Data Manipulation

**As you proceed with the assignment, follow the written instructions. Screenshots are provided ONLY as a reference.**

**Make sure you submit all screenshots with a clearly visible menu bar including the date and timestamp.**

### Objective

The objective of this assignment is to develop skills manipulating a very large data set in excess of 1 million rows using simple commands in a user-friendly notebook. Jupyter Notebooks provided by Anaconda is an emerging toolset taking a position in the world of business analysis similar to the historic application of the spreadsheet. Familiarity with the Jupyter Notebook provides the ability to clean and refine a very large data set in a short period of time.

### Setup

Install the latest version of Anaconda using the link below.

<https://www.anaconda.com/distribution/>

After installation the Anaconda Navigator window should be open on your desktop. Using the Anaconda Navigator, click on the launch button in the *Jupyter Notebook* application window.

**A screenshot of a cell phone

Description automatically generated**

In the Jupyter Notebook *Files* tab, navigate to your data folder. Click the down arrow beside *New* and create a *Python 3* notebook.

**A screenshot of a cell phone

Description automatically generated**

Rename the notebook by clicking on the word *Untitled* located in the top left corner of the header. Name your notebook with your FirstName\_LastName.

A screenshot of a cell phone

Description automatically generated

**Procedure**

**Step 1: Import the required libraries**

**Create a heading -** To create a heading, write your heading in markdown cells (windows) using the # character.Test an example by writing a # followed by your text heading. Click on run to see the output.

For example, if we write the statement *# Importing Requirements* in the cell and click on *Run* we will see the output. This is the way that a Jupyter Notebook works. We type lines of code in a notebook window, and when we click *Run* the notebook executes the lines of code we provided. The feedback is immediate and provided right below our window.

A screenshot of a cell phone

Description automatically generated

After clicking on *Run*, the code executes and looks like the screen capture below.

A screenshot of a cell phone

Description automatically generated

We now have a heading for this section of code. With a heading we will proceed. Throughout this assignment, you will create headings in a similar manner for each section of code. These headings help to organize different sections of our project.

Write the following code in the next cell, this will increase the width of the working area to the maximum. Remember to click *Run* after entering the code.



After you run the IPython code, write the following to import all of the required libraries we plan to use.

A screenshot of a cell phone

Description automatically generated

**Question**

1. **Take a screen shot of your page, showing page Title (your name) and page background.**

**Step 2: Setting the directory and reading the data file**

Add the code lines found in the next screen shot.

**Text

Description automatically generated**

**Explanation of the code**

* path – This is the folder address where you saved the dataset.
* data\_ dir - The file name for the dataset.
* data\_ output – The name of the file you will create. (used later in the assignment)

Using this code, we set the directory as the path, and read the file from that path.

Note: The path usually starts with C:/ for Windows users and /Users/yourname/ for Apple users. It is the path where you saved the file, for example Downloads or Desktop.

**# Exploring the data set**

Add the following code to see the different sheets present in the data.

**A screenshot of a cell phone

Description automatically generated**

**Question**

**b. How many Sheets are present, please name each of them?**

It’s time to explore the data elements present in the sheets. Enter the following code. Remember to select *Run* after each window of code entry.

**A screenshot of a social media post

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

*Head*is a function used in python to display the first few rows at the top of the dataset.You will see the column headings displayed with the first few rows of data.

**Step 3: Data manipulation**

To remove blank rows, we need to check and see if blank rows are present in our data set.

A close up of a logo

Description automatically generated

This code will provide the number of columns in the data set, what they are, and the number of non-null values.

**Question**

1. **How many non-null rows are present in the data set? (RangeIndex provides the total row count, columns provide the non-null row count)**

Enter the code below to see the number of null values present in each column.

**A screenshot of a cell phone

Description automatically generated**

A review of the output suggests that we have null values in every column. There are also some entries where all the values of a row are NULL. We will delete these rows. Null rows do not provide value for our data set. Enter the code below in the next window.

**A screenshot of a cell phone

Description automatically generated**

Now, we can check our work to see if the null values are present.

