12/12/2024

Business Data Analysis

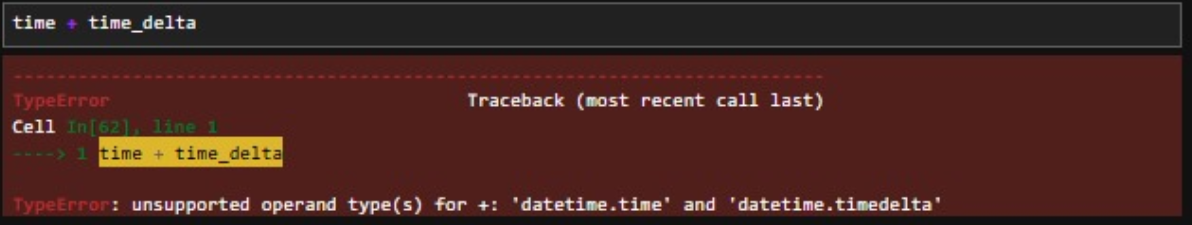
Errors of Chap 8,9,10

M.MOHIB IMRAN

sp23-bbd-035

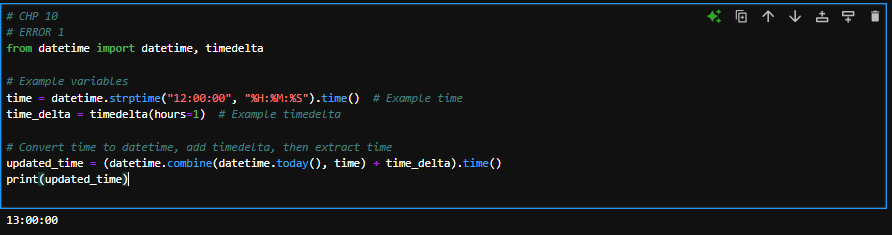
Errors of chap 10:

* Error 1



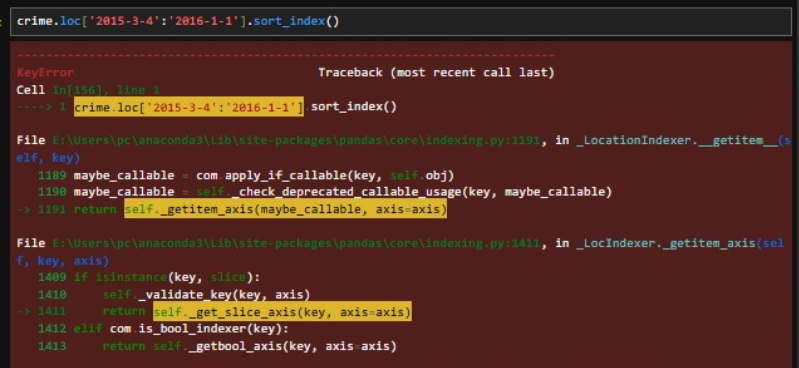
The error occurs because you are attempting to directly add a datetime.time object and a datetime.timedelta object, which is not supported in Python. To resolve this, you need to first convert the datetime.time object into a datetime.datetime object. This can be done using datetime.combine, which combines a date (e.g., today's date) with the given time. Once converted, you can add the timedelta to the datetime object. If you need only the updated time, you can extract it back using the .time() method. This approach ensures compatibility between the data types and avoids TypeError.

* Correct Code:



Changes:

* **Correct**: Convert time to datetime.datetime using datetime.combine() before adding time\_delta.
* **Correct**: Use .time() to extract the updated time after the addition.
* Error 2



