In this lesson's assignments, we will analyze data about American eateries. We will look for where to buy tacos in California, find out in which months restaurants were most often opened, and explore the potential pitfalls of working with pandas.

Data Description

- **id**: Unique identifier for the establishment.
- **address**: Physical address of the establishment.
- **categories**: Categories of the establishment (e.g., "Fast food restaurant").
- **city**: The city where the establishment is located.
- **cuisines**: Cuisines served at the establishment (e.g., "Mexican").
- **dateAdded**: The date the record was added. We will assume this is the opening date of the establishment.
- **dateUpdated**: The date the information about the establishment was last updated.
- **latitude**: Geographic latitude.
- **longitude**: Geographic longitude.
- **menus.category**: Food categories in the menu.
- **menus.currency**: The currency in which payment is accepted.
- **menus.dateSeen**: The date when this menu was recorded.
- **menus.description**: Description of the menu from the establishment.
- **menus.name**: The name of the menu.
- **name**: The name of the establishment.
- **province**: The state where the establishment is located.

Tasks

Task 1: List the first three values of the `dateAdded` column, separated by a comma and a space, in the order they appear. For example: `2000-01-01T00:00:00Z, 2001-01-01T00:00:00Z, 2002-01-01T00:00:00Z`

Task 2: Let's check the data types of the `city` and `latitude` columns.

Task 3: Which columns will be displayed when the `describe()` method is applied?

Task 4: What are the mean values for the columns that `describe()` returned in the previous task?

Task 5: Currently, there are missing values in the data. Let's remove them.

Task 6: Filter for establishments located in the city of 'California', Missouri. Provide the DataFrame indices for these establishments.

Task 7: Filter for 'Taco Bell' establishments in the city of 'California', Missouri. Provide the indices for these establishments.

Task 8: Let's find establishments that are either 'Taco Bell' or are located in New York City. It is required that the menu name does **not** contain 'Volcano Taco' or 'Fresco Soft Taco'. Save the resulting data into a DataFrame named `result`.

Task 9: Let's select restaurants where the payment currency value is not missing. Save the data into a resulting DataFrame named `result`.

Task 10: Try running `data['categories']` and `data[['categories']]`. Observe the type and the values of their elements.

Task 11: List the top 5 cities with the highest number of records, separated by a comma and a space.

Task 12: Count how many 'Taco Bell' restaurants are in each city. Select the 5 cities where this restaurant appears most frequently. Save the number of restaurants for these cities along with their names as a `pd.Series` object into the variable `result`.

Task 13: Let's find restaurants that opened in October. Provide the indices of the first five restaurants, separated by a comma and a space, in ascending order.

Task 14: Group the data by the month the restaurant opened (the `dateAdded` variable). How many restaurants were opened each month? Calculate this based on the count of unique restaurant IDs for each month. The `dateAdded` values should be the index. Save the resulting DataFrame in `result`.

Task 15: In this task, you need to add a column named `update_delta` with the number of whole days calculated from the difference between `dateUpdated` and `dateAdded`. Then, for each city, find (in the specified order):

- The mean of the `update_delta` variable.
- The latitude of the northernmost eatery (the more northern the eatery, the higher the latitude value). The city names should not be the index. Save the resulting DataFrame into the variable `result`. Also, save the mean `update_delta`for the city 'Zephyrhills' into the variable `zep_mean`.

Task 16: Select restaurants that have 'Pizza' mentioned in their `categories` column. Provide the index of the 3rd element from the filtered results.

Task 17: In this task, you again need to add a column `update_delta`, which will store the difference in *days* between `dateUpdated` and `dateAdded`. Calculate the mean and median of this new column. List the values separated by a space (mean first, then median), rounded to 2 decimal places.

Task 18: For this task, you need to select records that have more than 20 categories. Then, group them by province and calculate the minimum `longitude` in each. Round the `longitude` values to 3 decimal places. Next, save the resulting DataFrame to a CSV file using a semicolon (`;`) as the separator. The DataFrame itself should contain two columns: `province` and `longitude`.