TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Assignment 2

This is the assignment 2 of JBI010 2023-2024. In this assignment you will need to work in pairs and apply the programming skills you have acquired so far in the course. To ease the cooperation within the pair and keep track of your contributions, you will be using Git as version control system and GitLab as code repository. Remember to apply computational thinking every time you are solving a problem—that is, identify and understand the problem, inputs, output, and the steps you need to follow to go from the inputs to the output. The assignment must be developed exclusively by the members of the pair. One last piece of advice: Have fun!

1 Beer World

In the vast world of alcoholic beverages, beer stands out as one of the oldest and most beloved drinks. Its rich history has contributed to its enduring popularity, captivating both seasoned enthusiasts and newcomers alike. The beer market has been experiencing significant growth, with the global craft beer industry reaching a staggering size of $95.2 billion in 2020 [1]. Experts predict that this upward trajectory will continue, projecting the market to reach $210.8 billion by 2028.

As consumer demand for unique and innovative beer products continues to rise, breweries around the world are fervently striving to develop new offerings. This increasing competition has led to a complex and intense market landscape. In the United States of America, the number of breweries has skyrocketed in recent years: In 2010, there were only 1,800 breweries while in 2015 more than 4,800 were funded [2].

One of the driving factors behind beer's widespread appeal is its incredible range of flavors and styles. This diversity has made beer a favored choice among new generations, who are eager to explore craft beers. To facilitate understanding and categorization, the Beer Judge Certification Program (BJCP) has established guidelines for classifying beers based on their production processes, flavors, and other characteristics. These classifications help consumers and industry professionals navigate the wide array of beer styles and appreciate their unique qualities.

Each beer is classified within a style, and each style is also classified within a category. The beer styles are closely tied to the countries with rich brewing traditions [1]. Nations

[1] https://www.fortunebusinessinsights.com/industry-reports/craft-beer-market-100736