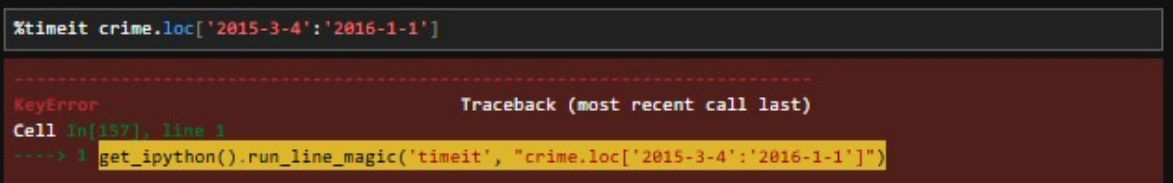
The error occurs because the .loc[] indexer in Pandas is being used with an invalid range for slicing. The issue likely lies in the format of the date strings or the structure of the DataFrame. To fix this, ensure that the DataFrame has a DatetimeIndex or that the column being indexed contains properly formatted datetime objects. Additionally, the date range should be in the correct order (start\_date:end\_date). You can resolve this by first converting the index or column to a DatetimeIndex using pd.to\_datetime() and then sorting the index with .sort\_index() before slicing.

* Correct Code:



Changes:

* Use pd.to\_datetime() to ensure the index/column is in proper datetime format.
* Apply sort\_index() after converting the index/column to ensure proper order for slicing.
* Error 3:



The error occurs because the .loc() method is attempting to slice the DataFrame with a date range, but the crime DataFrame's index is likely not a DatetimeIndex, or the provided date range is invalid or improperly formatted. To fix this, first ensure the index is converted to DatetimeIndex using pd.to\_datetime() and that the index is sorted using .sort index(). Additionally, verify that the date range provided in .loc() is valid and properly formatted in the YYYY-MM-DD format.

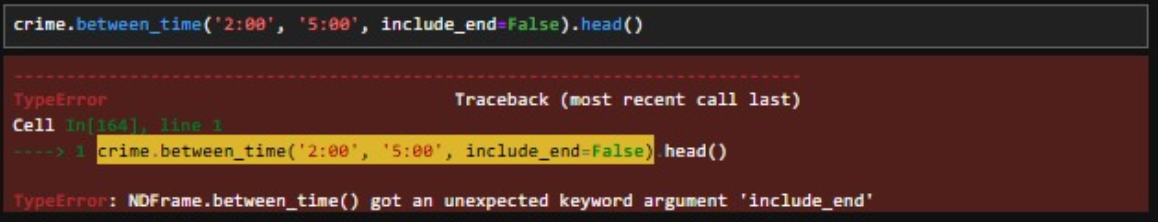
* Correct code:

A black screen with white text

Description automatically generated

Changes:

* Convert the DataFrame index to a DatetimeIndex using pd.to\_datetime().
* Use sort\_index() to ensure the datetime index is in the correct order.
* Verify and use a valid date range in the .loc[] method.
* Error 4:



The error occurs because the between\_time() method in Pandas does not support the keyword argument include\_end. To resolve this, simply remove the include\_end parameter, as it is not a valid argument for this method. By default, between\_time() includes both the start and end times in the selection

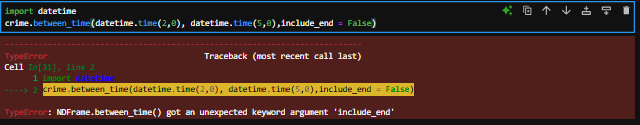
* Correct Code:

A screenshot of a computer

Description automatically generated

Changes

* Error: include\_end=False → This parameter is **not supported** in between\_time().
* **Correct**: Removed include\_end and used only between\_time('2:00', '5:00').
* Error 5:



The error arises because the between\_time() method in Pandas does not support the include\_end parameter. The method, by default, includes both the start and end times in the range. If you want to exclude the end time, you will need to manually filter the DataFrame after using between\_time(). To fix the issue, remove the include\_end parameter from the function call.

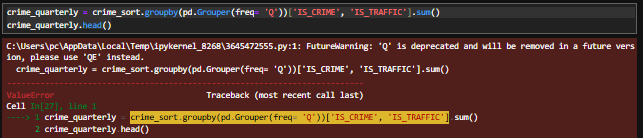
Correct Code:

