Convertible Bond Project Report

Yifan Zhang | 2021.09

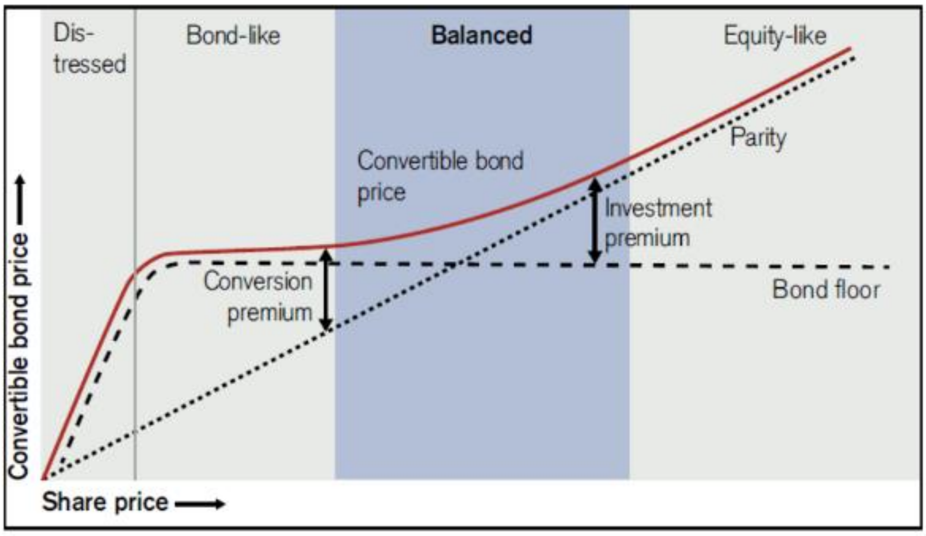
project code - gitlab

Abstract: This report introduces the main work of the convertible bond project. We first give a brief introduction about the convertible bond in Section 1. Next is the data overview and data processing, as well as the basic statistics of our dataset. Section 3 introduces the backtest framework designed for convertible bond trading strategies. Then we start exploring different kinds of trading ideas of convertible bond, including negative conversion premium arbitrage, overnight strategy, mean reversion, and intraday strategy. These topics will be discussed in detail from Section 4 to Section 6. Section 7 make conclusions of the whole project and talk about future works.

1. Introduction of Convertible Bond

1.1. Definition of Convertible Bond

A convertible bond is a fixed-income corporate debt security that yields interest payments, but can be converted into a predetermined number of common stock or equity shares. Convertible bond can be categorized into 4 types based on its value: distressed, bond-like, balanced and equity-like.



- **Equity-like (In the Money):** In the money means the share price trades above the conversion price, conversion takes places with high probability and the conversion premium is low. The price of CB is more sensitive to share price.
- **Balanced (At the Money)**: At the money means the share price trades close to the conversion price. The price of CB is sensitive to both the share price as well as interest rate and credit spread.

- **Bond-like (Out of The Money)**: Out of the money means the share price trades below the conversion price, the conversion is unlikely to happen and the investment premium is low. The price of CB is more sensitive to interest rate and credit spread.
- **Distressed**: Convertible bond is distressed when the share price is sufficiently low and the default risk of CB is high. The value of CB converges to parity or the recovery value.

1.2 Convertible Bond Terms

Except for the common bond's terms (interest rate, coupon, maturity), most of the convertible bonds have non-standardized terms as listed below:

- **Conversion Ratio / Conversion Price**: Conversion ratio is the number of stock shares each bond can be converted. Typically it is fixed throughout bond lifetime. Conversion price equals to the nominal value divided by conversion ratio.
- **Reset Clause**: Reset clause is commonly used to adjust conversion ratio. It usually happens when the stock price remains under a certain level for a long time. The firm can adjust the conversion ratio upwards to protect the investors.
- **Redemption Ratio**: It specifies what percentage of the bond's face value will be redeemed at maturity.
- **Call Provision**: Call provision gives the issuer the right to end the CB before maturity by forcing the investor either to convert the bond or redeem the bond at strike price.
- **Soft Call Clause**: The call provision is triggered only when specified conditions are met.
- **Put Provision**: Put provision gives the investor the right to end the CB before maturity by forcing the issuer to buy the bond back at strike price. The purpose is to protect investors from downside volatility.

reference: Convertible Bond Pricing: A Monte Carlo Approach

2. Data Overview & Preparation

This part will introduce the detailed data preparation work for the research project. Readers can find the main work of this part in Data Processing.ipynb.

