AIM 5001 Module 11 Assignment (100 Points)

Part 1: Tidying and Reshaping Data

		Los Angeles	Phoenix	San Diego	San Francisco	Seattle
ALASKA	on time	497	221	212	503	1,841
	delayed	62	12	20	102	305
AM WEST	on time	694	4,840	383	320	201
	delayed	117	415	65	129	61

Source: *Numbersense*, Kaiser Fung, McGraw Hill, 2013

The chart above describes arrival delays for two airlines across five destinations. Its content has been re-created within the provided **M11_Data.csv** file. Get started as follows:

- Upload the provided **M11_Data.csv** file to your online AIM 5001 GitHub repository.
- Using the **pd.read_csv()** function, read the **M11_Data.csv** file from your GitHub repository into a Jupyter Notebook WITHOUT removing any empty rows or columns from the content of the file. The content of the resulting dataframe should appear as follows:

	Unnamed: 0	Unnamed: 1	Los Angeles	Phoenix	San Diego	San Francisco	Seattle
0	Alaska	on time	497.0	221.0	212.0	503.0	1841.0
1	NaN	delayed	62.0	12.0	20.0	102.0	305.0
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	AM West	on time	694.0	4840.0	383.0	320.0	201.0
4	NaN	delayed	117.0	415.0	65.0	129.0	61.0

1.1 (**30 Points**): Use your knowledge of combining and reshaping data in Pandas to tidy and transform/reshape the data contained within the dataframe. To get started, think about how you would want the data to appear if it were converted to "long" format, e.g., how would you define a "single observation" for the data shown in the graphic?; How many key values are associated with each data value?; How many columns should your long format structure contain based on the information provided in the graphic shown above?; What would the column headings for the long structure be?; etc. Use your answers to these questions to guide your reshaping/transformational work on the data. **Your reshaping/transformational steps must include converting the above table to a "tidy" long format.** Additional transformational steps (e.g., filling in missing

data values, renaming columns, etc.) should be performed as needed to ensure that your data is, in fact, "tidy".

1.2 (15 Points) Using your reshaped/transformed data, perform analysis to compare the arrival delays for the two airlines. Two questions you **must** answer: For each city, which airline had the best on time performance?; Which airline had the best **overall** on time performance?; etc. Feel free to define and answer additional questions of your own choosing,

1.3 (15 Points) Finally, given your "tidy" long format structure, consider what, if any, changes you would make to the visual presentation of the data if you were then asked to transform your "long" data back into a "wide" format: would you mimic the structure of the graphic shown above? If not, how might you transform your "long" data to "wide" format to make its "wide" presentation easier to understand and work with? Provide an example of your recommendation.

Part 2: Using Your GroupBy and Data Aggregation Skills

Three Short Coding Challenges
Can you complete these three tasks using no more than 17 lines of code in total?

These coding challenges will give you a chance to exercise your **GroupBy/Aggregation/Split-Apply-Combine** skills based on your readings from Chapter 10 of the "**Python for Data Analytics**" textbook. See if you can answer these three questions using **no more than *17 total lines of Python code***.

For each of the three questions you will be making use of the Pittsburgh Bridges data set: https://archive.ics.uci.edu/ml/datasets/Pittsburgh+Bridges. (Links to an external site.) Upload the provided **briges.data.version1.csv** to your online AIM 5001 GitHub repository and then read the file from GitHub into your local Python environment

2.1 (**12 Points**): You've been asked to generate a quick report that tells us how many bridges of each '**Purpose'/'Material**' grouping within the data set have been constructed over each of the rivers listed in the data set. **For each river**, your output should include the Purpose, Material, and count (aka 'How Many?'), similar to the output shown in the graphic below for River 'A', and **your report should include similar content for each of the rivers** contained within the data set.

			How Many?
River	Purpose	Material	
A	AQUEDUCT	IRON	1
		WOOD	3
	HIGHWAY	?	1
		IRON	2
		STEEL	21
		WOOD	8
	RR	IRON	1
		STEEL	9
		WOOD	2
	WALK	STEEL	1

You are allowed to use **no more than three (3) lines** of Python/Pandas code to generate this report in its entirety (i.e., you **MUST** produce the results for all of the rivers at once) and you **MUST** use Pandas' groupby and/or aggregation functionality to accomplish the task. **Be sure to include a brief narrative explaining how your proposed code would accomplish the task.**

2.2 (14 Points): You've been asked to generate a second report that shows the average length for each 'Purpose'/'Material' bridge grouping within the data set. As you should recall from our previous work with the Pittsburgh Bridges data set, the 'Length' attribute is not provided to us in a numeric format and also contains many missing values. As such, you should clean up the contents of that column and convert it to numeric format before attempting to generate your report. The output of your report should appear as shown in the graphic below.

		Average Length
Purpose	Material	
AQUEDUCT	IRON	1000.000000
	WOOD	1092.000000
HIGHWAY	?	NaN
	IRON	1216.666667
	STEEL	1557.804348
	WOOD	1053.375000
RR	IRON	1100.000000
	STEEL	1946.850000
	WOOD	NaN
WALK	STEEL	NaN

You are allowed to use **no more than four (4) lines** of Python/Pandas code ***AFTER*** you've finished cleaning up the 'Length' column (which should take no more than 2-3 lines of code) and you **MUST** use Pandas' groupby and/or aggregation functionality to accomplish the task. **Be sure to include a brief narrative explaining how your proposed code would accomplish the task.**

2.3 (14 Points) Finally, you've been asked to generate one last report that shows the average length, count, minimum length, and maximum length of bridges built during 4 equal length time periods (1818 – 1860; 1860-1902; 1902-1944; 1944-1986). The output of your report should appear as shown in the graphic below.

Erected	Average Length	Count	Max Length	Min Length
(1818.0, 1860.0]	1094.625000	8.0	1500.0	990.0
(1860.0, 1902.0]	1603.347826	23.0	4558.0	1000.0
(1902.0, 1944.0]	1676.181818	33.0	3000.0	860.0
(1944.0, 1986.0]	1530.411765	17.0	3756.0	804.0

You are allowed to use **no more than seven (7) lines** of Python/Pandas code and you **must** use Pandas' groupby and/or aggregation functionality to accomplish the task. **Be sure to include a brief narrative explaining how your proposed code would accomplish the task.**

Save all of your work for this assignment within **a single Jupyter Notebook** and upload / submit it within the provided M11 Assignment Canvas submission portal. Be sure to save your Notebook using the following nomenclature: **first initial_last name_M11_assn**" (e.g., J_Smith_M11_assn).