CSCI446/946 Big Data Analytics - Week 3

Lab 2 – Data Preparation with Python

Introduction

This instruction is developed on Python programming language (or use Spyder) to run `Lab2.py`. In Lab2, you will practice big data analytics phase 2: Data Preparation by breaking it down to Task 1, Task 2 and Task3. You are required to

1. download `yearly_sales.csv` and `Lab2.py` from Moodle > Week3 – Toggle > Lab 2 files
2. implement the code shown in `blue-colou` into `Lab2.py`
3. Create a Word document to write down your report by 1) explaining what you will do next according to the "**Describe"s**, and 2) answering **Questions**
4. format your report: title, heading, body of text, code, programming results, etc.

Import packages:

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from scipy import stats
```

Task 1: Load data and basic evaluation

The first section gives an overview of how to use Python to acquire, parse, and filter the data, as well as how to obtain some basic descriptive statistics on a dataset [1].

Lab2 uses `yearly_sales` dataset, which provides information about the annual sales in U.S. dollars for 10,000 retail customers. The dataset is restored in the form of a comma-separated-value (CSV) file. The `read.csv()` function is used to import the CSV file.

Describe 1: import a CSV file of the total annual sales for each customer
Code 1:

```
sales = pd.read_csv("yearly_sales.csv", index_col=0)
print('sales:\n',sales)
```

Question 1: What's the variable name which restores `yearly_sales` data? What is the data structure, i.e., the meaning of rows and columns, data type of each column? Is there any missing data?

Describe 2: examine the imported dataset
Code 2:

```
print('sales head:\n', sales.head())
print('sales shape:', sales.shape)
print('sales describe:\n', sales.describe())
```

Question 2: What is the `head()` function? What can we get by running `sales.head()` and their meanings? What is the size, e.g., number of rows and columns? What is the

`describe()` function? What can we get by running `sales.describe()` and their meaning? Is there any abnormal result?

Task 2: Visualization in data evaluation

The second section examines using Python to perform exploratory data analysis tasks using visualization [1].

Describe 3: plot num_of_orders vs. sales
Code 3:

```
plt.scatter(sales['num_of_orders'], sales['sales_total'])
# Add title and axis names
plt.title('Number of Orders vs. Sales')
plt.xlabel('num_of_orders')
plt.ylabel('sales_total')
# Show graph
plt.show()
```

Question 3: What are x and y axes in the plotted figure? What is the meaning of a plotted data point? What is the observed relationship between these two variables (i.e., x and y axes)? What is the reason for the initial decision to apply linear regression?

Describe 4: Perform a statistical analysis (fit a linear regression model)
Code 4:

```
y = sales['sales_total'].values.reshape(-1, 1)
X = sales['num_of_orders'].values.reshape(-1, 1)
model = LinearRegression()
model.fit(X, y)
# summary of the model
print('linear regression model intercept :',
model.intercept_.item())
print('linear regression model coefficients : ',
model.coef_.item())
print('linear regression Model score : ', model.score(X, y))
```

Question 4: What can we get from linear regression model? Can you draw a conclusion from the `summary of the model`?

Describe 5: perform some diagnostics on the fitted model, plot histogram of the residuals
Code 5:

```
y_pred = model.predict(X)
residuals = y - y_pred
plt.hist(residuals, bins = 800)
plt.show()
```

Question 5: What are x and y axes in the plotted figure? What is the meaning of a plotted bar? What conclusion can be drawn from this figure? What's the difference in changing the value for variable `bins`? What is the most suitable value for `bins`? and why?

Task 3: Statistical evaluation

The third section focuses on statistical inference, such as hypothesis testing and analysis of variance in Python [2].

Describe 6: perform a t-test
Code 6:
```
t, p = stats.ttest_ind(X, y, equal_var=True)
print('t-test: t=',t.item(),'p=',p.item())
```
Question 6: Explain the t-test results as far as you can. Is there any irregular result?

Describe 7: obtain t value for a two-sided test at a 0.05 significance level
Code 7:
```
n1 = X.shape[0]
n2 = y.shape[0]
df = n1 + n2 - 2
t005 = stats.t.ppf(q=1-0.05/2, df=df)
print('when p = 0.05, t=',t005)
```
Question 7: Explain a two-sided hypothesis test based on results running on Code 6 and Code 7.

Describe 8: perform a Welch's t-test
Code 8:
```
t_welch, p_welch = stats.ttest_ind(X, y, equal_var=False)
print('Welchs t-test: t=', t_welch.item(), 'p=',
p_welch.item())
```
Question 8: Explain the Welch's t-test results as far as you can. Is there any irregular result? What are the differences between t-test and Welch's t-test?

Describe 9: perform a Wilcoxon Rank-Sum test
Code 9:
```
_, t = stats.ranksums(X, y)
print('Wilcoxon rank-sum test: t=', t.item())
```
Question 9: Explain the Wilcoxon Rank-Sum test results as far as you can. Is there any irregular result? What are the differences between Wilcoxon Rank-Sum test, t-test and Welch's t-test?

Describe 10: perform an ANOVA test
Code 10:
```
fvalue, pvalue = stats.f_oneway(sales['sales_total'],
sales['num_of_orders'], sales['gender'].replace('F',
1).replace('M', 2))
print('ANOVA test: f=', fvalue, 'p=', pvalue)
```
Question 10: Explain the ANOVA test results as far as you can. Is there any irregular result? What are the differences between ANOVA test, Wilcoxon Rank-Sum test, t-test and Welch's t-test? What is the most suitable statical evaluation method for `yearly_sales` dataset?

Reference

[1] E. E. Services, Data science and big data analytics: discovering, analyzing, visualizing and presenting data, Chapter 3.1, Wiley, 2015.

[2] E. E. Services, Data science and big data analytics: discovering, analyzing, visualizing and presenting data, Chapter 3.3, Wiley, 2015.