**FinTech Unit 4 Homework Grading Rubric**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Criteria** | **Ratings** | | | | **Points** | **Feedback** |
| **Data Preparation**  **Requirements**  • Pandas is used to read each CSV file as a DataFrame  • Null values have been detected and removed  • Numeric values have been formated and data types converted  • S&P 500 closing prices have been converted to daily returns  • Whale Returns, Algorithmic Returns, and the S&P 500 Returns are joined into a single DataFrame with columns for each portfolio's returns | **20 Points Mastery**  **• Completed 5 out of 5 requirements**  **• Code runs without error and produces the assigned results**  **• Code accounts for all possible scenario**  **• Code is free of bugs** | **19 > 15 Points Approaching Mastery**  **• Completed 3 out of 5 of requirements**  **• Code runs without error**  **• Code produces results as expected 80% or more of the time** | **15 > 14 Points Progressing**  **• Completed 2 out of 5 requirements**  **• Code runs without error**  **• Code produces results, but not necessarily the correct results** | **14 > 0 Emerging**  **• Completed 1 or none out of the 5 requirements**  **• No submission**  **• Code runs with error** | **15** |  |
| **Quantitative Analysis**  **Performance Analysis Requirements**  • Calculate and plot daily and cumulative returns of all portfolios.  **Risk Analysis Requirements**  • Create a box plot for each of the returns.  • Calculate the standard deviationf or each portfolio.  • Determine which portfolios are riskier than the S&P 500  • Calculate the annualized standard deviation for each portfolio.  **Rolling Statistics**  • Calculate and plot the rolling standard deviation for all portfolios using a 21-day window.  • Calculate and plot the correlation between each stock to determine which portfolios may mimick the S&P 500.  • Choose one portfolio, then calculate and plot beta it and the S&P 60 500. | **20 Points Mastery**  **• Completed 8 out of 8 requirements**  **• Code runs without error and produces the assigned results**  **• Code accounts for all possible scenario**  **• Code is free of bugs** | **19 > 15 Points Approaching Mastery**  **• Completed 5 out of 8 requirements**  **• Code runs without error**  **• Code produces results as expected 80% or more of the time** | **15 > 14 Points Progressing**  **• Completed 3 out of 8 requirements**  **• Code runs without error**  **• Code produces results, but not necessarily the correct results** | **14 > 0 Emerging**  **• Completed 2 or fewer out of the 8 requirements**  **• No submission**  **• Code runs with error** | 15 |  |
| **Sharp Ratios**  • Using the daily returns, calculate the Sharpe ratios.  • Visualize the Sharpe ratios using a bar plot.  • Determine whether the algorithmic strategies outperform both the market (S&P 500) and the whales portfolios. | **15 Points Mastery**  **• Completed 3 out of 3 requirements**  **• Code Runs without error and produces the assigned results**  **• Code accounts for all possible scenario**  **• Code is free of bugs** | **14 > 10 Points Approaching Mastery**  **• Completed 2 out of 3 requirements**  **• Code runs without error**  **• Code produces results as expected 80% or more of the time** | **10 > 9 Points Progressing**  **• Completed 1 out of 3 requirements**  **• Code runs without error**  **• Code produces results, but not necessarily the correct results** | **9 > 0 Emerging**  **• Completed none or partial out of the 3 requirements**  **• No submission**  **• Code runs with error** | **14** |  |
| **Custom Portfolio**  Requirements  • Google Finance function is used to choose portfolio  • Data downloaded as CSV files and portfolio returns calculated  • Portfolio returns added to the DataFrame with the other portfolios analyzed and compared | **15 Points Mastery**  **• Completed 3 out of 3 requirements**  **• Code Runs without error and produces the assigned results**  **• Code accounts for all possible scenario**  **• Code is free of bugs** | **14 > 10 Points Approaching Mastery**  **• Completed 2 out of 3 requirements**  **• Code runs without error**  **• Code produces results as expected 80% or more of the time** | **10 > 9 Points Progressing**  **• Completed 1 out of 3 requirements**  **• Code runs without error**  **• Code produces results, but not necessarily the correct results** | **9 > 0 Emerging**  **• Completed none or partial out of the 3 requirements**  **• No submission**  **• Code runs with error** | **15** |  |
| **Coding Conventions/Formating** | **10 Points Mastery**  **• Imports are at the top of the file, just after any module comments and docstrings, and before module globals and constants.**  **• Function names are lowercase, with words separated by underscores**  **• Variable names follow the same convention as function names.**  **• Code follows (DRY) principals, no repetition, maintainable and highly reusable code.** | **9 Points Approaching Mastery**  **• Variable names are specific and descriptive of the information held by the variable**  **• Imports are within the top of file** | **8 Points Progressing**  **• Code lacks proper indentation and length convention**  **- Limit all lines to a maximum of 79 characters.**  **• Variable names are generic and not descriptive of the information held by the variable**  **• Imports and files are located in a non-standard location** | **8 > 0 Emerging**  **• Code is excessively lengthy**  **• Variable names are missing or lacking any descriptive information**  **• Import and files are not loaded** | 9 |  |
| **Deployment/Submission** | **10 Points Mastery**  **• Repository cloned to local machine**  **• Files added to the repo via the command line**  **• Appropriate commit messages** | **9 Points Approaching Mastery**  **• Repository cloned to local machine**  **• Files added to repo via the command line** | **8 Points Progressing**  **• Repository created on GitHub**  **• Files added manually on GitHub** | **8 > 0 Emerging**  **• No Submission**  **• Submission via incorrect format** | 9 |  |
| **Documentation/Comments** | **10 Points Mastery**  **• Code is well commented with concise, relevant notes** | **9 Points Approaching Mastery**  **• Code is commented and mostly understandable to an outside user** | **8 Points Progressing**  **• Code has comments, but they are not understandable to an outside user** | **8 > 0 Emerging**  **• Code is not commented** | 9 |  |