



# Analysis of the Starbucks Beverage Menu

## A Step-by-Step Guide

*A Signature Term Project designed for*

**BUS 336: Python for Business Analytics**

*Revised 8/4/23*



# Table of Contents

<b>Table of Contents .....</b>	<b>2</b>
<b>Introduction.....</b>	<b>4</b>
<b>Overview of the Term Project .....</b>	<b>5</b>
<b>Project Objectives:.....</b>	<b>5</b>
<b>Deliverable One: Open a new Jupyter Notebook and load CSV file into the Jupyter Notebook .....</b>	<b>6</b>
<b>Deliverable One Tasks: .....</b>	<b>6</b>
<b>Task #1: Download the Starbucks_Beverage_Menu_Clean CSV file.....</b>	<b>6</b>
<b>Task #2: Load the CSV file into a new Jupyter Notebook using the pandas package. ....</b>	<b>6</b>
<b>Task #3: Rename the Jupyter Notebook.....</b>	<b>7</b>
<b>Deliverable Two: Explore rows with all blank values and remove them. ....</b>	<b>7</b>
<b>Deliverable Two Tasks: .....</b>	<b>7</b>
<b>Task #1: Using the isnull and sum functions, check the total NaN entries .....</b>	<b>7</b>
<b>Task #2: View the first 15 rows of the dataframe with the head function.....</b>	<b>7</b>
<b>Task #3: Assign the dataframe to df1 and remove the row with all blank entries.. ....</b>	<b>8</b>
<b>Task #4: Validate that the blank row(s) have been removed.....</b>	<b>8</b>
<b>Deliverable Three: Explore the data types and correct any errors.....</b>	<b>8</b>
<b>Deliverable Three Tasks: .....</b>	<b>8</b>
<b>Task #1: Using the info function, check the columns, NaN counts and data types.....</b>	<b>8</b>
<b>Task #2: Look for column with the incorrect data type, understand why it is incorrect, and change it to the correct data type.....</b>	<b>9</b>
<b>Task #3: Validate that the Sugar_g variable has been changed to a float.....</b>	<b>10</b>
<b>Deliverable Four: Explore the NaN values and remove the row with any NaN values.....</b>	<b>10</b>
<b>Deliverable Four Tasks: .....</b>	<b>10</b>
<b>Task #1: Using the info function, check for one NaN value. ....</b>	<b>10</b>
<b>Task #2: Assign a new dataframe df2 and remove the row with the NaN entry.....</b>	<b>10</b>
<b>Task #3: Validate that it has been corrected.....</b>	<b>11</b>
<b>Deliverable Five: Change a column name. ....</b>	<b>11</b>
<b>Deliverable Five Tasks: .....</b>	<b>11</b>

[Click here to go back to Table of Contents](#)



Task #1: Assign a new dataframe and change a column header name.....	11
Task #3: Validate that it has been corrected.....	11
<b>Deliverable Six: Describe the continuous and categorical data.....</b>	<b>11</b>
Deliverable Six Tasks: .....	11
Task #1: Use the describe() function to describe the continuous variables. ....	12
Task #2: Use the describe() function to describe the categorical variables. ....	12
Task #3: Analyze, interpret, and write about the continuous and categorical variables. ....	12
<b>Deliverable Seven: Write the dataframe to a CSV file, save it, and verify it as a CSV file.....</b>	<b>12</b>
Deliverable Seven Tasks:.....	13
Task #1: Using the to_csv function, write dataframe df4 to a CSV file. ....	13
Task #2: Check the C:\Users\YOUR COMPUTER NAME FOLDER for df4. ....	13
Task #3: Assign a new dataframe and load the CSV file back into the Jupyter Notebook.....	13
Task #4: Open the newly created CSV file in the Jupyter Notebook.....	13
<b>Deliverable Eight: Upload file into your GitHub Account.....</b>	<b>14</b>
Deliverable Eight Task:.....	14
Task #1: Open your GitHub account and upload the newly created df4.csv file.....	14
Task #2: Validate that it has been uploaded and can be opened by emailing it to yourself and instructor. ....	14
<b>Deliverable Nine: Make a short video explaining deliverables one through nine.....</b>	<b>14</b>
Deliverable Nine Tasks:.....	14
Task #1: Rehearse, record, and present a video presentation. ....	14
Task #2: Upload the video presentation.....	15
<b>Deliverable Ten: Write a reflection.....</b>	<b>15</b>
Deliverable Ten Tasks: .....	15
Task #1: Write a reflection in the Python file. ....	15
<b>Appendix A, Grading Rubrics.....</b>	<b>16</b>
Deliverables Grading Rubric.....	16

