Dataset:

Choose CSI300 as our stock pool, and download the data of stocks from BaoStock. For the convenience of processing, we will convert all data into Dataframe format, with the factor format as follows:

Factor_df: {

Index = dates,

Columns = codes,

Values = factor

}

Example: Part of factor_avgturn5

date	sh.600000	sh.600009	sh.600010	sh.600011	sh.600015
2017/6/7	0.2511	0.8114	0.3030	0.2043	0.3103
2017/6/8	0.2169	0.8453	0.2599	0.2203	0.2749
2017/6/9	0.2413	0.9824	0.2643	0.2304	0.3043
2017/6/12	0.2398	1.0236	0.2628	0.2210	0.2799
2017/6/13	0.2516	1.0028	0.2476	0.1998	0.2838

Backtest Criteria:

To test the effectiveness of our factors, we introduced evaluation criteria such as IC value, RankIC value, IR value, Sharpe ratio, and position return rate.

Before calculating, we first read in the factor data and turnover rate data.

```
# 读取数据
factor_df = pd.read_csv('/Users/zhuyining/PycharmProjects/pythonProject/factor_TVSTD6.csv', index_col=0)
pctChg = pd.read_csv('/Users/zhuyining/PycharmProjects/pythonProject/stock_data_pctChg.csv', index_col=0)
```

（1）IC value

IC value is an indicator used in stock selection strategies to evaluate the effectiveness of factors, known as Information Coefficient. The IC value measures the correlation between stock selection factors and future stock returns, typically ranging from -1 to 1.

The calculation method of the IC formula is as follows:

$$IC = \frac{\text{Covariance of Model's Predictions and Actual Results}}{\text{Standard Deviation of Model's Predictions} \times \text{Standard Deviation of Actual Results}}$$

Specifically:

Covariance of Model’s Predictions and Actual Results represents the covariance between the model’s predictions and the actual results.

Covariance is a measure of the relationship between two variables, indicating how they vary together.

Standard Deviation of Model’s Predictions represents the variability of the model’s predictions, i.e., the degree of dispersion in the predictions.

Standard Deviation of Actual Results represents the variability of the actual results, i.e., the degree of dispersion in the observed outcomes.

The analysis of IC values is crucial for optimizing and improving stock selection strategies. If the IC value of a certain factor is high, it indicates that the factor has good ability to predict future stock performance and can be used as an effective stock selection indicator. On the contrary, if the IC value is low, it maybe necessary to reconsider or conduct in-depth research on the effectiveness of this factor.

（2）IR value

IR value is an indicator in portfolio management, commonly known as Information Ratio. It is used to measure the contribution of a factor in an investment portfolio to excess returns, and to standardize this contribution, taking into account the risks associated with that factor.

The calculation method for Information Ratio (IR) is as follows:

$$\mathrm{IR} = \frac{\text{Active Return}}{\text{Active Risk}}$$

Among them:

Active Return of the Portfolio: The excess return of an investment portfolio, which is the difference between the actual return of the portfolio and the benchmark return.

Active Risk of the Portfolio: The volatility of the portfolio relative to the benchmark, which is the standard deviation between the portfolio return and the benchmark return.

IR value can be seen as the ratio between the IC value and the factor volatility.

$$\mathrm{IR} = \frac{\mathrm{IC} \times \text{factor volatility}}{\text{Overall volatility}}$$

Overall volatility refers to the volatility of the entire investment portfolio.

In short, the IC value measures the correlation between factors and future returns, while the IR value takes into account this correlation and combines it with factor volatility and overall volatility, providing a more comprehensive assessment of excess returns. The following is the code we use to calculate the IC and IR values:

```python
# 计算IC值和IR值
IC_series = factor_df.shift().corrwith(next_day_ret_df, axis=1)
IC = IC_series.mean()
IR = IC_series.mean()/IC_series.std()
print("IC: ", IC)
print("IR: ", IR)
```

（3）RankIC and RankIR value

RankIC is a metric commonly used in the financial domain to assess the performance of an investment model. It is similar to the Information Coefficient (IC) but takes into account the ranking of model predictions.

The formula for RankIC is as follows:

$$\text{RankIC} = \frac{\text{Spearman Rank Correlation Coefficient}}{\text{Number of Observations}}$$

In this formula:

Spearman Rank Correlation Coefficient represents the Spearman rank correlation between the ranks of the model predictions and the ranks of the actual results. It measures the monotonic relationship between these two sets of ranks, regardless of the specific numerical values.

Number of Observations denotes the total number of observations, indicating the size of the sample data.

The RankIC typically ranges between -1 and 1, similar to a regular correlation coefficient. A positive value indicates a positive correlation between the rank of predictions and the rank of actual results, a negative value indicates a negative correlation, and values close to zero suggest a weak relationship between the ranks.

RankIC is useful for assessing a model’s ability to predict relative rankings, especially in cases where the specific numerical values of model outputs maybe subject to inaccuracies or scale issues.