[2] https://www.brewersassociation.org/statistics-and-data/national-beer-stats/

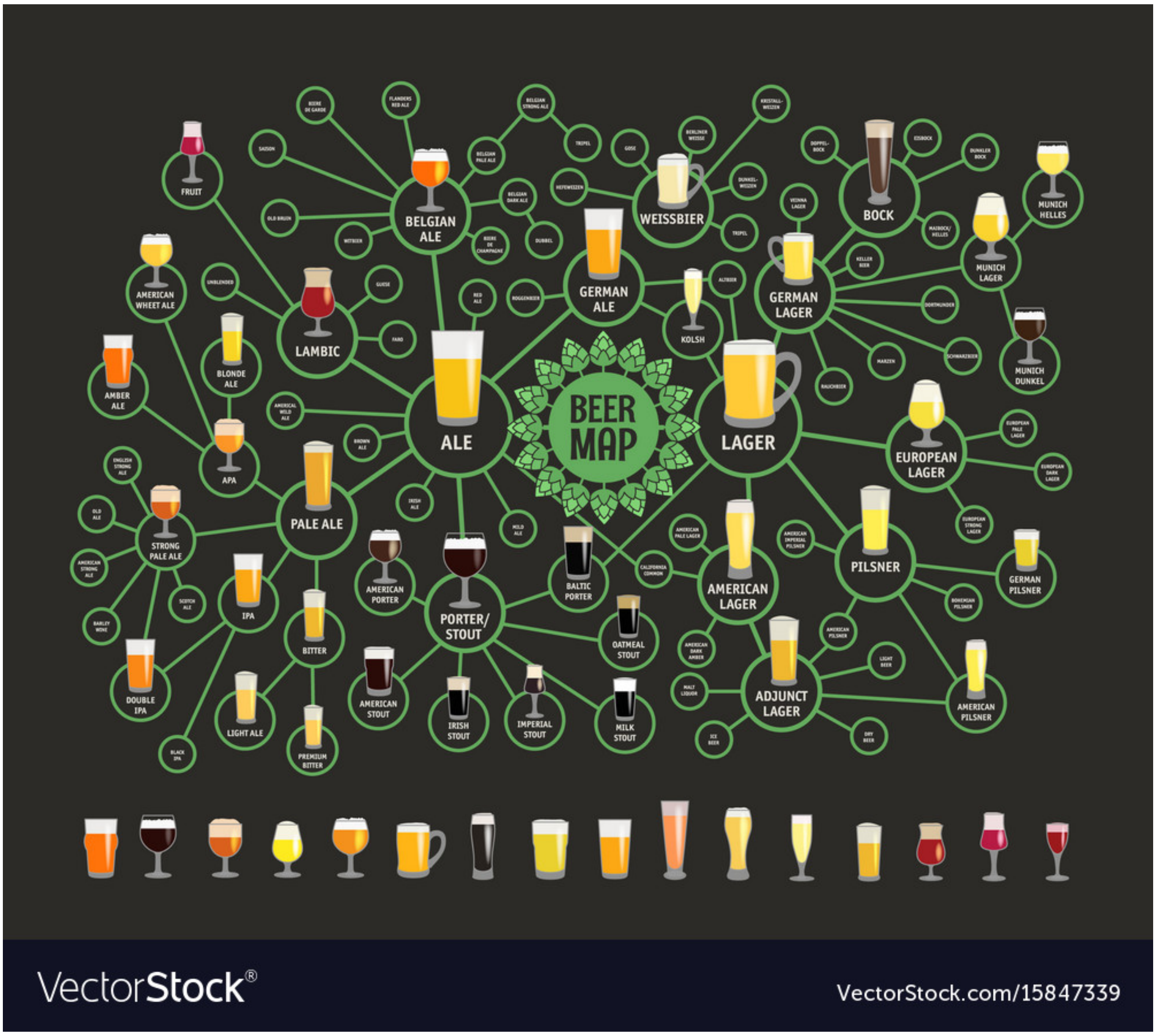

Figure 1: Map of different beers styles. Source: VectorStock.

such as Germany, Belgium, the Netherlands, the United Kingdom, Austria, and more recently, the United States of America, have made significant contributions to the art of brewing, resulting in a wide variety of styles. For example, Pils style was originated in Germany, while Porter and India Pale Ale (IPA) were originated in the United Kingdom.

Due to the market growth and complex industry, aspiring craft breweries face the challenge of staying competitive [2]. To achieve success, they must rely on robust data and data analysis methodologies to identify consumer trends, optimize production processes, and make strategic decisions that enable them to stay competitive in a rapidly evolving craft beer industry. Data on market sales, beer styles, breweries, and countries of origin becomes critical for new competitors to make informed decisions and participate in the market. However, it is not just the industry insiders who seek a better understanding of the beer market. A growing number of consumers, particularly those with a penchant for exploration and a desire to discover new flavors and styles, are keen on acquiring in-depth knowledge about the beer industry. They yearn to broaden their horizons and make well-informed choices when it comes to indulging in the world of beer.

Meet Josh, one of these beer enthusiasts who is curious to learn more about the beer and brewery market. Josh is determined to delve into the available data, seeking valuable insights that can enhance his understanding of this dynamic industry. In this assignment, you will assist Josh find insights providing him with analysis and information based on the dataset described in Section 1.1 in different tasks.

1.1 Dataset

This section presents the description of the beer breweries dataset. This dataset is a modification of the datasets of Craft Beer Data [3] The dataset is stored in your `assets` folder under the name “beer_db_v4.csv”. The type and description of each column can be found in Table 1. This dataset also includes information about categories and styles of the beer. This data can be found in the BJCP webpage [4].