## Introduction

Much business can be viewed as somewhat abstract or theoretical in nature and challenging to grasp, ranging from beginning to advanced business classes. Some students fair very well in this abstract learning environment, but many do not. Textbooks do their best to teach students how to apply these abstract concepts by assigning homework problems of hypothetical businesses and complete case studies of real companies. This learning strategy often falls short to exposure to real, concrete experiences dealing with real business issues.

This project is designed to address the lack of exposure dealing with real life businesses in at a high level active, or experiential learning environment. Many business classes do not offer opportunities to go outside of the classroom and try to incorporate what is learned in the books and students seemingly get stuck in the abstract, theoretical world. In this project, we attempt to provide an opportunity for students to make the connection between abstract, theoretical textbook learning to real, concrete experiences dealing with business analytical issues. This approach allows students to transform their learning from the abstract world to the real, concrete world by producing experiences that allow them absorb information and analyze a business more thoroughly.

This project addresses the theory-practice gap by exploring the Starbucks Corporation, which is a well-known, easily accessible publicly traded business entity. At the conclusion of the project, students will come to understand business analytics theory much better and how it is applied. This sets them up for the life-long journey to better understand information and improve chances to make more informed business decisions across various industries.

The instructions contained in this document are specifically written to allow students to connect the theoretical discussion to practical real world, concrete experiences driving toward a rich learning environment. It is hoped that students will take the opportunity to engage in this term project as a path to understanding abstract business analytics concepts, dealing with a real live business that has an impact in the communities where students may live/work.

It is hoped you will enjoy your experience.

If you see any errors or areas that need more clarity, please bring them to the attention of Dr. Joseph Foy at joseph.foy@cuny.edu. This document has undergone numerous edits based upon many suggestions from past students who have successfully completed the project. First time students are highly encouraged to give suggestions as well.

Good luck with this term project and let it enrich your understanding of business analytics!

Dr. Foy

**[Click here to go back to Table of Contents](#)**

## Overview of the Term Project

An upstart competitor in the coffee business would like to better understand Starbucks' beverage inventory. You have been hired as a consultant to help the upstart competitor to assist in this task. You could do the analysis in Excel, but the company hiring you would like you to share all the steps you take to analyze the data. In Excel, you could do this in Visual Basic coding. Basic Visual coding is the coding that is used behind the scenes in Excel. Here is the problem that you run against. Even if you possess the knowledge of how to code in Visual Basic, someone in the company that is hiring you would also have to know it, as well as anyone they would like to share it with. Visual Basic is not a widely used programming language. It is a Microsoft-specific product; therefore, not many programmers work with it. Fortunately, many programmers know how to code in open-access Python. Your client has obtained the extensive Starbucks beverage menu in a CSV file and provided you with it. Your overall objective is to import the file Starbucks beverage menu into Python, analyze and interpret the data, and make your code available to the company hiring you and present how you coded your analysis.

The project has six objectives, as follows:

### Project Objectives:

- Project Objective 1: Organize complex data for business analysis.
- Project Objective 2: Explore business data for insights.
- Project Objective 3: Synthesize concepts to structure data to allow for business analysis.
- Project Objective 4: Share business data to different audiences.
- Project Objective 5: Present business data to make decisions.
- Project Objective 6: Reflect with a critical review of what students learned in the course and how Python can build your skill set.

To achieve the objectives, this project requires that you complete ten deliverables. Each deliverable has specific tasks. Each deliverable is weighed based on the level of work involved. The term project is due at the end of the semester.

