

[Return to Classroom](#)[DISCUSS ON STUDENT HUB](#)

Multifactor Model

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Great job building a Multifactor Model! 👍

There are many great quantitative finance books out there. [This one](#) provides a great introduction that could be a useful companion for the rest of the program.

Congratulations on finishing the first term and all the best for the rest of your learning journey!

Statistical Risk Model

The function `fit_pca` fits the PCA model with returns.

The function `fit_pca` correctly fits the PCA model with returns. Good job!

Just to recap a bit here: PCA is a form of unsupervised learning where you are trying to maximize the variances in order to determine feature importance. Here is a great [step-by-step explanation of PCA](#) for your review.

The function `factor_betas` gets the factor betas from the PCA model.

Good work getting the factor betas from the PCA model!

The function `factor_returns` gets the factor returns from the PCA model.

The function `factor_returns` correctly returns the factor returns from the PCA model. Nice job!

The function `factor_cov_matrix` gets the factor covariance matrix.

Great use of `np.diag` to return the factor covariance matrix!

Note that for any covariance matrix the main diagonal are always the variances - see [here](#).

The function `idiosyncratic_var_matrix` gets the idiosyncratic variance matrix.

The function `idiosyncratic_var_matrix` correctly gets the idiosyncratic variance matrix. Excellent use of numpy matrix algebra!

The function `idiosyncratic_var_vector` gets the idiosyncratic variance vector.

Great job computing the idiosyncratic_var_vector!

The function `predict_portfolio_risk` gets the predicted portfolio risk.

Create Alpha Factors

The function `mean_reversion_5day_sector_neutral` generates the mean reversion 5 day sector neutral factor.

The function `mean_reversion_5day_sector_neutral` correctly generates the mean reversion 5 day sector neutral factor. Simple and correct code!

The function `mean_reversion_5day_sector_neutral_smoothed` generates the mean reversion 5 day sector neutral smoothed factor.

Great job calculating the smoothed factor! Zipline is always not so intuitive but very powerful when used in the right hands.

Evaluate Alpha Factors

The function `sharpe_ratio` gets the sharpe ratio for each factor for the entire period.

That's right! You'll notice that the smoothed factors have a lower sharpe ratio.

The student correctly mentions what would happen if you smooth the momentum factor and why.

Student Answer:

Sharpe Ratio of 0.98 for momentum factor is good but if we look at the **Factor Rank Autocorrelation** plots, FRA for momentum factor looks stable. So smoothing the momentum factor will not have any significant change.

That's right! There will be no major change in performance as a result of smoothing the momentum factor.

Some important facts to keep in mind here are:

Smoothing: Smoothing means taking weighted average of portfolio weights over multiple days so as to reduce our turnover and thus our transaction costs. Our full service brokerage charges commissions (and not flat fees) and thus if we trade in and out a lot, we would spend a lot in brokerage charges. Thus even if we save 100 to 200 basis points (1%-2%) in transaction costs, we can easily make up our management fees of AUM which over the long run on \$5 billion (WorldQuant's AUM is a very significant amount).

Performance: We measure our risk adjusted returns as sharpe which takes volatility into consideration. The most important thing we have to remember here is that the performance is given if we had employed this strategy in the past and thus it does not guarantee future returns. We usually formulate alpha factors by some intuition and then frame the raw alpha. Then we try to perform fine tuning. We have to be very careful while fine tuning and shouldn't rely excessively on past performance since that will make our model to over-fit the data which we have leading to historical bias. Read more about this here: <https://mathinvestor.org/2018/04/the-most-important-plot-in-finance/>

To conclude, we smooth out to reduce our transaction costs irrespective of whether the sharpe increases or decreases.

Optimal Portfolio Constrained by Risk Model

The function `OptimalHoldings._get_obj` returns the correct objective function.

You returned the correct objective function.

The function `OptimalHoldings._get_constraints` returns the correct list of constraints.

The function `OptimalHoldings._get_constraints` correctly returns the correct list of constraints. Excellent!

The function `OptimalHoldingsRegualization._get_obj` returns the correct objective function.

Looks good. The diversification of the holdings has improved!

The function `OptimalHoldingsStrictFactor._get_obj` returns the correct objective function.

Awesome job with the objective `OptimalHoldingsStrictFactor._get_obj` function to wrap up the project!

 [DOWNLOAD PROJECT](#)

[RETURN TO PATH](#)

Rate this review

START