Timedeltas & Date Ranges

Learning Objectives:

- Apply timedeltas to calculate changes between date/times.
- Create date_ranges of equally spaced intervals.

More Pandas Datetime Functions/Variable Types

- Pandas has other Objects for more advanced datetime functionality.
- See the Pandas user guide for the summary table of datetime objects, including:
  - Timedeltas: calculating a change in time.
  - date_ranges: creating equally spaced intervals

Panda's Timedeltas

- A time delta is a way to represent an increase or decreases in time. It is used primarily for calculating relative datetimes (e.g. "30 days before").
- Using `pd.to_timedelta` ([Documentation]) (https://pandas.pydata.org/docs/reference/api/pandas.to_timedelta.html):
  - First Argument is the number of time steps (e.g. 30).
  - Unit: the unit of time (e.g. "D").
    - Possible values:
      - 'W'
      - 'D' / 'days' / 'day',
      - 'hours' / 'hour' / 'hr' / 'h',
      - 'm' / 'minute' / 'min' / 'minutes' / 'T'
      - 'S' / 'seconds' / 'sec' / 'second'
      - 'ms' / 'milliseconds' / 'millisecond' / 'milli' / 'millis' / 'L'
      - 'us' / 'microseconds' / 'microsecond' / 'micro' / 'micros' / 'U'
      - 'ns' / 'nanoseconds' / 'nano' / 'nanos' / 'nanosecond' / 'N'

We will return to our Delhi weather dataset and convert the 'date' feature to a datetime column.

```
import pandas as pd
url="https://docs.google.com/spreadsheets/d/e/2PACX-1vQcpVvVioO23cndDwr1UmKhndrSq6ES6ZUKZ4fkBBqIAavd1_coVPO_yeOye-Ub-cAWlkX3psJvOU8o/pub?output=csv"
df = pd.read_csv(url)
df['datetime'] = pd.to_datetime(df['date'])
```

Now we will set a time delta of 3 days:

```
# make the time delta
delta_3d = pd.to_timedelta(3,'D')
delta_3d
```

```
Timedelta('3 days 00:00:00')
```

- Example Use:
  - For the most humid day in the weather data, what was the average wind speed over the 3 days prior and the 3 days after the most humid day?

```
max_date = df['humidity'].idxmax()
# calc 3 days BEFORE
pre_max = max_date - delta_3d
pre_max
```

```
Timestamp('2016-12-29 00:00:00')
```

```
# calc 3 days AFTER
post_max = max_date + delta_3d
post_max
```

```
Timestamp('2017-01-04 00:00:00')
```

Now that we have our pre and post dates, we can obtain average windspeed by setting those dates as the range and calculating the mean:

```
mean_windspeed = df.loc[pre_max:post_max,'wind_speed'].mean()
mean_windspeed
```

```
4.89791666675
```

Another option we can use is to store our range of dates as a variable using pd.date_range.

pd.date_range

Another option we can use is to store our range of dates as a variable using pd.date_range.

- pd.date_range Documentation

```
## making a date range to cover the pre-max to post-max window
date_range = pd.date_range(pre_max, post_max)
date_range
```

```
DatetimeIndex(['2016-12-29', '2016-12-30', '2016-12-31', '2017-01-01',
               '2017-01-02', '2017-01-03', '2017-01-04'],
              dtype='datetime64[ns]', freq='D')
```

Notice that we have a list of indices. When we attempt to run our calculation, we will get an error:

```
# this will give an error
df.loc[date_range,'wind_speed'].mean()
```

```
...
KeyError: "[Timestamp('2017-01-02 00:00:00'), Timestamp('2017-01-03 00:00:00'),
Timestamp('2017-01-04 00:00:00')] not in index"
```

Can you see what caused the error? It turns out that our date range goes beyond the index of our data.

This did not cause an error when just using .loc with our pre and post dates because using the range within .loc will just pull any dates that fall within the range, and not a list of each date in the range.

The difference is subtle, but understanding your options and the way each works will give you more versatility when writing code.

Summary

This lesson explored some of the advanced functionality of Pandas related to dates and times. You learned how to create time deltas and define a date_range.

Additional Resources

- For more information about working with DateTime in pandas and the different type of datetime variables, read the following:
    1. Date Time in Pandas: A Simple Guide for Beginners (2022)
    2. Pandas Doc Page: Time series/ date functionality

Previous

Next