In stock selection strategies, RankIR value is an indicator that evaluates the contribution of factors to excess returns, similar to traditional IR ratio, but it focuses on the ability to rank factor values rather than the absolute predictive ability of specific values.

Steps for calculating RankIR values:

1. Factor Value Ranking: Ranks the values of stock selection factors in the stock pool, which can be arranged in ascending or descending order.

2. Investment portfolio construction: Construct an investment portfolio based on the ranking of factor values, such as selecting stocks with a certain proportion of ranking in the top.

3. Calculate excess return: Calculate the excess return of the investment portfolio, which is the difference between the actual return of the investment portfolio and the benchmark return.

4. Calculate Risk: Calculate the volatility of the investment portfolio relative to the benchmark, that is, the standard deviation between the portfolio return and the benchmark return.

5. Calculate RankIR value: RankIR value is the ratio of excess return to risk of the investment portfolio, which is constructed based on the ranking of factor values.

$$\text{RankIR} = \frac{\text{Active Return of the Portfolio (based on factor ranking)}}{\text{Active Risk of the Portfolio (based on factor ranking)}}$$

The RankIR value measures the correlation between the ranking of factor values and the excess returns of investment portfolios. A high RankIR value indicates that the ranking ability of factor values contributes significantly to generating excess returns, while the risk associated with ranking is lower.

The following is the code we use to calculate the RankIC and RankIR values:

```
# 计算RankIC值和RankIR值
RankIC_series = factor_df.shift().corrwith(next_day_ret_df, axis=1, method='spearman')
RankIC = RankIC_series.mean()
RankIR = RankIC_series.mean()/RankIC_series.std()
print("RankIC: ", RankIC)
print("RankIR: ", RankIR)
```

（4）Sharpe Ratio

The Sharpe Ratio is an indicator used to evaluate the risk adjusted return of an investment portfolio, and is also commonly used to evaluate the performance of factors in stock selection strategies. The Sharpe ratio measures the excess return a portfolio receives for each unit of overall risk it undertakes.

The formula for calculating the Sharpe ratio is as follows:

$$\text{Sharpe Ratio} = \frac{\text{Portfolio Return} - \text{Risk-Free Rate}}{\text{Portfolio Volatility}}$$

Among them:

Portfolio Return: Expected return of the investment portfolio.

Risk Free Rate: Risk free rate, usually short-term treasury bond bond rate.

Portfolio Volatility: The standard deviation of a portfolio measures the volatility of portfolio returns.

In stock selection strategies, the Sharpe ratio can be used to evaluate the risk adjusted return performance of factors. If the Sharpe ratio of a stock selection factor is high, it indicates that the factor has a better adjustment effect relative to risk when obtaining excess returns, and maybe an effective stock selection factor. On the contrary, a lower Sharpe ratio may indicate that the returns of the factor are insufficient to compensate for the corresponding risks.

It should be noted that the limitation of the Sharpe ratio is that it assumes that the return follows a normal distribution, while the actual return distribution in the market may have spikes or skewness. Therefore, when using the Sharpe ratio, investors should carefully consider its limitations.

The following is the code we use to calculate the Sharpe Ratio values:

```
# 计算夏普比率
Daily_pnl.mean()*242
Sharpe = Daily_pnl.mean()/Daily_pnl.std()*np.sqrt(242)
print("Sharpe ratio: ", Sharpe)
```

（5）Position return image

We hope to evaluate factors by simulating the use of a single factor value to obtain our purchasing strategy and calculating daily and long returns.Firstly, we need topreprocess the factor values. As the factor values are greater than 0 but less than 1, we set the factor value to -0.5 to obtain a factor value between -0.5 and 0.5.

Subsequently, we obtained our position by standardizing the factor values for the positive and negative parts of the factor values separately. Among them, we long the parts with factor values greater than zero and short the parts with factor values less than zero.

Due to the prohibition of short selling in the A-share market, we pay more attention to long returns and the gap between long returns and daily returns.

The following is the code for calculating positions, daily returns, long returns, and drawing a position return image:

```
#### 由因子值计算仓位和 ####
# 计算仓位
Factor_rk_df = factor_df.rank(axis=1,pct=True)-0.5
Pos = Factor_rk_df.shift()
Pos[Pos>0] = Pos[Pos>0].div(Pos[Pos>0].sum(axis=1),axis=0)
Pos[Pos<0] = Pos[Pos<0].div(Pos[Pos<0].abs().sum(axis=1),axis=0)
next_day_ret_df = pctChg.shift(-1)

# 计算日收益
Daily_pnl = (Pos * next_day_ret_df).sum(axis=1) / 2
# 多头收益
Daily_long_pnl = (Pos[Pos > 0]*next_day_ret_df).sum(axis=1) - next_day_ret_df.mean(axis=1)

# 画图
Daily_pnl.cumsum().plot()
Daily_long_pnl.cumsum().plot()
```