Grading rubrics are provided in [Appendix A, Grading Rubrics](#) of this document so students can see the weight of each deliverable and how they are to be scored. Students should become familiar with the grading rubrics as early as possible. If students have any questions, they are to email their instructor and explain the issue as succinctly as possible. The grade for the project will appear in the Course Learning Management System.

To be successful with this term project, students integrate the course materials with the project requirements. Therefore, students should become familiar with the course homework assignments, required course readings, supplemental readings, and course videos. If you are curious about Starbucks and its operations, here are helpful videos to understand Starbucks store level operations.

- [A Behind the Scenes Look at Starbucks Retail Operations](#)
- [Day in a Life: Nick Lister, Starbucks Barista](#)
- [Beverage Sequencing](#)

**[Click here to go back to Table of Contents](#)**

## **Deliverable One: Open a new Jupyter Notebook and load CSV file into the Jupyter Notebook**

Deliverable one focuses on term project objective to organize complex data for business analysis. This objective is designed to allow students to become familiar with organizing files within their own computer and load them into a commonly used Python IDE. This is normally the first step in the process to analyze business data.

Students will have to perform the tasks below. The first deliverable is divided into three tasks that must be performed in the exact sequence that is shown in this guide. It is also important that students become familiar with the grading rubric to understand how this deliverable is graded.


### **Deliverable One Tasks:**

- Task 1: Download the BUS336\_Signature\_Term\_Project\_Starbucks\_Beverage\_Menu CSV file from either the Projects/Assignments folder in Blackboard or the GitHub account provided. Save the CSV file in your C:\Users\YOUR COMPUTER NAME FOLDER.
- Task 2: Import pandas as pd, then load the BUS336\_Signature\_Term\_Project\_Starbucks\_Beverage\_Menu CSV file into the new Jupyter Notebook using the pandas package. Assign it to new object, df.
- Task 3: Rename the Jupyter Notebook from Untitled to the following name convention: LastName\_FirstName\_BUS 336\_Signature\_Term\_Project.

### ***Task #1: Download the Starbucks\_Beverage\_Menu CSV file***

The BUS336\_Signature\_Term\_Project\_Starbucks\_Beverage\_Menu CSV file can be found in Blackboard under the Projects/Assignments folder or in the GitHub folder provided by your instructor. This is a “dirty” CSV file in the respect that it has been originally downloaded as PDF file from a website and converted to a CSV file. However, when downloaded from a PDF file, it did not retain the column names correctly, contained entries which did not retain the same formatting as the PDF, contained blank rows, and had numerous numeric data characterized as objects (that is, strings). Although the CSV file has been cleaned up quite a bit, it still contains items that must be further cleaned in order to conduct exploratory data analysis. Your job is to load this nearly clean file and make further adjustments to it. Once the file is loaded into Python, correct the issues so it is presentable as a “clean” CSV file and ready to share with anyone.

### ***Task #2: Load the CSV file into a new Jupyter Notebook using the pandas package.***

Students are required to import pandas as pd and use the pd.read\_csv command to load the CSV file into the Jupyter Notebook and assign it to new object, df. In Jupyter Notebooks, when the run button  is clicked, a \* will temporarily appear indicating that the file is loading. It will turn into a number when it is successfully loaded.

**[Click here to go back to Table of Contents](#)**

Hint: `df = pd.read_csv('BUS336_Signature_Term_Project_Starbucks_Beverage_Menu')`

### ***Task #3: Rename the Jupyter Notebook***

It is important to rename the Jupyter Notebook from Untitled to the following name convention: LastName\_FirstName\_Starbucks\_Project. Use your real last name and first name. The file will be stored in the your C:\Users\YOUR COMPUTER NAME FOLDER.

## **Deliverable Two: Explore rows with all blank values and remove them.**

Deliverable two begins the process of exploring business data types and deal with NaN (Not a Number) entries. The objectives of deliverable two are to 1) Explore business data for insights, and 2) Synthesize concepts to structure data to allow for business analysis.