A screenshot of a computer

Description automatically generated

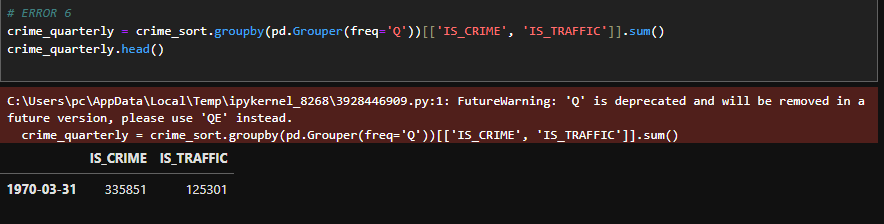
Changes:

* Error: include\_end=False → This parameter is invalid in between\_time().
* **Correct**: Appended .head() to display the first few rows of the result.
* Error 6:



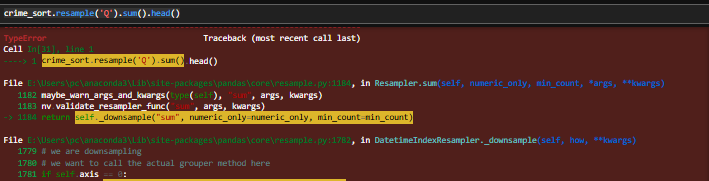
The error occurs because in the line crime\_sort.groupby(pd.Grouper(freq='Q’)['IS\_CRIME', 'IS\_TRAFFIC'], the columns ['IS\_CRIME', 'IS\_TRAFFIC'] are specified using a tuple instead of a list. In Pandas, when selecting multiple columns, you must use a list syntax, not a tuple. This causes the ValueError. To fix this, replace the tuple with a list when selecting the columns.

* Correct Code:



Changes:

* **Error**: ['IS\_CRIME', 'IS\_TRAFFIC'] as separate arguments (deprecated).
* **Correct**: Use double square brackets [[]] to select multiple columns explicitly.
* Ensures the code aligns with the proper syntax for groupby() operations in pandas.
* Error 7:



The error occurs because the resample('Q').sum() operation is being applied to columns with a category data type, and the sum() function is not supported for categorical data. To resolve this issue, you need to exclude categorical columns before performing the sum() operation. This can be done by selecting only numeric columns using the select\_dtypes() method

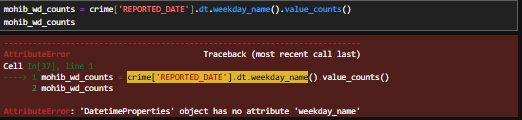
* Correct Code:

A screenshot of a computer

Description automatically generated

Changes:

* **Error**: sum() implicitly sums numeric columns.
* **Correct**: Added numeric\_only=True to explicitly sum only numeric columns, ensuring clarity and avoiding potential warnings or errors.
* Error 8:



The error occurs because weekday\_name is no longer available as an attribute of DatetimeProperties in recent versions of Pandas. This attribute has been replaced with day\_name(). To fix this, simply replace .dt.weekday\_name() with .dt.day\_name() in the code.

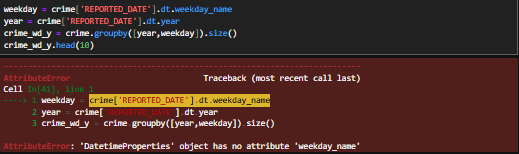
* Correct Code:

A screen shot of a computer

Description automatically generated

Changes:

* **Error**: .weekday\_name() → This method is deprecated in recent versions of pandas.
* **Correct**: .day\_name() → The updated method for getting the name of the day of the week.
* Error 9:



The error occurs because weekday\_name is no longer supported in recent versions of Pandas. This attribute has been replaced by day\_name(). To fix the error, you need to use .dt.day\_name() instead of .dt.weekday\_name. Additionally, ensure that the variables year and weekday are correctly extracted before grouping.

