

Project 2: Quantitative Trading Strategy Construction and Evaluation

Introduction

Effectiveness in Finance research and industry necessitates a facility with data aggregation, combination, selection, and manipulation. Building on the foundational skills acquired in Project 1, this project extends the focus to the construction, implementation, and evaluation of a quantitative trading strategy. In this project, students will integrate price data from multiple sources, compute stock returns, derive volatility measures, and construct equal-weighted portfolios sorted by total volatility. The objective is to investigate whether a volatility-based long-short strategy can generate statistically and economically significant returns.

Beyond technical implementation, this project emphasises communication and presentation skills simulating the real-world task. Students will prepare a written report and deliver a group presentation to showcase their methodology, findings, and insights.

By completing this project, students will:

- Develop a structured approach to construct and test financial trading strategies;
- Gain practical experience with data processing and portfolio analysis in Python;
- Enhance their ability to communicate technical findings to a professional audience.

This project offers a comprehensive, applied learning experience that reflects the expectations of roles in quantitative finance and investment analysis.

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/
| | | example_prc.csv < This is a sample file provided for testing purposes only>
| | | <where you save data files>
```

where

- project2/ represents the main folder containing all the project files.
- config.py, 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 you save data files for this project. You will find a sample file provided for testing purposes only.
- config.py is the configuration module for this package.
- util.py is the module that contains auxiliary functions. **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, marks will be deducted if this rule is violated.

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                       <-This is one of the files you need to edit
| | __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/
| | | example_prc.csv < This is a sample file provided for testing purposes only>
| | | < where you save data files >
| | ...
| __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.

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.

In the Week 6 slides, we introduced Git and demonstrated how to collaborate using it. You can follow the instructions in those slides to work with your teammates. The team leader should create a Project 2 repository on GitHub and grant access to the rest of the team. Team members will then need to clone the repository and collaborate on the code. This step is optional and will not be assessed.

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: config.py, 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: Prepare the project presentation. See the step-by-step instructions below.
- Part 11: Prepare the project report. See the step-by-step instructions below.

How to submit the project

Please make sure each *group* submits one project:

1. Each **group** should submit **4 completed version** of .py files: config.py, 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 four 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.
3. Each **group** must submit **one presentation deck (e.g., PowerPoint or PDF)** and **one written report (e.g., PDF or Word document)**. These should represent your team’s collective work. Further submission details will be announced on Ed closer to the due date.

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 config.py and zid_project2_<...>.py files.

- Student A will submit the files through ED. Student A will copy the content of the completed 4 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.
- Student A will submit one presentation deck and one written report, following the instructions that will be announced on Ed closer to the due date.

Completing the config.py and three zid_project2_<...>.py modules

After setting up your PyCharm development environment with the project files (see instructions above), modify the config.py and 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. 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 download price data from a database into the project's data folder and then calculate daily and monthly stock returns based on that data. 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 raw 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 download_prc_to_csv function**

- **Part 4.2.1 Define the TICMAP Dictionary in config.py**

Define the TICMAP dictionary as instructed in config.py. It will be used throughout the project to download and organize price data for your selected investment universe.

- **Part 4.2.2 Complete download_prc_to_csv function**

Complete the indicated part of the function download_prc_to_csv so it produces the output described in the docstring. You can test this function by calling the _test_download_prc_to_csv test function.

- **Part 4.3: 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 4.4: 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 4.5: 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 4.6: 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 cha_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 config.TICMAP dictionary define your team's investment universe