### **Deliverable Two Tasks:**

Task 1: Using the `isnull` and `sum` functions, check the total NaN entries for all columns in the dataframe.

Task 2: View the first 15 rows in the dataframe with the `head` function.

Task 3: Assign the dataframe to `df1` and remove the row with all blank entries.

Task 4: Validate that it has been removed by running the `isnull` and `sum` functions again.

### ***Task #1: Using the isnull and sum functions, check the total NaN entries***

The `isnull` and `sum` functions can be used to explore which data type each column. This is important to immediately assess any adjustments that need to be made.

Hint: `df.isnull().sum()`

When running these functions, you should see that all the columns have just one NaN entry, except one column with two NaN entries.

### ***Task #2: View the first 15 rows of the dataframe with the head function.***

To view rows, we can use the `head()` function. With no entry between the `()`, the `head` function will enable one to quickly view the first five rows to assess the dataframe. However, if one wants to view a specific number of rows, specify the actual number between the `()`.

Hint: `df.head(number of rows)`

**[Click here to go back to Table of Contents](#)**

### *Task #3: Assign the dataframe to df1 and remove the row with all blank entries..*

In Python, the dropna function can be used to remove rows. However, to delete rows with just NaN in each column, one needs to tell Python how to do it. The (how='all') option is placed after the dropna function to accomplish this.

Hint: `df1 = df1.dropna(how='all')`

To better understand what is happening behind the scenes in Python when assigning a name to a dataframe, consider this explanation. The new object (that is, new dataframe name – in our case, df1) takes the old dataframe (df), changes it, and gives it a new name (df1). As an analogy to understand what is happening, let's compare this to Excel. In Excel, this is akin to opening up a worksheet in tab one called "Sheet1", then renaming "Sheet1" to "df". Then, do a little work in df, then copy and paste the contents of "df" into a new worksheet, Sheet2 and rename it "df1". Now you have df and df1 in the Excel file. The df worksheet has the original dataframe, and df1 has the updated dataframe. Anytime that you rename a dataframe, it is just like opening up a new tab in Excel and copy and pasting the contents from some other worksheet.

### *Task #4: Validate that the blank row(s) have been removed.*

Use the isnull and sum functions to validate that df1 has only one column remains with a NaN.

Hint: `df1.isnull().sum()`

## **Deliverable Three: Explore the data types and correct any errors.**

Deliverable three continues the process of exploring business data types and correct any errors. The objectives of deliverable two are to 1) Explore business data for insights, and 2) Synthesize concepts to structure data to allow for business analysis.

### **Deliverable Three Tasks:**

- Task 1: Using the info function, check the column names, Non-Null Counts, and the Dtype (data type) in the dataframe.
- Task 2: Look for column with the incorrect data type, understand why it is incorrect, and change it to the correct data type.
- Task 3: Validate that the Sugar\_g variable has been changed to a float. Only validate the column changed and not the whole dataframe.

### *Task #1: Using the info function, check the columns, NaN counts and data types.*

The info function is used to explore which data type each column. This is important to immediately assess any adjustments that need to be made.

[Click here to go back to Table of Contents](#)



Hint: df1.info()

***Task #2: Look for column with the incorrect data type, understand why it is incorrect, and change it to the correct data type.***

Using the info function in the first task, the Dtypes column revealed to you that the Sugar\_g has an incorrect string entry when the entries in the column are intended to be numeric. When strings are in a column, Python recognizes them as an object. The read out from the info function shows you that the Sugar\_g is an object. To fully understand why, examine the df dataframe closely and look down the Sugar\_g column, you will see a “zero” entry instead of 0. Python will recognize the “zero” as a string. To do any type of meaningful statistical analysis, we need to convert the string “zero” to an actual numerical 0.

The following table has code that is already inserted in the Python file to facilitate this conversion. The code explains how the “zero” to 0 conversion works. For this task, the “zero” value has been converted to a 0 for you. You do not need to do anything regarding the string to number conversion. However, you will need to convert the date type from an object to a float, as explained after the following code.