2.1 Raw Data & Processing

The raw data includes 1) company information, 2) convertible bond items like issurance, redemption, repurchase and conversion, 3) convertible bond trading data. The notebook walked through each dataset, processed data cleaning and re-organized them into easy-to-use data structure.

As for the stock data, we transfer the data into dictionary. The dictionary key is the trading date and the value is a cross-sectional table of each stock.

For the purpose of research and strategy design, we dropped the data that satisfies any of the following situations:

- We use company identity code to concatenate bond and stock. A few samples have missing stock ticker or bond ticker, thus impossible to link them.
- Some convertible bonds have zero trading volumes during most of its lifetime. These CB have few liquidity and thus should not be considered as tradable asset.
- We use redemption begin date and redemption end date to identify the convertible period of convertible bond. Any samples have missing data on these two features are dropped.

After the data cleaning, the trading period of our data is from 2017/01/03 to 2020/08/21, and the total number of convertible bonds is 359.

2.2 Data Problem & Solutions

There exist several data problems in our dataset. This part will discuss 2 main problems and the solutions.

A. Data quality of stock data

The source of stock data is Bloomberg while the source of convertible bond is Wind. We found that the stock data from Bloomberg have at least 3 problems which may have negative impact on the further analysis.

- The stock dataset contains records on non-trading days. For example, 2017/01/27 is the first day of Chinese Spring Festival. The stock dataset keeps all the calendar business days by forward fill the ohlc data. This will lead to wrong calculation of rolling return and volatility. Solution to this problem is to re-index the trading dates of stock data using convertible bond trading dates.

- Because of dividend payment and stock split/combination, the stock price should be adjusted to avoid historical price jump. However, the adjust factor provided by Bloomberg cannot perfectly match with the price change. Based on several case studies, we found that the adjusted return (7th column in 'adj2' dataset) lagged one day and the adjusted factor (13th column in 'adj2' dataset) lagged two days. The solution is group the dataset by ticker and then shift (-1) and shift(-2) respectively for these two columns.

- Two stocks (688023.SH and 688021.SH) have two rows in May 2020 each day and one of them have negative stock share. These redundant and meaningless data should be deleted.

B. Missing Historical Conversion Price

The convertible bond data starts from 2017/01/03. However, some convertible bonds announced their conversion price before 2017. Since the further research need historical conversion price to calculate the conversion premium, we have to supplement the historical records. The solution is to collect data through web crawler on Sina Finance. The code can be found in SinaFinance.ipynb.

2.3 Cleaned Data Sample

In this part we reorganized the raw data and generated 3 integrated datasets named cbond_info, cbond_price, and stock_data.

- **convertible bond basic information**: includes tickers and item-related features.

	bond_ticker	stock_ticker	name	listed_date	start_date	exit_date	conv_date	conv_price	rdmpt_bgndt	rdmpt_enddt	rdmpt_transit
0	110030.SH	600185.SH	格力地产	20150113	20170103	20191210	20141223	20.90	20150630	20191224	130.0
1	110030.SH	600185.SH	格力地产	20150113	20170103	20191210	20160526	7.39	20150630	20191224	130.0
2	110030.SH	600185.SH	格力地产	20150113	20170103	20191210	20160825	7.26	20150630	20191224	130.0

rpchs_bgndt	rpchs_enddt	rpchs_transit	term_year	interest_freq	payment	put_win	put_price	call_win	call_min	call_price
20161225	20191225	70.0	5.0	1	106.0	30	103	30	15	100
20161225	20191225	70.0	5.0	1	106.0	30	103	30	15	100
20161225	20191225	70.0	5.0	1	106.0	30	103	30	15	100