2 Tasks

In this section, we introduce the tasks that you will develop together with your partner. There are five tasks in total. It is important to **follow conding style conventions, provide a docstring for every implemented function, specify the appropriate type hints for both parameters and return types, and provide at least two pytests for each implemented function.** Otherwise, some points might get deducted from your grade. Each task has its TODO section within the project template. These sections are labeled in the same way as your weekly exercises, namely:

[3] `https://data.world/brettcarpenter/craft-beer-data`

[4] `https://www.bjcp.org/bjcp-style-guidelines/`

Table 1: Description of the Beer World dataset.

Column name	Type	Description
beer_id	int	The id of a beer.
beer_name	str	The name of a beer.
brewery_id	int	The id of a brewery, that produces the beer.
brewery_name	str	The name of a brewery, that produces the beer.
cat_id	int	The category id of a beer.
cat_name	str	The category name of a beer.
styleId	str	The style id of a beer.
style_name	str	The style name of a beer.
abv	float	Alcohol by volume (ABV) of a beer—that is, the number of milliliters of alcohol per 100 milliliters in a beer.
address1 & address2	str	Address information about a brewery.
city	str	City where a brewery is located.
state	str	State where a brewery is located.
code	str	Postal code of the brewery.
country	str	Country where the brewery is located.
phone	str	Phone number of the brewery.
website	str	URL to the website of the brewery.
<Month>_<year> (e.g. January_22)	float	The amount of beer in tons that the brewery produced that month.

```
#// BEGIN_TODO [task_<Nr>] Task title

#// END_TODO [task_<Nr>]
```

Ensure all your code is written within such markers, otherwise the graders won't be able to find it.

2.1 Task 1: Version Control

Version control systems help tracking and managing changes performed to software project artefacts. They are of special importance when working within development teams, where programmers and other members need to introduce changes to the projects. A version control system then eases the coordination, sharing, and collaboration within the team. An example of such systems is Git, an open source distributed system that has gain vast popularity in the last years. In Git, there is usually one or more remote servers containing all snapshots of the code. Additionally, all programmers within the team can have a clone of the project in their own machines, which is a complete mirror of the project.

Nowadays, there are different cloud providers hosting Git and offering additional features that support parts of the software development cycle, e.g. GitLab, GitHub, and BitBucket. Some of these additional capabilities include (but are certainly not limited to) issue reporting, continuous integration and deployment, code reviews, among others. In this course, we use a TU/e GitLab instance as our Git provider. You and your partner will coordinate and use this service to speed up the collaboration and development process.

(a) Create (at least) one issue for each of the subsequent tasks (tasks 2, 3, 4, and 5). You are completely free to choose the level of granularity for the definition of the issues. We suggest you to identify atomic subtasks within the assignment tasks, supporting a more efficient tasks distribution and planning.

(b) All members of the team need to push their commits frequently (and, ideally, evenly) to the repository. Ensure commit messages are properly written, that is, start with a verb (in the infinitive form) and give a short, concrete, and clear message about the change. For example, "Compute average of beers alcohol volume". You can see that "compute" is a verb in the infinitive (not "computed", not "computing", etc.), and the rest introduces a short, concrete, and clear message about the change. You can also consider using conventional commits. For more information visit `https://www.conventionalcommits.org/en/v1.0.0/`.

(c) Close each issue once you are done with its implementation. You can automatically close an issue with a commit by writing one of the predefined keywords, namely, "close", "closes", "closed", "fixes", or "fixed", followed by the character "#" and the ID or number of the issue (e.g. "Close #10"). You can also manually close the issue via GitLab.

