Project 2: Portfolio Analyses and Showcase your work

Introduction

Effectiveness in Finance research and industry necessitates a facility with data aggregation, combination, selection, and manipulation. In the previous project, you are asked to write general code to assemble a data set from different sources, and developed foundational skills of data acquisition, cleaning, and merging. In project2, you will combine data from multiple CSV files into a single table, calculate stock returns, compute variables using returns and merge tables to achieve our desired output variables. The coding work needs to be done collaboratively using git. Beyond the coding, this project 2 codebase helps us investigating the relation between total volatility and expected stock returns using univariate portfolio analyses. It tries to address whether investors can earn significant returns by investing in a volatility long-short portfolio. Furthermore, we want to help you develop the skills to showcase your work. Your team will need to record a video primarily for describing the portfolio analyses methodology and how the codebase has been designed and implemented.

The Source Files

All required files are included in a zip archive with the following structure:

```
<ZIPFILE>
|__ project2/
| |
| |__ __init__.py
| |__ config.py
| |__ project_desc.pdf
| |__ util.py
| |__ zid_project2_characteristics.py
| |__ zid_project2_main.py
| |__ zid_project2_portfolio.py
| |__ zid_project2_etl.py
| |
| |___data/
| | | <many csv files here>
```

where

- project2/ represents the main folder containing all the project files.
- zid_project2_main.py, zid_project2_etl.py, and zid_project2_characteristics.py contains the functions you need to write for this project. These are the files you need to submit.
- project_desc.pdf is the PDF version of this document.
- data/: This is the sub-directory where all the data files for this file are stored. Inside this folder you will find many files. Each <tic>_prc.csv contains stock price data for the ticker <tic>. These CSV files include the column names in a header row of text.
- config.py is the configuration module for this package. **You should not modify this file.**

Instructions

Important: This project is a group project. Do not exchange complete or partial codes with students from **other** groups. Please do not post any project related questions in public online forums.

Again, please do not post any project related question in a public online forum, including the discussion board in ED. We will deduct marks from your project if you violate this important rule.

Preparing the files for this project

1. Copy the project2 folder into the toolkit project folder. Afterwards, your toolkit folder will look like:

```
toolkit/                        <- PyCharm project folder
| ...
|__ project2/                   <- New folder
|    |
|    |__ __init__.py
|     |__ config.py
|     |__ project_desc.pdf
|     |__ util.py
|     |__ zid_project2_characteristics.py   <-This is one of the files you need to edit
|     |__ zid_project2_main.py              <-This is one of the files you need to edit
|     |__ zid_project2_portfolio.py
|     |__ zid_project2_etl.py               <-This is one of the files you need to edit
|     |
|     |___data/
|     |     | <many csv files here>
|     ...
|__ toolkit_config.py                       <- Your toolkit_config.py (required)
| ...
```

2. Unless explicitly stated below, **do not change any variable, import statement, function, or parameter names in the project2 module.**

How to complete this project

This project coding section has 11 parts, which should be completed in sequence. Except parts 1 ,3, and 6, all other parts will be marked.
You can find the number of marks for each part at the end of this document. Each part is described in detail in the next section.

Overview:
• Part 1: Read the documentation for the following methods:
– pandas.DataFrame.mean
– pandas.Series.concat
– pandas.Series.count
– pandas.Series.dropna
– pandas.Series.index.to_period
– pandas.Series.prod
– pandas.Series.resample ……
In the project coding section, you can utilize modules covered in our lectures, listed above and any others. We suggest reading the documentation and examples of a new module before using it and incorporating it into your project.
• Parts 2 to 7: Complete the functions in scaffolds: zid_project2_main.py, zid_project2_etl.py, and zid_project2_characteristics.py. See the step-by-step instructions below.
• Part 8: Answer the described in Part 8 below by setting the value of the relevant variables in zid_project2_main.py (i.e., Q1_ANSWER, Q2_ANSWER, etc. . .)
• Part 9: Add t_stat function in zid_project2_main.py. See the step-by-step instructions below.
• Part 10: share your team's project 2 git log. See the step-by-step instructions below.
• Part 11: share your recording link of the project 2 mini-presentation. See the step-by-step instructions below.

How to submit the project and complete the cross-marking task

Please make sure each *group* submits one project:

1. Each **group** should submit **3 completed version** of .py files: zid_project2_main.py, zid_project2_etl.py, and zid_project2_characteristics.py. Your team should choose one member who will be responsible for submitting the project. There is no need to tell us in advance who the team representative will be.
    - The group representative should copy and paste the entire contents of the three zid_project2_<..>.py modules to ED.
    - Please make sure only one team member submits the group project. If multiple members of the same group submit, we will only consider the last submission by any student belonging to the group.
    - Remember to press “Mark” to submit your project. Your project will not be submitted until you do so.