- **convertible bond trading data**: includes ohlc data, basic statistics and status label.

bond_ticker	date	open	high	low	close	volume	amount	pre_close	ovnt_ret	intra_ret	return	next_return	is_last
110033.SH	20170103	114.820	115.800	114.820	115.320	3046.0	3513.594	NaN	NaN	0.004355	NaN	0.007371	False
128011.SZ	20170103	128.000	128.780	127.654	127.654	897.0	1149.171	NaN	NaN	-0.002703	NaN	0.002703	False
110031.SH	20170103	108.790	109.380	108.400	108.500	7584.0	8258.409	NaN	NaN	-0.002666	NaN	0.001106	False

- **stock trading data**: includes ohlc data, basic statistics and status label.

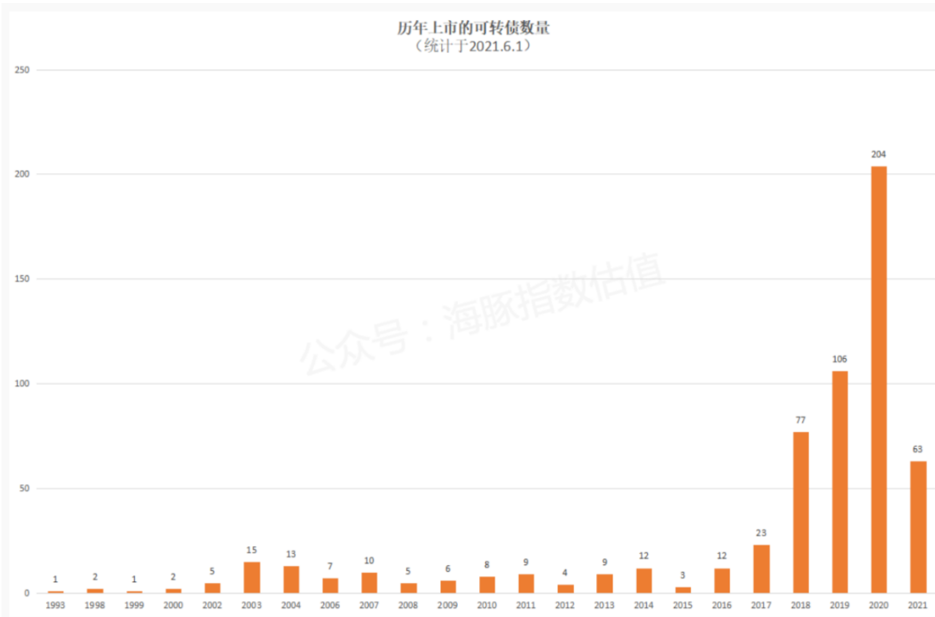
BloombergID	stock_ticker	date	adj_close	adj_open	adj_pre_close	adj_factor	vol100	vol250
EQ0000000025800901	600959.SH	20170103	14.774415	14.735121	14.800611	0.763482	0.010620	0.010620
EQ0000000001131660	002001.SZ	20170103	143.885410	141.219532	141.219532	0.138791	0.016235	0.016235
EQ0730727200001000	600380.SH	20170103	29.018077	28.786396	28.757436	0.345302	0.023298	0.023298

value	mkt	top1500	CSI300	CSI800	limit_buy_open	limit_sell_open	limit_buy_close	limit_sell_close	is_unusual
1.518388e+08	4.393403e+10	True	True	True	False	False	False	False	False
2.014444e+08	2.153882e+10	True	False	True	False	False	False	False	False
5.276599e+07	1.584759e+10	False	False	True	False	False	False	False	False

2.4 Basic Statistics

The convertible bond market differs a lot from the A-share stock market. Here we present some basic statistics of convertible bond market to help the readers get familiar with it.