2.2 Task 2: Data Preparation

The read_dataset function will be used to load in our data and prepare it. The dataset that we wish to load is called beers_world.csv. In the data_loader.py file, write the read_dataset function, which opens and reads the dataset file given a path to the dataset. Based on the content of the file you should return a list of dictionaries, where each dictionary represents a row in the dataset. The keys of the dictionary must be strings with the column name (as it appears in the file) and the values are either strings or numeric values. You would then expect to have **one** dictionary in the form[5]:

```
{
        "column_1": "value_1",
        "column_2": 2.0,
        "column_3": 3,
        ...
}
```

Apart from plainly loading the file and creating the expected output, you are to undertake the following actions:

- Ensure proper naming of columns.
  - The naming style of the header styleId does not match the style used for the other headers. To ensure consistency, please change the column name from camel case (styleId) to snake case (style_id). It is important to reflect this change in the dictionaries.
  - There are several columns associated with the breweries' addresses. However, there is one particular column with a name that doesn't explicitly convey its purpose as an address component. The current name is quite generic and could be misinterpreted. Therefore, it is necessary to rename this column to indicate that it is indeed related to the address. If the new name consists of multiple words, please ensure to use snake case. It is important to update this change in the dictionaries as well.
- Remember to give the right data type to the dictionary values.
- Add unit tests to test your implementation in the data_loader_tests.py. First, think about what you want to test and then implement your code.
- The file is encoded in **UTF8**, you will likely need this information to load the file correctly. Investigate what UTF means and why setting the format is important when reading a file. Write down your findings in the README file.

[5] Beware that these are dummy values and are not expected to come like this from the dataset.

Notes:

- The `DictReader` from the CSV module can be handy in solving this exercise.

2.3 Task 3: Data Exploration

In order to explore the Beer World data, we will create several functions that serve different analyzes. Your task is to define these functions in the file named `data_exploration.py`. Each function will be provided with a name and an informal description of its (ordered) parameters and output. Don't forget to follow the coding conventions, write meaningful docstrings, declare type hints, and write at least two unit tests per function.

(a) Josh was born and raised in Franklin County in Ohio. He is curious about what styles of beer his home state produces, and how many beers of each style. Create the function `get_state_style_count` that, given the data (as returned by the `read_dataset` function) and a state, returns a dictionary where each style name is a key with the count as its value. The result should also include the styles with count equal to 0.

(b) As a beer enthusiast, Josh wants to use some of his vacation days to visit a brewery. However, he would like to make sure that his best half, who is not particularly fond of all beer types, will also enjoy the experience. To this end, Josh plans to choose a brewery that offers blonde Ales, their favorite type of beer.

- Create a function `get_breweries_from_style` that, given the data (as returned by the `read_dataset` function) and a substring of a beer style, returns a list of dictionaries with information about breweries that have beers with style containing the given substring. For example, we can use this function by passing ”Blonde” as the substring. Lower and upper case should not be strictly consider—that is, for both ”Blonde” and ”blonde” you should return the same matching cases.
- The brewery information in each dictionary should include the brewery name, ID, and columns referring to the address, city, state, country, and postcode.
- To do this, you will need to iterate through a list of beers and generate a list of breweries that meet the given criteria. However, since a brewery can have multiple beers that satisfy the criteria, simply adding a brewery to the output list each time a matching beer is found could result in duplicates. Therefore, it is important to ensure that your function avoids this by handling duplicates appropriately.

(c) Since Josh would love to experience the Californian weather, he decides to travel to California for his trip. He has compiled a list of breweries that offer Blonde Ales. Now, his curiosity leads him to inquire about the alcohol content of the beers available at these breweries. Specifically, he wants to calculate the average alcohol percentage of the beers for each individual brewery. First, create the function `compute_mean_abv` that, given a list of dictionaries with beer information (the keys are the ones that appear in the initial data you were given), computes the mean alcohol content of these beers and outputs it as a float.

(d) Create an auxiliary function `get_breweries_from_state` that, given a list of dictionaries with brewery information (whose keys are described in Task 3b) and a string indicating a state, returns a list of dictionaries only retaining the breweries that are located in the state specified in the input. You can use this function to filter the output of Task 3.b.

(e) Create an auxiliary function `get_beers_from_brewery` that, given the data and a brewery ID, returns a list of dictionaries with the beers belonging to the specified brewery. These dictionaries need to contain all the keys present in the initial data.