<pre># In the dataframe, the Sugar_mg column has an incorrect string entry when it should be numeric. # A 0 value was entered as "zero". This will not result in a NaN, but Python will recognize this as a string. # We need to convert the string “zero” to an actual 0. As a demonstration, we use the current dataframe that we are # working with and use the index numbers to isolate the row and column. The iloc function helps us isolate that the entry # zero is in the second row, 14th column (index positions 1 and 13). We set that cell value to the number 0 and it will change the "zero" to 0. # Run the code and look compare the Sugar_g column in df1 to df and see how the "zero" turns into a 0. df1.iloc[1,13] = 0 df1</pre>
---

Although the “zero” has now been changed to a 0, the data type did not change, and Python still cannot do any meaningful statistical analysis. Python still recognizes the column as an object. We still need to change the column to a float (a numeric value with decimals). To make the correction, we can use the .astype(int) function. You do not need to create a new object to assign the changes to an updated dataframe. You just make the change to an existing column in the dataframe you are working in. In this case, you are still working with df1 and you are going to only change the data type of the Sugar\_g column.

**[Click here to go back to Table of Contents](#)**

Hint: `df1['Column_Name'] = df1['Column_Name'].astype(float)`

***Task #3: Validate that the Sugar\_g variable has been changed to a float.***

The dtypes function can be used to validate the change.

Hint: `df1.Column_Name.dtypes`

**Deliverable Four: Explore the NaN values and remove the row with any NaN values.**

Deliverable four continues the process of exploring business data types and deal with NaN (Not a Number) entries. The objectives of deliverable two are to 1) Explore business data for insights, and 2) Synthesize concepts to structure data to allow for business analysis.

**Deliverable Four Tasks:**

Task 1: Using the info function, determine the column in the df1 dataframe with an entry that still has one NaN value.

Task 2: Assign the dataframe to df2 and remove the row with the NaN entry.

Task 3: Validate that it has been corrected by running the info function again.

***Task #1: Using the info function, check for one NaN value.***

The info() function will give important information that will disclose how many rows and columns are in the dataframe. It will also contain a column that has Non-Null Count information. If you see a count that is not the same as the columns that you are certain that do not contain NaN values (which can be discovered with the isnull and sum functions), then there are NaN values. For example, if the Non-Null Count has multiple “1204 non-null” entries, but one has a “1203 non-null” entry, then that column still has a NaN value.

Hint: `df1.info()`

Hint: look in the Saturated\_Fat(g) column.

***Task #2: Assign a new dataframe df2 and remove the row with the NaN entry.***

Since only one column will contain a NaN value, one would have to consider the values in the entire row relative to the dataframe. If one believes that the whole row can be removed, and is confident that any other remaining NaN values are inconsequential in the dataframe, one can remove all the NaN values in the dataframe. This can also be accomplished with the dropna function.

Hint: `df2 = df1.dropna()`

**[Click here to go back to Table of Contents](#)**

### ***Task #3: Validate that it has been corrected.***

Validate that it has been corrected by running the info function again.

Hint: df2.info()

## **Deliverable Five: Change a column name.**

Deliverable five continues the process of exploring business data types and deal with column names that do not make sense or are incorrect. The objectives of deliverable two are to 1) Explore business data for insights, and 2) Synthesize concepts to structure data to allow for business analysis.

### **Deliverable Five Tasks:**

- Task 1: Assign the dataframe to df3 and using the rename function, change the Caffiene\_milligrams column header from the df2 dataframe to Caffiene\_mg.  
Task 2: Validate that it has been changed by running the head() function.

### ***Task #1: Assign a new dataframe and change a column header name.***

Not every spreadsheet that is imported from Excel has the exact column names that you may want. Changing column names is a standard practice in business analytics. This can be accomplished with the rename function.

Hint: new\_dataframe\_name.rename(columns = {'old\_column\_name': 'new\_column\_name'})  
df3 = df2.rename(columns = {'Caffiene\_milligrams': 'Caffiene\_mg'})

### ***Task #3: Validate that it has been corrected.***

Validate that it has been corrected by running the head() function again. Refer to the hints above regarding validating. Make sure that you are validating df3.

