Project Name: Data-Wrangling With Date-Time Concept

Table of Contents

Demo

```
4
5  import datetime
6  import pytz
7
8  # Naive
9  # d = datetime.date(2001, 9, 11)
10
11 tday = datetime.date.today()
12
13
14 # weekday() - Monday is 0 and Sunday is 6
15 print(tday)
16
17 # isoweekday() - Monday is 1 and Sunday is 7
18 print(tday)
19
20
21 # datetime.timedelta(days=0, seconds=0, microseconds=0, milliseconds=0, minutes=0, hours=0, weeks=0)
22
23 tdelta = datetime.timedelta(hours=12)
24
25 print(tday + tdelta)
26
27 # date2 = date1 + timedelta
28 # timedelta = date1 + date2
29
30 bday = datetime.date(2016, 9, 24)
31
32 till_bday = bday - tday
33
34 print(till_bday.days)
35
36 t = datetime.time(9, 30, 45, 100000)
```

Overview

This is a Data Wrangling with Date and Time Concept.
Data wrangling involves processing the data in various formats like - merging, grouping, concatenating etc. for the purpose of analysing or getting them ready to be used with another set of data. Python has built-in features to apply these wrangling methods to various data sets to achieve the analytical goal.
This repository contains the code for Data Wrangling using python's various libraries.
It used pytz and datetime libraries.
These libraries help to perform individually one particular functionality.
Pytz allows accurate and cross platform timezone calculations.
Datetime module supplies classes to work with date and time.
The purpose of creating this repository is to gain insights into how to wrangle or transform data.
These python libraries raised knowledge in discovering these libraries with practical use of it.
It leads to growth in my ML repository.
This above screenshot will help you to understand flow of output.

Motivation

The reason behind building this is, I know that wrangling costs analytics professionals as much as 80% of their time therefore, to become ML Professional this is one of the skillsets necessary in bucket-list. One very common truth of life is that, when we hear proverbs or saying, we do not understand it much or completely rather we can absorb it fully when we go through it. Right. That means, here information from advice format to practical self-experience format got transferred and then we could understand it completely. Similarly, in Data Wrangling is the process of converting and mapping data from its raw form to another format with the purpose of making it more valuable and appropriate for advance tasks such as Data Analytics and Machine Learning. For companies, their clients are important and so for employee because its all about business and decisions taken for business. Data Wrangling exactly here comes for rescue as Data wrangling is the art of providing the right information to business analysts to make the right decision on time. Data wrangling also provides organisations with the right information in a short span of time to access the right information thereby helping make strategic decisions for the business. One of the most common steps taken in data science work is data wrangling to solve complex business problems.
One of the important goal of Data Wrangling is to drive better decisions based on data in short time span and that is what I am trying to achieve by doing this project and that is also one of the important qualities that employee should have. Performing data wrangling in right direction will rescue an organization 6 on 10 times from drowning. When analyst can reach to precise decision through this process then it is good deal. Hence, I continue to gain knowledge while practicing the same and spread literary wings in tech-heaven.

Technical Aspect

Python pytz module provides implementations of tzinfo class that can be used to create timezone aware datetime instances.

Reason for selecting pytz is because working with date-time concept is tricky and it is sometimes required in application development especially while working for global companies so this is simply trying to get along with it.

Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps. Working with dates and times is one of the biggest challenges in programming. Between dealing with time zones, daylight saving time, and different written date formats, it can be tough to keep track of which days and times you're referencing. Fortunately, the built-in Python datetime module can help you manage the complex nature of dates and times. For example, one great example of this irregularity is daylight saving time**.** In the United States and Canada, clocks are set forward by one hour on the second Sunday in March and set back by one hour on the first Sunday in November. However, this has only been the case since 2007. Prior to 2007, clocks were set forward on the first Sunday in April and set back on the last Sunday in October. Things get even more complicated when you consider time zones. Ideally, time zone boundaries would follow lines of longitude exactly.

Installation

Using intel core i5 9th generation with NVIDIA GFORECE GTX1650.

Windows 10 Environment Used.

Already Installed Anaconda Navigator for Python 3.x

The Code is written in Python 3.8.

If you don't have Python installed then please install Anaconda Navigator from its official site.

If you are using a lower version of Python you can upgrade using the pip package, ensuring you have the latest version of pip, *python* -m pip install --*upgrade pip and press Enter*.

Run/How to Use/Steps

Keep your internet connection on while running or accessing files and throughout too.

Follow this when you want to perform from scratch.

Open Anaconda Prompt, Perform the following steps:

cd <PATH>

pip install pandas

pip install pandas-ui

You can also create requirement.txt file as, pip freeze > requirements.txt

run files.

Follow this when you want to just perform on local machine.
Download ZIP File.
Right-Click on ZIP file in download section and select Extract file option, which will unzip file.
Move unzip folder to desired folder/location be it D drive or desktop etc.
Open Anaconda Prompt, write cd <PATH> and press Enter.
eg: cd C:\Users\Monica\Desktop\Projects\Python Projects 1\13)Data_Wrangling\Project_3_Data_Wrangling_with_datetimeConcept

In Anconda Prompt, pip install -r requirements.txt to install all packages.
In Anconda Prompt, write <filename>.py and press Enter. That is,
In Anconda Prompt, write 1)Data_Wrangling_with_datetimeConcept.py and press Enter.
Please be careful with spellings or numbers while typing filename and easier is just copy filename and then run it to avoid any silly errors.
Note: cd <PATH>
[Go to Folder where file is. Select the path from top and right-click and select copy option and paste it next to cd one space <path> and press enter, then you can access all files of that folder] [cd means change directory]

Directory Tree/Structure of Project

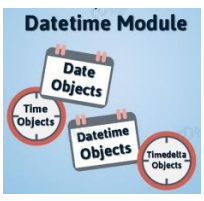
Folder: 13)Data_Wrangling > Project_3_Data_Wrangling_with_datetimeConcept
1)Data_Wrangling_with_datetimeConcept.py

To Do/Future Scope

Can take another dataset.

Technologies Used/System Requirements/Tech Stack

pytz Module



Credits

Corey Schafer Channel