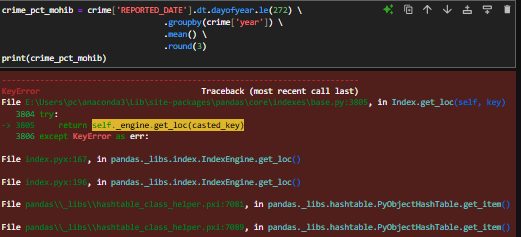
* Correct Code:

A screenshot of a computer program

Description automatically generated

Changes:

* Error: .weekday\_name → Deprecated method for getting the name of the day.
* **Correct**: .day\_name() → Updated method for getting the name of the day of the week.
* Used the updated day\_name() method to ensure the code works with the latest version of pandas.
* Error 10:



The error occurs because there is no column named 'year' in the crime DataFrame. To fix this, you need to create a 'year' column explicitly by extracting the year from the REPORTED\_DATE column before using it in the groupby operation

* Correct Code:

A screen shot of a computer code

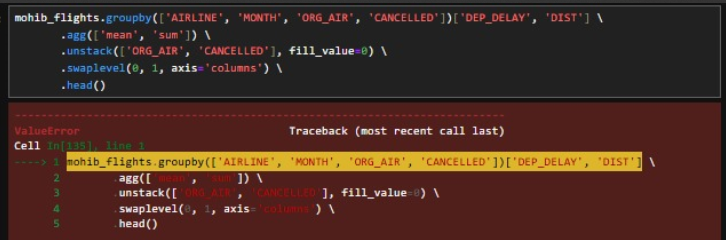
Description automatically generated

Changes:

* **Error**: The code assumes crime['year'] already exists, but it’s not defined.
* **Correct**: Added crime['year'] = crime['REPORTED\_DATE'].dt.year to explicitly create the 'year' column before using it in the groupby() operation.

Errors of chap 8:

* Error 1



The error occurs because the column names in the groupby method or subsequent operations are not properly recognized. This typically happens if there is a mismatch in column names, either due to typos, extra spaces, or missing columns in the DataFrame. The solution is to verify that all column names specified in groupby (like 'AIRLINE', 'MONTH', 'ORG\_AIR', 'CANCELLED', 'DEP\_DELAY', 'DIST') and the subsequent operations exist in the DataFrame and match exactly as spelled.

* Correct Code:

A screenshot of a computer program

Description automatically generated

Changes:

* Verify Column Names: Check the DataFrame column names with mohib\_flights.columns and ensure they match ['AIRLINE', 'MONTH', 'ORG\_AIR', 'CANCELLED', 'DEP\_DELAY', 'DIST'].
* Nested Brackets: Ensure you're using double brackets ['DEP\_DELAY', 'DIST']] when selecting multiple columns for aggregation.
* Debugging: If the issue persists, verify the structure and data types of the DataFrame to ensure it can handle the operations correctly.
* Error 2

A screenshot of a computer screen

Description automatically generated

The error occurs because the syntax for selecting multiple columns in the groupby method and subsequent agg function is incorrect. Specifically, the agg function is applied directly to the groupby object, but the columns to be aggregated ('UGDS' and 'SATMTMID') are not properly specified. Use double brackets [['UGDS', 'SATMTMID']] to select multiple columns before applying the agg function

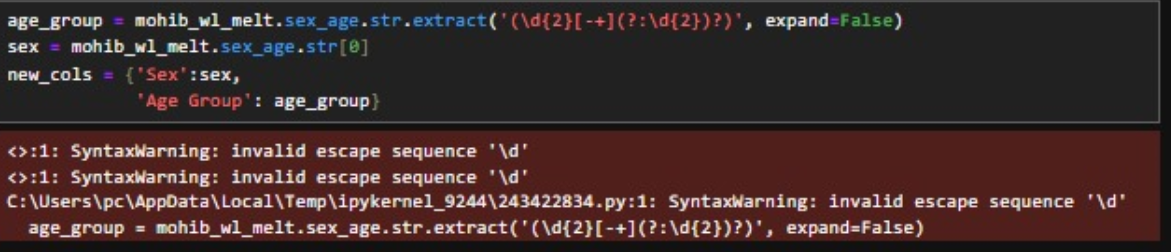
* Correct Code:

A screenshot of a computer program

Description automatically generated

Changes:

* Ensure all column names ('STABBR', 'RELAFFIL', 'UGDS', 'SATMTMID') exist in the DataFrame by checking with mohib\_clg.columns.
* Use double brackets when selecting multiple columns for aggregation.
* Error 3



The error is caused by an **invalid escape sequence** in the regular expression \d. In Python strings, backslashes (\) are treated as escape characters. To avoid this warning and ensure the regular expression works as intended, the string should either use a raw string (r'...') or double the backslashes ([\\d](file:///\\d)).