## **Deliverable Six: Describe the continuous and categorical data.**

Deliverable six continues the process of exploring business data types and deal with column names that do not make sense or are incorrect. The objectives of deliverable two are to 1) Explore business data for insights, and 2) Synthesize concepts to structure data to allow for business analysis.

### **Deliverable Six Tasks:**

[Click here to go back to Table of Contents](#)

- Task 1: Using dataframe df3, use the describe() function to describe the continuous variables.  
Task 2: Using dataframe df3, use the describe() function to describe the categorical variables.  
Task 3: Analyze and interpret what you understand about the continuous and categorical variables. Write about it as a comment.

***Task #1: Use the describe() function to describe the continuous variables.***

The dataframe contains continuous variables. Continuous variables are numeric. With continuous variables, we can get the measures of central tendency, such as the mean and median, as well as variance information such as standard deviations. Obtaining these key data can be accomplished with the describe function. By default, Python returns the measures with about six decimal places. Python has a nice function to round numbers to the exact decimal place you would like. To do this, just add .round() at the end of the function and insert how many decimal places you want inside the parentheses.

Hint for continuous variables: df3.describe().round(2)

***Task #2: Use the describe() function to describe the categorical variables.***

The dataframe contains categorical variables. Categorical variables contains strings, such as text. With categorical variables, you can get descriptive information such as the mode, frequency, and number of unique value counts. Obtaining these key data can be accomplished with the describe function. You also use the describe function as you would for continuous variables, but to get Python to understand that you only want objects (which are categorical variables), you have to tell it to include='object' inside the parenthesis in the describe function.

Hint for categorical variables: df3.describe(include='object')

***Task #3: Analyze, interpret, and write about the continuous and categorical variables.***

In this task, write a brief interpretation of the data. Comment about the continuous and categorical variables. In the code section, create a separate line that begins with a #. This is a comment line. Limit each line to about 20 words and start a new line with a #. It should look something like this:

```
# This is the first line.  
# This is the second line.  
# This is the third line.
```

**Deliverable Seven: Write the dataframe to a CSV file, save it, and verify it as a CSV file.**

[Click here to go back to Table of Contents](#)



Deliverable seven continues the process of exploring business data types and deal with column names that do not make sense or are incorrect. The objectives of deliverable two are to 1) Explore business data for insights, and 2) Synthesize concepts to structure data to allow for business analysis.

### **Deliverable Seven Tasks:**

Task 1: Using the `to_csv` function, write dataframe `df3` to a new `df4.csv` file.

Task 2: Check the `C:\Users\YOUR COMPUTER NAME FOLDER` to see if `df4` is there.

Task 3: Assign a new dataframe and load the CSV file back into the Jupyter Notebook using the `pandas` package.

Task 4: Show the completed dataframe.

#### ***Task #1: Using the `to_csv` function, write dataframe `df4` to a CSV file.***

One of the nice things one can do with Python is to save a dataframe as a CSV file and work with it in another program or save it for a later date to be imported back into Python (as required in this deliverable). It is relatively simple to write a dataframe to a CSV file. When writing to a CSV file, the `to_csv` function gives one an option to include the index found in the dataframe. By placing the `index=False` option after the name that you give to the file, it will save the CSV file without an index.

Hint: `df3.to_csv('df4.csv', index=False)`

#### ***Task #2: Check the `C:\Users\YOUR COMPUTER NAME FOLDER` for `df4`.***

If you see `df4` in the folder, it saved correctly, but you should verify it as instructed in the next task.

#### ***Task #3: Assign a new dataframe and load the CSV file back into the Jupyter Notebook.***

It is a good habit to verify that the newly created CSV file opens in Python. This is relatively easy to do with `Pandas`. Use the `read_csv` function to import the file back into Python.

Hint: `df4 = pd.read_csv('df4.csv')`

#### ***Task #4: Open the newly created CSV file in the Jupyter Notebook.***

Hint: `df4`

**[Click here to go back to Table of Contents](#)**

## **Deliverable Eight: Upload file into your GitHub Account.**


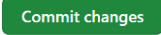
The objective of deliverable eight is to enable one to learn how to share Python code that contains business data with different audiences.

