GPR: Python Coding Exam

Windows Users

Create a conda environment using the CodingTest.yml file and work from this environment.

Linux/macOS Users

Build your own environment. Use conda instead of pip. Here are the constraints:

python==3.9.1

pandas==1.2.3

Python Exam Instructions

Clone the coding test repo from github: https://github.com/scravo001/CodingTest

Create a new branch for your submission. Name the branch FIRSTNAME_LASTNAME.

Create a function called transform inside the main.py module that returns a dataframe properly formatted to match the expected output. Do not mutate the input dataframe. The runtime of the transform function should be below 1 second.

1. Drop any rows with "Not Found" in the Error column. The Error column may or may not be in the columns.
2. Create a column called contract, which is a copy of the Term column. If there are nulls in the Term column, fill it with the Period column.
3. Check for nulls in Trade Date column. Raise a warning if any nulls exist and drop the rows with null Trade Date.
4. Check for expired instruments. An instrument is expired if the Expiration Date is older than the Trade Date. Raise a warning if there are expired instruments and drop the rows with expired instruments.
5. Check for nulls in contract column. Raise a warning if nulls exist and drop the rows with null contract name.
6. Parse the RIC base and moneyness from the RIC column and merge instruments dataframe on the base.
   Example: 1BO50Nc1=R -> moneyness = 50, base=1BO
   Example: 1BO100Nc1O=R -> moneyness = 100, base=1BO
7. Create two columns called contract_year and contract_month. For contract_year, parse the year from Expiration Date. For contract_month, parse the month from Period (JAN, FEB, etc.). If the last digit of the contract_year does not equal the last digit of the Period column, increment contract_year by 1. You may or may not see Jan/Feb contracts that do not match the year of the expiration date. For instance, a Jan 2021 contract may expire in December 19, 2020.
8. Create a column called month_code that maps contract_month using the MONTH_NAME_TO_CODE dict.
9. Rename some columns using FIELD_MAP.
10. Create a column called symbol that is a concatenation of "FUTURE_VOL_", Exchange, Bloomberg Ticker, month_code, contract_year, and moneyness. Single character Bloomberg Tickers are special.
    Example: FUTURE_VOL_CBT_BOM2022_50, FUTURE_VOL_CBT_W_N2023_150

11. Transform the rest of the raw data to match the expected output.
    Hint: Start with the FLOAT_FIELDS list to extract contract values, and ensure that the column types are appropriate.

How To Submit

Push the branch to github for us to review. Once the branch has been pushed, please email pierre@greenpoint-research.com to inform him of the successful exam completion.