Project Description

Abraham Cepeda

U75425818

MET CS550 A1

Computational Mathematics for Data Analytics

Project Description: Mayan Calendar

The objective of this project is to fully understand the main components of the Mayan Calendar and see how they can relate to our Gregorian Calendar. Therefore, before going into coding the python functions, it was crucial to analyze how the Mayan Calendar works and the different components it takes into account.

After researching, it was clear that the Mayan Calendar has two main components. The first one is known as the "Long Count," a system to count the number of days that have passed since an arbitrary initial date, believed to be Wednesday 13, August 3114 BC. This system works similarly to binary numbers, where each position represents a base number related to the power of 20. However, the second position has a base of 18. Therefore, the order of the numbers would be as follows: $20^0, 20^1, 20^1 * 18, 20^2 * 18, 20^3 * 18$ ... On the other hand, the second part is known as the Calendar Round. This component of the Mayan Calendar is further divided into two sections called the Tzolkin and the Haab. The first is a spiritual and energetic calendar composed of 260 days divided into 13 20-day months, while the Haab represents the 365 days it takes the Earth to rotate around the Sun. The Haab has 18 20-day months and a 5-day period called "Wayeb" to complete the 365 days.

After comprehending how the Mayan Calendar functions, it was time to develop some python functions to represent different types of calculations. Hence, six python functions were developed.

Function 1

- Name: days_since_long_count
- Parameters: lst → string (“0.0.0.0.0”)
- Returns: days → integer
- Example: (“9.8.9.0.0”) → 1356840
- Description: Calculates the number of days that have passed since the initial date given a long count date.

Function 2

- Name: days_to_long_count
- Parameters: days → integer (Any number greater than 0)
- Returns: lst → array ([0,0,0,0,0])
- Example: (1746840) → [12,2,12,6,0]
- Description: The function calculates the long count date based on the number of days given.

Function 3

- Name: haab_to_gregorian
- Parameters: haab → string (“5 Yaxkin”)
- Returns: num_days → integer (1-31), gregorian_month → string (“Jan”-“Dec”)
- Example: (5, “Yaxkin”) → (5, “May”)
- Description: This function converts a Haab date to its equivalent in the Gregorian Calendar.

Function 4

- Name: gregorian_to_haab
- Parameters: day → integer, month → string
- Returns: haab_day → integer, haab_month → string
- Example: (31,"December") → (5, 'Wayeb’)
- Description: This function converts a Gregorian date to its equivalent in the Haab Calendar.

Function 5

- Name: long_count_to_calendar_round
- Parameters: long_count → string (“0.0.0.0.0”)
- Returns: trecena → integer, veintena → string, haab_day → integer, haab_month → string
- Example: ("12.19.5.5.0") → (8, 'Ahau', 13, 'Sots’)

- Description: This function calculates the calendar round date (Tzolkin and Haab) based on the long count given.

Function 6

- Name: gregorian_to_mayan_date
- Parameters: day → integer, month → integer, year → integer
- Prints: long count date and calendar round date
- Example: (6,12,2022) → (13.0.10.1.17 1, 'Caban', 10, 'Mak')
- Description: This function calculates the complete mayan date (Long count and Calendar Round) from a Gregorian date.

Instructions to run code

- Open *ipynb* file in the Jupyter Lab (https://jupyter.org/try-jupyter/lab/) or any text editor that supports this type of file.
- At the end of the file all the functions are called to calculate its respective value with comments regarding the restrictions for the parameters of each function.