(f) Finally, create a function `add_mean_abv_state_breweries` that takes as input the initial data, the output of the function you created in Task 3b (a list of breweries), and a string representing a state. The function must add to each brewery dictionary a field `mean_abv` with the average alcohol content of each of the breweries made in the given state.

(g) Josh chose to calculate the *mean* alcohol percentage in order to get an idea of the alcohol content of the beers of each brewery. This might not always be the best statistic to go off of. Explain what can cause the mean to not be reliable measure. What other statistic can you use to avoid this pitfall? Write your answer in the `README` file.

2.4 Task 4: Further Analysis

In Task 3, you were asked to perform specific analyses. This will not be the case in Task 4. Here, you are free to explore what you desire. **Choose one additional analysis you would like to perform and implement it in your code.** Of course, the chosen analysis needs to aim at answering questions outside the regular exploratory data analyzes; it need to serve the purpose of understanding better an aspect of the domain. Don't forget to follow the coding conventions, write meaningful docstrings, declare type hints, and write at least two unit tests per function.

- In the `README` file, write an introduction to the question you want to answer with your analysis. Explain why this analysis is interesting to you and what sort of problem needs to solve.
- Write the code to perform this analysis in the form of functions. Unit tests and documentation should also be provided.

- In the README file, document where in the project your code has been implemented and the design choices you made. Remember to document the expected input and output of each function you have defined.

If you are struggling with creativity, here are two analyses you can choose from to implement this part of the assignment.

- Explore the distribution of breweries and beer styles across different countries or states. Calculate the number of breweries and the most common beer styles for each state/country. Identify states/countries that have a high concentration of breweries or unique beer styles.
- Analyze the breweries based on certain criteria such as the number of beers they offer, their mean or median alcohol content, or the total beer production (in tons) in 2022. Create a function that ranks the breweries according to these criteria and displays the top-ranked breweries.

2.5 Task 5: Bonus

Josh wishes to have a nice application that displays brewery data by country. For this purpose, we will be using Dash. Dash is a framework that helps in the creation of interactive data visualization applications written in Python. Buttons, dropdowns, and other user interaction elements can be added to the page, as well as the Plotly graphs (see `https://dash.plotly.com/`). In the first bonus subtask, you won't need to write any Dash code, it has been done for you! However, you can investigate a bit more in the second subtask.

In `main__.py`, change the Dash mode variable to `True`. If you now try to execute your code, it should return a different output. Most notably, you will see a message along the lines of "Dash is running on ...", where the latter will be a hyperlink. Follow this hyperlink to open the dashboard in your browser. The dashboard is set up to run in debug mode, meaning that if you make any changes to your code the dashboard will automatically refresh itself without the need to rerun your code. Occasionally, you might find that during coding you cause an error that the application cannot recover from, in which case you do have to rerun your code after fixing the error.

(a) In `dashboard.py`, go the the `update_bar_chart` function and in the TODO section here write a line of code that assigns a Plotly bar chart to the fig variable.

(b) In `dashboard.py`, go the the `update_cat_median_abv` function and in the TODO section here write a line of code that gets the key with the highest value in the dictionary `cat_median_abv_dict` and assigns it to the output variable.

Notes:

- Use the `item_data` variable, which is the data that you have prepared in the previous tasks stored in a data format that Dash likes to work with.
- To create a bar chart, you just need to use the following code: `fig = px.bar(item_data, x='cat_name')`.
- The usual print function does not work in Dash! In order to print things in console when you are running Dash you must use `app.logger.info()`.

References

[1] Charles Papazian. Beer Styles: Their Origins and Classification. *Food Science and Technology*, 157:39, 2006.

[2] Neil Reid and Jay D Gatrell. Craft Breweries and Economic Development: Local Geographies of Beer. *Polymath: An Interdisciplinary Arts and Sciences Journal*, 7(2):90–110, 2017.