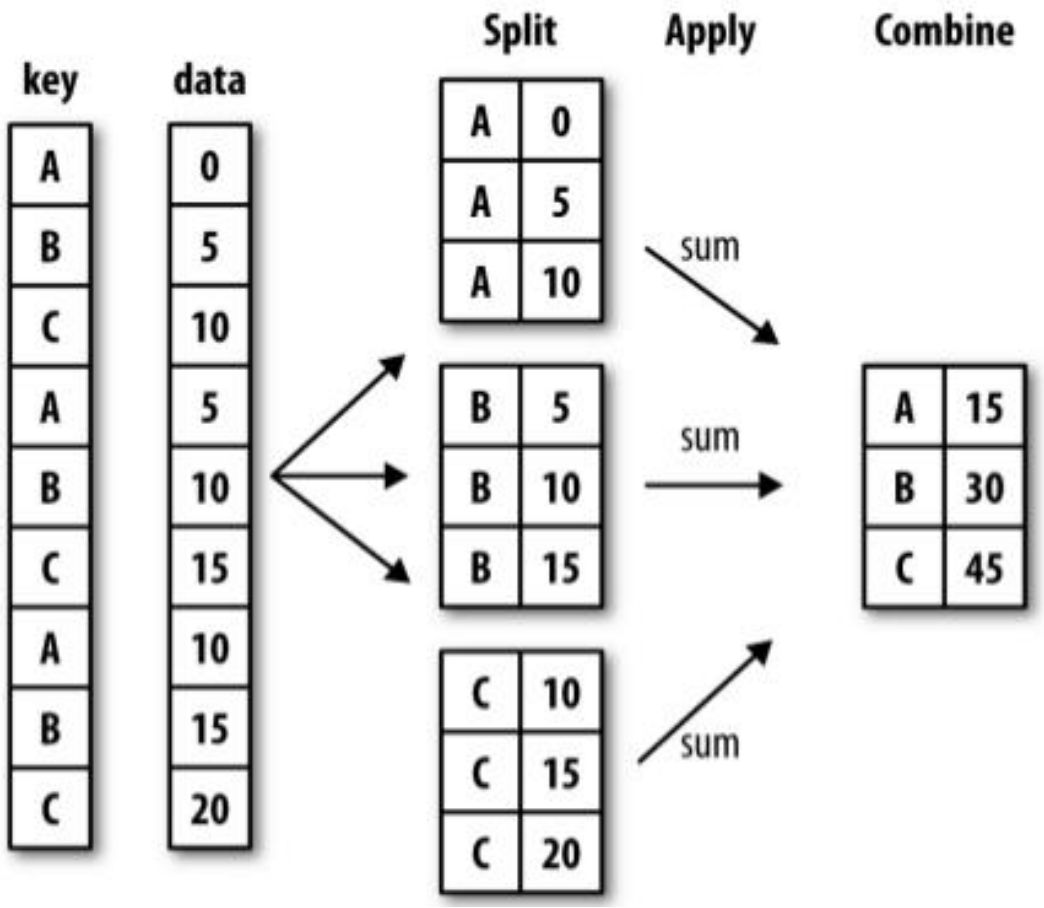

Figure 1: Group Aggregation

```
grouped = df[['a', 'b']].groupby(df['i'])
grouped.mean()
grouped = df[['a', 'b']].groupby([df['i'], df['c']]).mean()
grouped.unstack()
```

1.2.7 Summarize Data

```
df['a'].nunique()
df['a'].value_counts() # count the number of rows with each unique
```

```
# value of variable
df.describe() # basic descriptive statistics for each columns
df.mean()
df.median()
df.sort_index().head()
```

1.2.8 Visualization

```
df.plot(kind='bar')
df.plot(kind='hist')
df.boxplot()
```

1.3 Try Out

Data wrangling is the process of transforming and mapping data from *raw* data into another format with the intent of making it more appropriate and valuable for a variety of downstream purposes such as analytics. Let try to do wrangling on the lab03Exercise dataset which is the same dataset used in lab 01, but this time it comprises of missing values and there is no way of knowing about the dataset. Once you proceed with below steps you would able to understand how Pandas can be used for data wrangling.