- 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:

$$t = \text{mean of 'ls'} / \text{Standard error for mean of 'ls'}$$

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: project presentation

Your team will deliver a project presentation during the **Week 10 lecture**. Further details about the presentation will be announced on Ed closer to the date.

Please refer to "How we will mark your assessment" section below for detailed instructions on Part 10.

Part 11: project report

Your team is required to submit a project report.

Please refer to "How we will mark your assessment" section below for detailed instructions on Part 11.

Important:

- The file config/zid_project2_main/zid_project2_etl/zid_project2_characteristics.py contains placeholders for your answers.
- You should replace the relevant variables in config/zid_project2_main/zid_project2_etl/zid_project2_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 FOUR MODULES YOU SHOULD SUBMIT ARE config/zid_project2_main/zid_project2_etl/ zid_project2_characteristics.py.

- All 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 `config/zid_project2_main/ zid_project2_etl/ zid_project2_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 `config/ zid_project2_main/ zid_project2_etl/ zid_project2_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 `config/zid_project2_main/ zid_project2_etl/ zid_project2_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 mark)
- Part 4: complete etl scaffold to generate returns dictionary and to make `ad_ret_dic` function works (15 marks)
- Part 5: complete cha scaffold to generate dataframe containing monthly total volatility for each stock and to make `char_main` function work (12 marks)
- Part 7: Auxiliary functions:
 - `get_avg` function (1.5 marks)
 - `get_cumulative_ret` (1.5 marks)
- Part 8: Answer the following questions:
 - Each question is worth 0.5 marks (for a total of 5 marks)
- Part 9: Add `t_stat` function(4 marks)
- Part 10: Project Presentation (30 marks)
 - Your team will deliver a project presentation during the **Week 10 lecture**. Further details about the presentation will be announced on Ed closer to the date.

Imagine this: You are a junior quant developer at a fintech startup that has built a prototype trading strategy back-testing system. Your task is to showcase the codebase and your analysis to a fund manager at a global asset management firm.

Your audience is technical but pressed for time. You have **10 minutes** to deliver a concise, convincing presentation that addresses the **five questions below**. The goal is to demonstrate that your system is well-designed, the methodology is sound, and the results are insightful.

1. Trading Strategy Construction and Implementation (25%)

- What characteristic or signal is used in the strategy?
- How is the long-short portfolio constructed?
- What are the null and alternative hypotheses being tested in Part 9?
- How is the methodology implemented in the codebase?

2. Investment Universe Selection (15%)

- Describe the key institutional features of the stock exchange your team selected.
- Explain why you chose this market and how it supports your strategy.
- Provide an overview of your investment universe.

3. Results Interpretation and Evaluation (10%)

- Based on the output in Part 9, what can we infer from the average return and t-statistics of the trading strategy?

- Are the results statistically and economically meaningful?
- Do you consider the findings reliable? Why or why not?

4. Exploring Alternative Strategies (10%)

- Propose another investment strategy that you would consider exploring.
- Briefly justify your choice and discuss its potential advantages.

5. Outlier Detection in Stock Returns (25%)

Imagine your codebase was reviewed, and a key missing piece in the ETL process—**data cleaning**—was identified. Your managers have now asked you to implement this missing step.

- How do you detect outliers in stock return data?
- Why are your chosen methods appropriate for your investment universe?
- How do outliers affect financial data analysis and portfolio performance?
- Based on your analysis, does your investment universe contain outliers?

Hint: Avoid presenting exhaustive results for all stocks—select representative cases for both your report and presentation.

At the end of your presentation, your team will be asked **one random question (15%)**, which may relate to your teamwork. You will have **30 seconds to prepare** and **1 minute to respond**.

Points will be deducted if the presentation exceeds the allocated time limit. Time management is part of the assessment criteria.

Part 11: Project Report (30 marks)

Your report should serve as a leave-behind document—concise, yet informative. It should go beyond your presentation slides, providing a deeper and more detailed explanation of your analysis and methodology. The report should be no longer than **12 pages** and must address the **five core questions** listed above. Think of it as a comprehensive overview of your system, your findings, and the potential application of this tool in a real-world investment process.

Hint: Avoid presenting exhaustive results for all stocks—select representative cases for both your report and presentation.

Please note that the 11 parts above, totaling 100 marks, will contribute 30% to your overall score. An additional 5% is allocated to peer evaluation, both within and across groups. The peer evaluation form will be provided in Week 9.