2. The purpose of test functions in zid_project2_<…>.py is assisting your coding. You can edit it when coding. For the finalised version you submit in ED, please only update the parts we point out, i.e., <COMPLETE THIS PART>, the variables we ask you to define a value, like Q1_ANSWER etc, and add functions if we ask you to do. Keep all the other parts the same, like functions' docstring, test functions, etc.

Please make sure *each student* finishes cross-group marking task:

3. Each student must also complete a cross-group marking task. We will share the presentations of at most five other groups with you. You will need to mark them according to specific criteria. The workshop will be conducted in Moodle (not in ED). An announcement will be made when the workshop is ready. This task is to be completed individually by each student.

To make sure this process is clear to everyone, here is an example. Suppose a group consists of three students A, B, and C. The group decides that student A will submit the zid_project2_<…>.py files.

- Student A will submit the files through ED. Student A will copy the content of the completed 3 zid_project2_<…>.py files, navigate to the "Submit your codes here" slide in ED, and then paste the code. After that, student A will press "Mark" to submit the code.
- Students B and C should ignore the "Submit your codes here" slide in ED.
- Students A, B, and C will then separately complete their cross-group marking task in Moodle (not ED).

Completing the three zid_project2_<…>.py modules

After setting up your PyCharm development environment with the project files (see instructions above), modify the zid_project2_<…>.py modules by following the steps below, in sequence.

Part 1: Read the relevant documentation

Read the documentation for the following methods:

- pandas.DataFrame.mean (note the parameter axis which will indicate if the mean will be computed
- column-wise or row-wise)
- pandas.Series.concat
- pandas.Series.count
- pandas.Series.dropna
- pandas.Series.index.to_period
- pandas.Series.prod
- pandas.Series.resample

Part 2: Include a statement to import the config and util modules

Open the config.py and util.py files included in this project in PyCharm. **Please do not modify these modules.** Note that these files include constants (e.g., DATADIR, TICMAP, TICKERS etc.) and auxiliary functions (e.g. test_print, color_print, etc.). The config and util modules must be imported by the zid_project2_main.py module. This is because the module needs access to the constants and functions defined in config.py and util.py.

Complete the import portion of the zid_project2_main.py module by creating two new import statements. These statements should import the module config.py and util.py which is part of the zip file provided to you. Your import statement must:

- Take into account that the config.py and util.py modules are inside the project2 package.
- Import the config.py module using "cfg" as an alias (so, "as cfg")
- Import the util.py module without giving it an alias

Part 3: Follow the workflow in portfolio_main function to understand how this project construct total volatility portfolio

In this project, we are going to use the simplest methodology to construct equal-weighted quantile and long-short portfolios. To understand this codebase, you need to read the functions and its docstrings following the workflow in portfolio_main function in zid_project2_main.py.

The project consists of three parts. The first part is called ETL, which stands for extract, transform, and load. It involves a three-phase process where data is extracted from an input source, transformed (including cleaning), and then loaded into an output data container. In this project, the zid_project2_etl.py file is used to calculate daily and monthly stock returns using price data imported from the project's data folder. The output of the step is a dictionary containing the return series.

The second part focuses on constructing characteristics. In this project, our aim is to test whether a volatility long-short portfolio yields significant returns. To achieve this, we will compute the characteristic of volatility for the stocks in our investment universe. The zid_project2_characteristics.py script is designed to calculate monthly volatility for each stock using daily returns. The output of the script is a DataFrame containing the monthly returns and volatilities of the stocks."

The third part involves constructing portfolio returns. It will generate a DataFrame containing the equal-weighted average monthly returns of quantile and long-short portfolios.

Part 4: Complete etl scaffold to generate returns dictionary and to make ad_ret_dic function works

- **Part 4.1: import needed modules**
  Create import statements to import all modules you need in this script. Please keep the aliases consistent throughout the project. For example, use 'cfg', 'etl', 'cha', and 'pf' as an alias for config.py, zid_project2_etl.py, zid_project2_characteristics.py and zid_project2_portfolio.py in project2 folder.

  For modules like pandas, you can decide the shortcut for them. But you need to keep the aliases consistent throughout the project.

- **Part 4.2: Complete the read_prc_csv function**
  Complete the indicated part of the function read_prc_csv so it produces the output described in the docstring. You can test this function by calling the _test_read_prc_csv test function.
- **Part 4.3: Complete the daily_return_cal function**
  Complete the indicated part of the function daily_return_cal so it produces the output described in the docstring. You can test this function by calling the _test_daily_return_cal test function.
- **Part 4.4: Complete the monthly_return_cal function**
  Complete the indicated part of the function monthly_return_cal so it produces the output described in the docstring. You can test this function by calling the _test_monthly_return_cal test function.
- **Part 4.5: Complete the aj_ret_dict function**
  Complete the indicated part of the function aj_ret_dict so it produces the output described in the docstring. You can test this function by calling the _test_aj_ret_dict test function.