### **Deliverable Eight Task:**

Task 1: Open your GitHub account and upload the newly created CSV file.

Task 2: Validate that it has been uploaded and can be opened by emailing it to yourself and your instructor.

#### ***Task #1: Open your GitHub account and upload the newly created df4.csv file.***

Sign into your GitHub account and go into your repository that contains your newly created CSV file. Click on the  icon and choose upload files. Once you choose your file from the C:\Users\YOUR COMPUTER NAME FOLDER, in GitHub make sure that you scroll down the page and hit the  icon to complete the upload. The file should now be in your repository to share with the world. To share the file, click on the file name and copy the URL.

#### ***Task #2: Validate that it has been uploaded and can be opened by emailing it to yourself and instructor.***

If you intend to share the file with others, it is good practice to copy the URL and email it to yourself and open the file. For this task, email it to yourself and cc your instructor.

## **Deliverable Nine: Make a short video explaining deliverables one through nine.**

The objective of deliverable nine is to enable one to learn how to present business data analyzed through Python to make decisions.

### **Deliverable Nine Tasks:**

Task 1: Rehearse, record, and present a video presentation that clearly shows how you processed deliverable one through eight.

Task 2: Upload the video presentation or provide a link via the institution's course learning management system (Blackboard).

#### ***Task #1: Rehearse, record, and present a video presentation.***

An important part of business analytics is not only to present information in an easy to understand format, but to present how information is processed. It is important to rehearse the presentation a few times before recording. The recording should be as concise as possible so the [Click here to go back to Table of Contents](#)

file size is manageable. Try to keep in under five minutes. You can present it any medium as long as it can be opened by the instructor. Some suggestions are to use a narrated powerpoint, zoom, Canva, Snag It, Camtasia, etc. You can upload it into YouTube, Panopto or other location that you can easily share and play video files.

To assist students in developing a narrated PowerPoint, consider the following video examples:

- How to Record Narration for a PowerPoint Presentation For Dummies:  
<https://www.youtube.com/watch?v=QZp3jumnWUg>

### ***Task #2: Upload the video presentation.***

Upload the video or provide a link to the video in Blackboard.

### **Deliverable Ten: Write a reflection.**

The objective of deliverable ten is to enable one to reflect with a critical review of what you learned in the course and how Python can build your skill set.

### **Deliverable Ten Tasks:**

Task 1: Write a reflection in the Python file.

### ***Task #1: Write a reflection in the Python file.***

Students write a reflection with a critical review of what you learned in the course and how Python can build a desired skill set. Discuss your perception about Python and programming and how it may be able to help you better understand abstract business information and develop your skill set. Comment on whether or not Python is interesting and why. Also comment on your plans to study Python further.

**[Click here to go back to Table of Contents](#)**

## Appendix A, Grading Rubrics

Deliverables Grading Rubric			
Levels of Achievement			
Criteria	100 pts = Complete	50 pts = Incomplete	0 pts = No Attempt
<b>Deliverable One</b> Open a new Jupyter Notebook and load CSV file into the Jupyter Notebook.	10	5	0
<b>Deliverable Two</b> Explore rows with blank values and remove the row(s).	5	2.5	0
<b>Deliverable Three</b> Explore the data types and correct any errors.	5	2.5	0
<b>Deliverable Four</b> Explore the NaN values and remove the row with any NaN values.	5	2.5	0
<b>Deliverable Five</b> Change a column name.	5	2.5	0
<b>Deliverable Six</b> Describe the continuous and categorical data.	15	7.5	0
<b>Deliverable Seven</b> Write the dataframe to a CSV file, save it, and verify it as a CSV file.	5	2.5	0
<b>Deliverable Eight</b> Upload file into your GitHub Account.	10	5	0
<b>Deliverable Nine</b> Make a short video explaining deliverables one through nine.	20	10	0
<b>Deliverable Ten</b> Write a reflection.	20	10	0

[Click here to go back to Table of Contents](#)