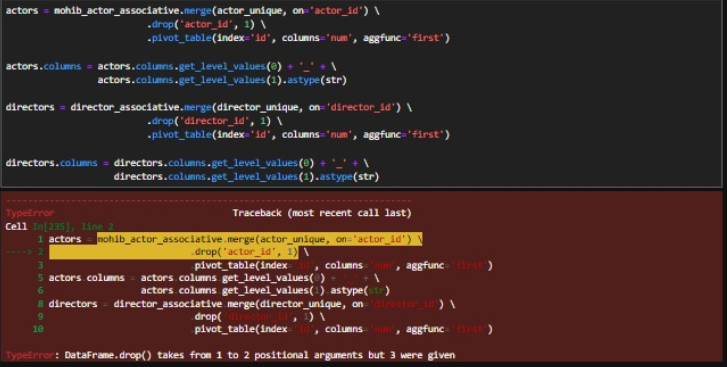
* Correct Code:

A black screen with red text

Description automatically generated

Changes:

* Use r'...' to define raw strings for regular expressions to prevent Python from interpreting backslashes as escape characters.
* Verify the regular expression matches the desired format for the sex\_age column.
* Error 4



The error occurs because the drop method in pandas uses axis as a keyword argument, not a positional argument. Passing 1 as a positional argument to specify the axis (indicating columns) is deprecated and no longer valid in recent pandas versions. Update the drop method to use the axis keyword argument explicitly.

* Correct Code:

A computer screen shot of colorful text

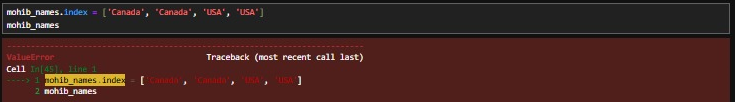
Description automatically generated

Changes:

* Replace .drop('actor\_id', 1) with .drop('actor\_id', axis=1).
* No changes are needed for .drop(columns='director\_id') as it is already correct.

Errors of chap 9:

* Error 1



The error occurs because the number of new index values provided (['Canada', 'Canada', 'Canada', 'USA'], which has 4 elements) does not match the number of rows in the DataFrame or Series mohib\_names (which has 5 elements). In pandas, the length of the new index must match the length of the data.Ensure that the number of new index values matches the number of rows in mohib\_names.

Correct Code:

A black screen with red text

Description automatically generated

Changes:

* **Error**: The index has 4 elements (['Canada', 'Canada', 'Canada', 'USA']), but the mohib\_names object likely has 5 elements, causing a ValueError.
* **Correct**: Updated the index to match the number of elements in mohib\_names, which now has 5 elements (['Canada', 'Canada', 'Canada', 'USA', 'USA']).
* Error 2

A screen shot of a computer

Description automatically generated

The error occurs because the append method has been removed in recent versions of pandas (starting from version 2.0). To add a new row to a DataFrame, you should use the pd.concat method instead, which is the recommended approach.Use pd.concat to append a new row to the DataFrame.

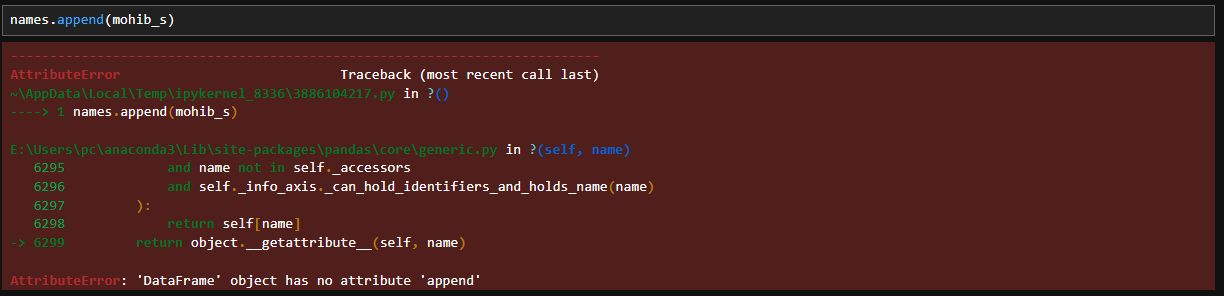
* Correct Code:

A computer screen shot of a computer code

Description automatically generated

Changes:

* Create the new row as a DataFrame (pd.DataFrame).
* Use pd.concat to combine the original DataFrame with the new row.
* Use ignore\_index=True to reset the index after concatenation.
* Error 3



The error occurs because the append method is no longer supported in pandas 2.0 and later versions. To append a DataFrame or Series (mohib\_s) to another DataFrame (names), you should use the pd.concat method, which is the replacement for the append method.Use pd.concat to append the mohib\_s object to the DataFrame

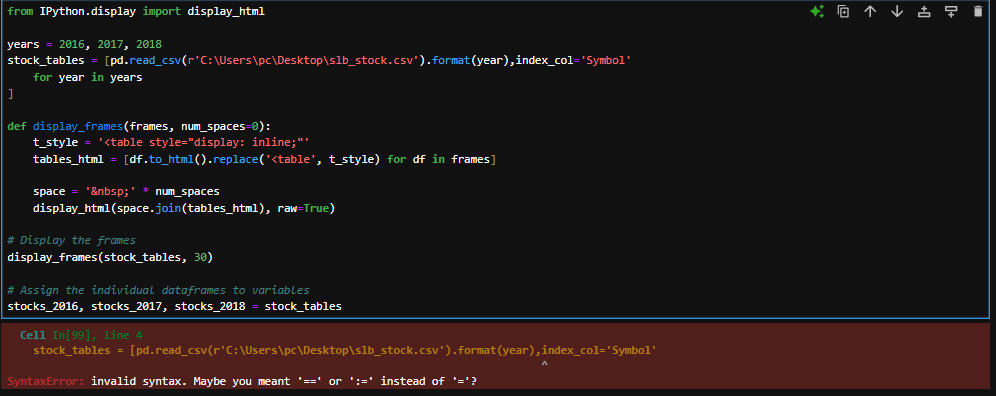
* Correct Code:

A screenshot of a computer

Description automatically generated

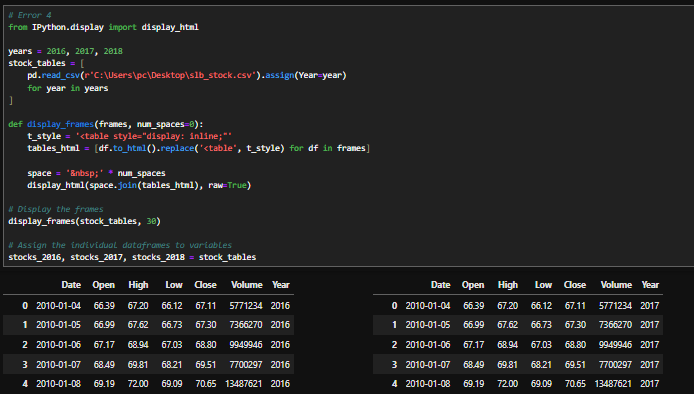
Changes:

* Ensure mohib\_s is either a DataFrame or a Series with matching columns to names.
* Use ignore\_index=True to reset the index after concatenation if necessary.
* pd.concat is the recommended method for combining DataFrames and Series in modern pandas versions.
* Error 4



The error in the code occurs because of a syntax mistake in the list comprehension for stock\_tables. The format(year) method is incorrectly placed outside the pd.read\_csv function call, leading to syntax errors. Additionally, the index\_col='Symbol' is not properly inside the read\_csv method call. You need to correctly format the list comprehension and ensure that method calls are properly separated.

Correct Code:



Changes:

* Fixed the Placement of assign() Method: The assign(Year=year) is used to add the Year column for each year in the loop.
* Corrected index\_col='Symbol' Inside read\_csv(): This ensures that the Symbol column is set as the index when reading the CSV.