Part 5: Complete cha scaffold to generate data frame containing monthly total volatility for each stock and to make char_main function work

- **Part 5.1: import needed modules**
  Create import statements to import all modules you need in this script. Please keep the aliases consistent throughout the project. For example, use 'cfg', 'etl', 'cha', and 'pf' as an alias for config.py, zid_project2_etl.py, zid_project2_characteristics.py and zid_project2_portfolio.py in project2 folder.

  For modules like pandas, you can decide the shortcut for them. But you need to keep the aliases consistent throughout the project.
- **Part 5.2: Read the cha_main function and understand the workflow in this script**
  Please read the docstring of cha_main and figure out the workflow of this script. You can test this function by calling the _test_cha_main function.
- **Part 5.3: read the vol_input_sanity_check function and use it to test if the inputs of zid_project2_characteristics are proper**

Read the vol_input_sanity_check function. You can test this function by calling the _test_vol_input_sanity_check test function. When use or test this function, you should specify its three parameters. For 'ret', you can use either the made-up return dictionary generated by _test_ret_dict_gen function or the output from etl script. For cha_name and ret_freq_use, please follow the cha_main docstring to specify.

- **Part 5.4: Complete the vol_cal function**
  Complete the indicated part of the function vol_cal so it produces the output described in the docstring. You can test this function by calling the _test_vol_cal test function.
- **Part 5.5: Complete the merge_tables function**
  Complete the indicated part of the function merge_tables so it produces the output described in the docstring. You can test this function by calling the _test_merge_tables test function.

Part 6: Read and utilize portfolio construction functions in zid_project2_portfolio.py to answer some of the questions in Part 7

Read the functions in this script and fully understand how the project is constructing total volatility portfolios. You can utilize the test functions to get a better understanding.

Part 7: Complete the auxiliary functions

Complete the following auxiliary functions following the instructions specified in their docstrings:
• get_avg: calculate the average value of all columns in the given data frame for a specified year.
• get_cumulative_ret: calculate the cumulative returns for portfolios in the given data frame.
You can test these functions by calling appropriate test functions.

Part 8: Answer questions

For this part of this project, you should answer the questions below. Your answers should be included in the zid_project2_main.py module. For example, answer Q1 by setting the value of Q1_ANSWER in the zid_project2_main.py file.
All your answers should be strings. If they represent a number, include 4 decimal places unless otherwise specified in the question description. When marking this part of the project, we will ignore string capitalization (i.e., lowercase vs uppercase characters).

To answer the questions below, you need to finish the coding work and then run portfolio_main function in the zid_project2_main.py with the following parameter values:

- tickers: all tickers included in the dictionary config.TICMAP,
- start: '2000-12-29',
- end: '2021-08-31',
- cha_name: 'vol'.
- ret_freq_use: ['Daily',],
- q: 3

Specifically, these parameters instruct the codebase to construct equal-weighted average monthly returns of tertile and long-short portfolios. These portfolios are formed by sorting the stocks specified in the TICKERS variable in config.py based on their volatility within the previous month (vol). The sample period spans from 2000-12-29 to 2021-08-31.
Please name the three output files as DM_Ret_dict, Vol_Ret_mrg_df, EW_LS_pf_df. You can utilize the three output files and auxiliary functions to answer the questions.

Part 9: Add t_stat function

We've outputted EW_LS_pf_df file and save the total volatility long-short portfolio in 'ls' column from Part 8. Please add an auxiliary function called 't_stat' in zid_project2_main.py. You can design the function. But make sure that when function get called, t_stat(EW_LS_pf_df), the output is a DataFrame with one row called 'ls' and three columns below:

1. ls_bar, the mean of 'ls' columns in EW_LS_pf_df, keep 4 decimal points
2. ls_t, the t stat of 'ls' columns in EW_LS_pf_df, keep 4 decimal points
3. n_obs, the number of observations of 'ls' columns in EW_LS_pf_df, save as integer

When calculate t stat of 'ls', use formula below:

t = mean of 'ls'/Standard error for mean of 'ls'

Be careful, please add the function in zid_project2_main.py. The name of the function should be t_stat and including docstring.
After calculating, replace the '?' of ls_bar, ls_t and n_obs variables in zid_project2_main.py with the respective values.

Part 10: share your team's project 2 git log

In week6 slides, we introduce Git and show how to work collaboratively on Git.