Convertible bond market in China developed rapidly especially after 2018. Hence, the sample distribution is imbalanced over years in our dataset. The following table shows the number of convertible bonds each year in our dataset (359 in total, some may overlap in different years). The image below represents the total number of newly listed convertible bonds in each year, with a significant growing trend from 2018.



source: *集思录*

2017	2018	2019	2020
37	111	215	329

Next is the comparison between convertible bond market and stock market. The table below shows the daily trading value of these two market in our dataset (CB and stock pair-matched). The median trading value in CB market is around 10 million, the 90% percentile is around 0.1 billion and 99% percentile is around 1 billion. The most actively traded convertible bond is at the same level with the most actively traded one in stock market while most of the convertible bonds are less active than stock significantly.

Daily Trading Value (亿, 0.1B)	Convertible Bond Market	Stock Market
mean	0.74	2.12
std	3.29	4.80
25%	0.04	0.33
50%	0.11	0.82
75%	0.34	2.13
max	164.71	172.04

As for the market return, the average close to close daily return between convertible bond market and stock market shares similarities but distinguished from each other when we split the daily return into intraday and overnight return. Two takeaways from the following table: 1). Stock market has a more significant difference between intraday and overnight return. 2). The return of convertible bond has large positive skew, implying much more extreme positive return happens in this market.

daily return	intra_return (CB)	intra_return (stock)	ovnt_return (CB)	ovnt_return (stock)
mean	0.03%	0.15%	0.06%	-0.07%
min	-32.93%	17.75%	-30%	-16.94%
25%	-0.52%	-1.19%	-1.06%	-0.47%
50%	0.01%	0.00%	0.00%	0.00%
75%	0.56%	1.31%	0.17%	0.38%
max	78.73%	20.42%	30.04%	10.14%
skew	3.49	-0.51	2.32	0.53

Finally, let's have a look on the stock index label. In our dataset, around 50% samples are among top 1500 stock, 20% in CSI800 and 8% in CSI300. We take top 1500 as the standard convertible bond pool in the upcoming strategy implementation.

3. Backtest Framework & Strategy Design

For the purpose of convertible bond strategy backtesting, this project designed two python classes, CbondBacktest and Strategy. The backtest framework supports daily trading of convertible bond and stock when market open or close. Readers can find the implementation of Backtest class in Backtest.ipynb. The basic backtest parameters including data path, trading period, initial capital, available trading time, trading cost, etc are integrated in backtest_config.py and is shown in the image below.

```
# config file for Backtest
data_path = '../../../../export/scratch/for_yifan/research/'

start_date = '20170103'          # min date is 20170103
end_date = '20200821'            # max date is 20200821

# strategy param
capital = 100000000              # total capital
times = ['open', 'close']        # available time to trade

longshort = {
        'bond':[1],              # long only
        'stock':[1, -1]          # long & short
}

fee_rate = {
        'bond':0.0001,           # 0.01% transaction cost (buy/sell)
        'stock':0.0001,          # 0.01% transaction cost (buy/sell)
        'lending': 0.1,          # stock annual lending cost
        'selling': 0.001,        # 0.1% stock selling tax
}

cbond_min_share = 10             # minimum trading volume of convertible bond
stock_min_share = 100            # minimum trading volume of stock

valid_both = True                # required 'valid' status for both convertible bond and stock if trade any of them

# analysis param
annual_date = 250                # annual trading day
riskfree_rate = 0.04             # annual risk-free rate
```

Strategy class is designed for the sake of flexibility. Strategy class is called in CbondBacktest pipeline when market open or close every trading day. CbondBacktest provides Strategy class APIs to request historical pnl and portfolio as well as APIs to buy/sell/convert. The backtest framework acts as a security broker and the strategy class works as a signal generator and portfolio manager.