1. Apply what you have done with exercise 01 in lab 01 on lab03Exercise dataset and comment on the result.

2. Load the lab03Exercise dataset while specifying column names (channel1, channel2, channel3, channel4 and channel5) using $pandas.read\_csv$ function.

3. If there are any missing values in each column fill those values with the mean of the corresponding column.

4. To see the correlation between one column to all other columns, use following code segment and comment on diagonal plot.

   ```
   from pandas.tools.plotting import scatter_matrix
   scatter_matrix(data, alpha=0.2, figsize=(6, 6), diagonal='kde')
   ```

5. Add a new column named as **"class"** on lab03Exercise dataset. Values of this column either 1 or 0 which should be derived based on the following condition.

   if $((\bar{column}_1 + \bar{column}_5)/2 < (\bar{column}_2 + \bar{column}_3 + \bar{column}_4)/3)$ **then**
   $value \leftarrow 1$
   else
   $value \leftarrow 0$
   end if
   where $\bar{column}_i \in \{1, \ldots, 5\}$ is the mean value of ith column.

2 How to Decide Test Cases for Unit Tests?

Ask yourself. What EXACTLY do I want to test. And test the most important. Test to make sure it basically does what you are expecting it to do in the expected cases.

3 Lab Exercise

Let say X and Y are two random vectors for sample size of N. Then covariance between X and Y can be defined as $cov_{X,Y} = \frac{\sum_{i=1}^{N}(X_i-\bar{X})(Y_i-\bar{Y})}{N-1}$ where $\bar{X}$ and $\bar{Y}$ are the mean values of X and Y random vectors. Then correlation of (X, Y) is defined by $cor_{X,Y} = \frac{cov_{X,Y}}{sd(X)sd(Y)}$ where sd(X) is standard deviation of X and sd(Y) is standard deviation of Y. Sample standard deviation can be defined as $S_x^2 = \frac{\sum_{i=1}^{N}(X_i-\bar{X})^2}{N-1}$.

1. Calculate the covariance matrix and correlation matrix for the provided dataset (lab03Exercise). As an example, if you have four random variables (a,b,c and d) covariance and correlation matrices of those can be defined as

$$\textbf{covarianceMatrix} = \begin{bmatrix} V_a & C_{a,b} & C_{a,c} & C_{a,d} & C_{a,e} \\ C_{a,b} & V_b & C_{b,c} & C_{b,d} & C_{b,e} \\ C_{a,c} & C_{b,c} & V_c & C_{c,d} & C_{c,e} \\ C_{a,d} & C_{b,d} & C_{c,d} & V_d & C_{d,e} \\ C_{a,e} & C_{b,e} & C_{c,e} & C_{d,e} & V_e \end{bmatrix}$$

$$\textbf{correlationMatrix} = \begin{bmatrix} r_{a,a} & r_{a,b} & r_{a,c} & r_{a,d} & r_{a,e} \\ r_{a,b} & r_{b,b} & r_{b,c} & r_{b,d} & r_{b,e} \\ r_{a,c} & r_{b,c} & r_{c,c} & r_{c,d} & r_{c,e} \\ r_{a,d} & r_{b,d} & r_{c,d} & r_{d,d} & r_{d,e} \\ r_{a,e} & r_{b,e} & r_{c,e} & r_{d,e} & r_{e,e} \end{bmatrix}$$ where V_a is the variance of a, $C_{a,b}$ is the covariance between a and b and $r_{a,b}$ is correlation between a and b.

2. If X and Y are uncorrelated, if only $cov_{X,Y} = 0$. By looking at result you get from exercise 1 explain correlation between columns.

3. Comment on the condition which is given in section 1.3 is used to derive the class value based on result you got in exercise 01 and suggest a proper condition for deriving class value.

4 Submission

Submit a single .py file as [12|13]xxxlab03.py where xxx is your registration number. Add answers for questions 2 and 3 as comments in the same file.

5 Important

Make sure that you have the basic understanding of today's lab. This lab is really important for successive labs. If you do not understand any concepts, make sure you get some help from instructors.

6 Deadline

July 18, 23:59:59 GMT+5:30.