- You are not necessary to use your UNSW email to register the git account.
- When you set up your username in git, you need to follow the format zidFirstNameLastName,
  - for example: zid1234567SarahXiao
- Please follow the instruction in week6 slides to work with your teammates. The team leader will need to create a Project 2 Repo on GitHub and grant teammates access to the Repo. For teammates, you will need to clone the repo and then coding as a team.
- The team will need to generate a git log from git terminal.
  1. You can use 'cd <...>' direct your git terminal access to the project 2 directory,
  2. Export the git log using command:
     ```
     git log --pretty=format:"%h%x09%an%x09%ad%x09%s" >file_name.txt
     ```
     Here is an example output:

```
.......
dae0fa9 zid1234SarahXiao        Mon Feb 12 16:33:22 2024 +1100   commit and push test
fa26a62 zid1234SarahXiao        Mon Feb 12 16:32:02 2024 +1100   commit and push test
800bf27 zid5678DavidLee         Mon Feb 12 16:12:30 2024 +1100   for testing
......
```

Please replace the """?""" of git_log variable in zid_project2_main.py with your team's project 2 git log.

Part 11: project 2 mini-presentation

In this part, you are going to record and present a strictly less than 15 minutes long presentation. You should seek to briefly describe:

1. What are the null and alternative hypotheses that the project 2 is testing
2. What's the methodology of the portfolio construction and how is it implemented in Project 2 codebase?
3. What inferences can we draw from the output of Part 9, including the average return and t-stats of the long-short portfolio?
4. Do you think the results are reliable? Why or why not?
5. Is there any further work you would like to pursue based on Project 2? Share your to-do list.

For this mini-presentation, the group can decide whether all members should appear in the presentation video. You can use websites like YouTube or Zoom to record and share your videos with us, or share your videos via OneDrive. Please **AVOID** using VooV, QQ, and WeChat to share videos, as we have faced access issues with these platforms previously.

Please replace the '?' of Presentation_link variable with your team's video link. If you have set a password, please replace the '?' of Presentation_Password variable with the actual password to ensure accessibility or leave it as it is.

Important:
• The file zid_project2_main/etl/characteristics.py contains placeholders for your answers.
• You should replace the relevant variables in zid_project2_main/etl/characteristics.py files with your answers. For instance, your answer to Q1 in Part 8 should be included in the variable Q1_ANSWER.
• You can create a separate module and then use the functions you defined to answer the questions.
HOWEVER, THE ONLY THREE MODULES YOU SHOULD SUBMIT ARE zid_project2_main/etl/characteristics.py.
• All your answers should be strings. If they represent a number, include 4 decimal places unless otherwise specified in the question description.
• Here is an example of how to answer the questions below. Consider the following question:

Q0: Which ticker included in config.TICMAP starts with the letter “C”?
Q0_ANSWER= ‘?’
You should replace the ‘?’ with the correct answer:
Q0_ANSWER = ‘CSCO’

Administrative Guidelines and Hints

We will enforce the following:
1. This assessment must be completed in groups, but you should not cooperate with students from other groups. Failure to do so may result in a full loss of marks.
2. Late submissions are allowed, but will be penalised following the guidelines described in the course outline.

Hints

Your code should be portable, working in a variety of settings. For example, we should be able to run your codes in different computers using different operating systems. We should also be able to import and run your code from other modules.
The following hints should help you correct any portability mistakes:
1. The contents of your zid_project2_main/etl/characteristics.py modules **must not contain any direct reference to folders in your computer.** In other words, you must use the variables in the config.py and the os module to create path variables.
2. When writing functions in the file zid_project2_main/etl/characteristics.py:
• Do not modify the function names or the parameters.
• Only modify the parts indicated by the "<COMPLETE THIS PART>" tag or add functions if you are asked to.
3. **Only** submit zid_project2_main/etl/characteristics.py modules.

How we will mark your assessment

The following parts of this assessment will be marked. The project is worth a total of 100 marks.
• Part 2: Importing modules inside the project2 package (1 marks)
• Part 4: complete etl scaffold to generate returns dictionary and to make ad_ret_dic function works (20 marks)
• Part 5: complete cha scaffold to generate dataframe containing monthly total volatility for each stock and to make char_main function work (20 marks)
• Part 7: Auxiliary functions:
– get_avg function (2 marks)
– get_cumulative_ret (2 marks)

• Part 8: Answer the following questions:
– Each question is worth 1.5 marks (for a total of 15 marks)
• Part 9: Add t_stat function(5 marks)
• Part 10: share your team's project 2 git log (5 marks)
• Part 11: project 2 mini-presentation (30 marks in total)
1.What are the null and alternative hypotheses that the project 2 is testing (2 marks)
2.What’s the methodology of the portfolio construction and how is it implemented in Project 2 codebase? (20 marks)
3.What inferences can we draw from the output of Part 9, including the average return and t-stats of the long-short portfolio? (3 marks)
4.Do you think the results are reliable? Why or why not? (2marks)
5.Is there any further work you would like to pursue based on Project 2? Share your to-do list. .(3 marks)