After the pipeline, CbondBacktest will output strategy performance analysis. The analysis mainly contains 4 parts:

- **P&L statistics**: annual return, annual volatility, sharpe ratio, winning rate and max draw down will be calculated based on several P&L time series. The P&L type includes:
  - total pnl (pnl)
  - pnl after open (pnl_open)
  - pnl after close (pnl_close)
  - pnl from trading (pnl_trade)
  - pnl from conversion (pnl_trans)
  - intraday pnl (pnl_intra)
  - overnight pnl (pnl_ovnt)
  - hedged pnl, hedge CB holdings using 'top1500' (pnl_hedge)
  - net pnl (net)
  - net hedged pnl (net_hedge)

- **Order & Portfolio statistics**: This part aims to present more detailed analysis about order and portfolio. The statistics below are calculated separately according to the asset type (bond/asset/total).
  - total trading dates available (dates)
  - trading dates of the strategy, % of total available days (trade_dates)
  - total number of order (order_sum)
  - coverage number of assets (cover_num)
  - winning rate of each order (trade_win)
  - winning rate of each conversion (conv_win)
  - average invested capital in trading day (order_cost)
  - order rejection rate (refuse_rate)
  - portfolio annual turnover (turnover)
  - average daily holding number (port_num)
  - average daily holding value (port_value)
  - portfolio long short ratio (long_ratio)

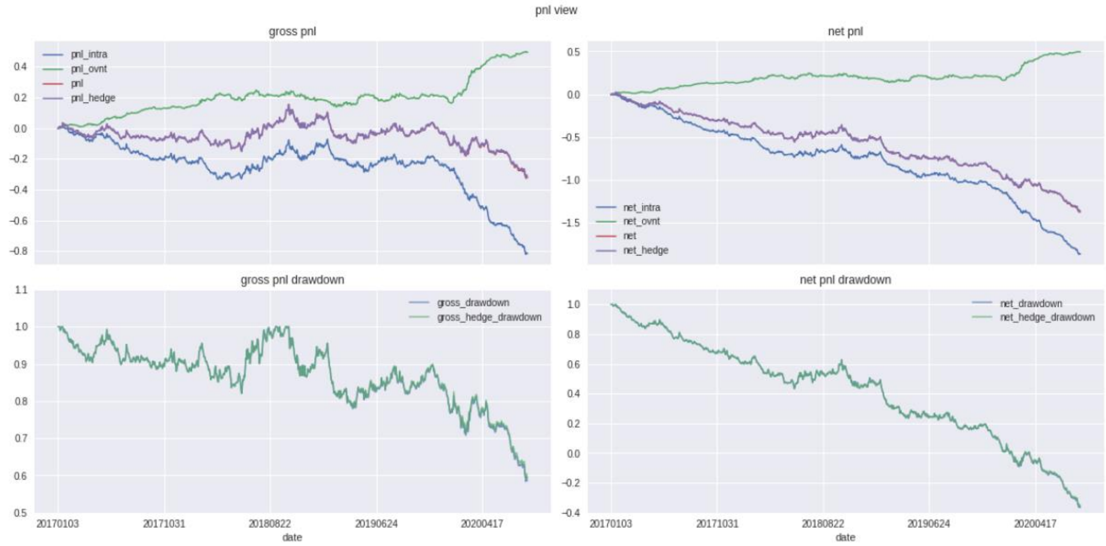
- **P&L view**: Gross and net P&L time series plotting and historical max drawdown.

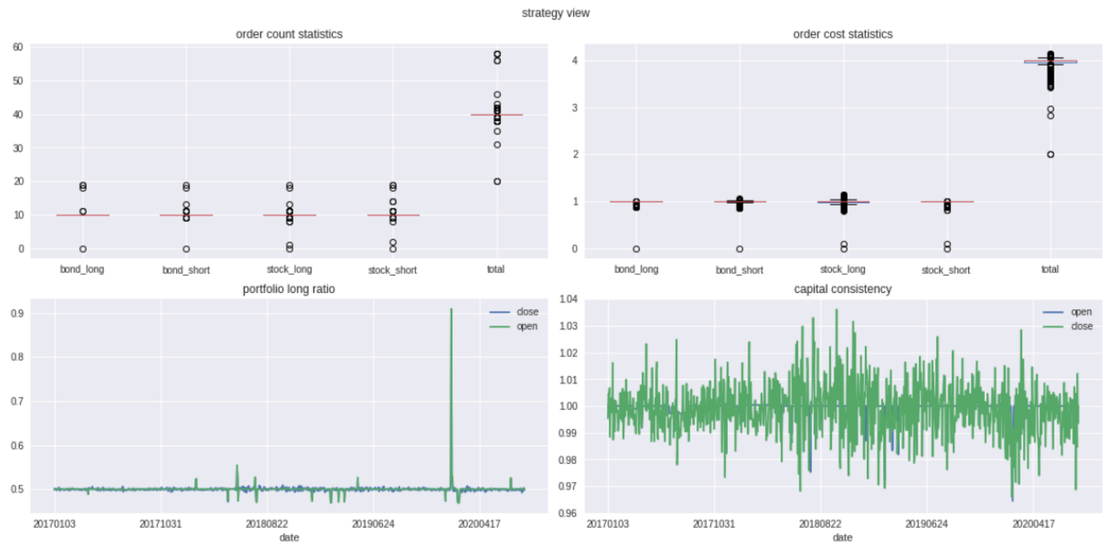
- **Strategy view:** Order statistics, portfolio long-short ratio and capital consistency visualization.