**A screenshot of a cell phone

Description automatically generated**

**Question**

1. **How many empty rows were present in the data set before we removed them?**

Perform the same null rows check for the products sheet.

**A screenshot of a cell phone

Description automatically generated**

**Question**

1. **Using the code above, how many empty rows are present in the Products sheet? Paste the Screenshot of the output showing page title (your name).**

**Step 4: Imputing the blank cells in the price column**

As you can see,many rows have missing values in the price column. Since the total number of missing values exceed 2% of the entire dataset, we cannot delete the rows. To clean the data, we will impute values into the blank cells. Imputing means we are estimating the values that should have been present in the blank cells and inserting them. The method to be used for imputation is determined by looking at a histogram. A normal distribution adopts the mean value for imputation. A skewed histogram suggests adopting the median as the method for imputation.

To plot a histogram with *hist*, we need to create bins(groups) that range from 1 to 27. Now compute the *Mean* and the *Median*. To achieve the results, add the following statement to a code window.

A screenshot of a cell phone

Description automatically generated

It is acceptable to ignore the warning messages in this step.

**Question**

1. **Paste a screenshot of the histogram and explain the methodology chosen for imputation.**

After you have chosen your imputing method, we will replace the null values in the price column with imputed values. Using the histogram, chose the *Mean* or *Median* imputing method that is appropriate for your distribution. (review the lecture for more information on this subject) Use the first line of code if you want to use the *Median*, or use the second line of code if you decide to use the *Mean* as your imputing method. You must choose one of the two methods.

A screenshot of a cell phone

Description automatically generated

Use the following line of code to check the number of Null values that are present.



**Question**

1. **Paste a screenshot of the screen (with your name displayed) with the imputed method code showing the result of the *.isna().sum* function. How many null values are there?**

**Step 5: Removal of duplicate values**

We need to correct the names for countries with spelling errors and duplicate entries. Make sure only one record exists for each country (delete any one duplicate entries)

To obtain a list the unique values in the country column use the code below.

**A close up of a logo

Description automatically generated**

**Question**

1. **List the countries with duplicate entries and/or misspellings.**

Remove spelling errors by executing the following code.

**A close up of a logo

Description automatically generated**

**Question**

1. **Run the unique value command again and paste the screenshot of the whole screen. Explain what has happened.**

We have updated our data set for the spelling errors. Now let’s check for duplicate entries. The *duplicated* function will highlight the number of duplicate entries that are present. A duplicate entry here means that 2 rows have exact same entries for all columns.

A screenshot of a cell phone

Description automatically generated

**Question**

1. **How many duplicate entries are present?**

Use the following code to see the duplicate entries.

A screenshot of a cell phone

Description automatically generated

**Question**

1. **Explain the outcome of the executed query. Paste the screenshot of the code and the output using the entire screen.**

**T**o remove the duplicates from the data frame, use the following code.

**A screenshot of a cell phone

Description automatically generated**

**Question**

1. **Use the duplicated(). Sum() function from above, to again check the number of duplicate entries present now. Paste the screenshot of the code and the output using the entire screen.**

**Step 6: Merging both sheets present in the excel file**

Create a heading using *# Merging Both Data Sets* query and run it as a heading. Use the code statements below to join the data sets.

**A screenshot of a cell phone

Description automatically generated**

Now we need to sort the entire file by country column. Use the code below to perform this adjustment.

A screenshot of a cell phone

Description automatically generated

**Question**

1. **Paste the screenshot of the head of *df\_full*.**

Now you can see that our file is sorted, and we have removed all the duplicates. It is now time for us to export the output to a file. Enter the code found below.

A picture containing screenshot

Description automatically generated

Make sure the data\_output is set correctly with the requested filename. Remember to select *Run*.

A screenshot of a cell phone

Description automatically generated

The output will create a cleaned data file in an excel format.

Now we need to create a file to capture your code in this notebook. Using the menu path File>Download As>Notebook, create your .ipynb file for this notebook.

**Submission Instructions**

1. Submit the assignment document in microsoft word.
2. Submit the excel file.
3. Submit the .ipynb file created.

Make sure you attach all three files before clicking on submit.