Here we design a random selecting strategy as a sample to show the backtest analysis. All the strategy testing code can be found in Strategy.ipynb.

	total_return	annual_return	annual_volatility	sharpe_ratio	win_rate	max_drawdown
pnl	-0.3231	-0.0912	0.1716	-0.7643	0.4695	-0.4161
pnl_open	0.6965	0.1965	0.0778	2.0110	0.5609	-0.0617
pnl_close	-1.0196	-0.2877	0.1517	-2.1609	0.4447	-1.0243
pnl_trade	-0.1192	-0.0336	0.1692	-0.4353	0.4831	-0.2940
pnl_trans	-0.2039	-0.0575	0.0009	-103.8063	0.0011	-0.2037
pnl_intra	-0.8156	-0.2301	0.1517	-1.7812	0.4549	-0.8245
pnl_ovnt	0.4925	0.1390	0.0778	1.2717	0.5339	-0.0877
pnl_hedge	-0.3144	-0.0887	0.1723	-0.7471	0.4695	-0.4076
net	-1.3729	-0.3874	0.1716	-2.4911	0.4300	-1.3647
net_hedge	-1.3643	-0.3849	0.1722	-2.4671	0.4300	-1.3562

	dates	trade_dates	order_sum	order_mean	cover_num	trade_win	conv_win	order_cost	refuse_rate	turnover	port_num	port_value	long_ratio
bond	886	1.0000	17749.0000	20.0327	325	0.4820	0.0038	1.9678	0.0089	245.9755	10.1524	0.9985	0.9994
stock	886	1.0000	17724.0000	20.0045	319	0.4913	0.0038	1.9634	0.0112	245.4229	10.1603	0.9987	0.0006
total	886	1.0000	35473.0000	40.0372	644	0.4866	0.0038	3.9312	0.0100	491.3984	20.3126	1.9972	0.5002





Note that the total capital maintains to be the initial capital each time the strategy clear the holdings. The image 'capital consistency' above shows the total value of assets (portfolio + cash) over time and it should always be around 1.

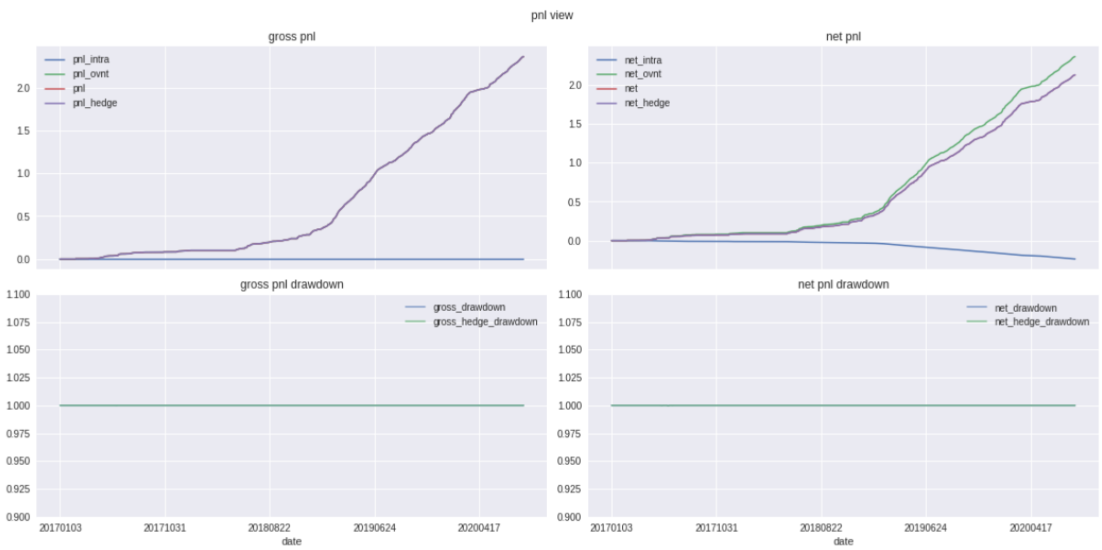
4. Negative Premium Arbitrage

We start exploring different types of trading ideas and design profitable strategies from this part.

The first idea comes from the uniqueness of convertible bond -- investors can convert it to certain amount of stock at any time during the conversion period (generally 5.5/6 years, which is a considerably large time range). One can calculate the premium of convertible bond, which is, the trading price of convertible bond over its conversion value (the conversion ratio multiplies by the corresponding stock trading price). In most cases, convertible bond is traded with a positive premium when the convertible bond is in the conversion period and the premium can be recognized as the 'premium of call option'. However, there exists around 4% (3900 / 98000) opportunities of negative premium when the market close. In that case, the negative premium arbitrage strategy is designed to capture the risk-free profitable trading opportunities caused by limited efficiency between convertible bond and stock market. To be specific, when we find a negative premium CB - stock pair, buy the convertible bond and short the stock simultaneously with delta = 1 (the portfolio becomes zero after conversion). The long-short strategy locks the profit at the moment we open the position and thus is a risk-free trading strategy. A threshold of negative premium can be set to guarantee that the profit can cover both the transaction cost and the lending cost of stock.

	total_return	annual_return	annual_volatility	sharpe_ratio	win_rate	max_drawdown
pnl	2.3644	0.6672	0.0511	12.2808	0.9977	-0.0001
pnl_open	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
pnl_close	2.3644	0.6672	0.0511	12.2808	0.9977	-0.0001
pnl_trade	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
pnl_trans	2.3644	0.6672	0.0511	12.2808	0.9977	-0.0001
pnl_intra	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
pnl_ovnt	2.3644	0.6672	0.0511	12.2808	0.9977	-0.0001
pnl_hedge	2.3644	0.6672	0.0511	12.2808	0.9977	-0.0001
net	2.1284	0.6006	0.0479	11.7077	0.9932	-0.0003
net_hedge	2.1284	0.6006	0.0479	11.7077	0.9932	-0.0003

	dates	trade_dates	order_sum	order_mean	cover_num	trade_win	conv_win	order_cost	refuse_rate	turnover	port_num	port_value	long_ratio
bond	886	0.6625	3473.0000	5.9165	156	0	0.9718	0.6667	0.0000	55.2165	0.0000	0.0000	0.5000
stock	886	0.6625	3473.0000	5.9165	153	0	0.9718	0.6708	0.0000	55.5500	0.0000	0.0000	0.5000
total	886	0.6625	6946.0000	11.8330	309	0	0.9718	1.3375	0.0000	110.7665	0.0000	0.0000	0.5000



Above is the backtesting result of the negative premium arbitrage strategy. The premium threshold is -0.1% and the asset pool is 'all'.

Next we loop over the negative premium threshold and different asset pools to test the strategy sensitivity. Reasonably, the smaller the asset pools, the less trading opportunities we have. Meanwhile, the strategy performance sharply decreased when the pool is 'CSI800' or 'CSI300'. Hence, we mainly focus on pool 'all' and 'top 1500' in the further analysis. As for the premium threshold, there exists optimal parameters (-0.1% to -0.5%) when pool is 'top1500'.

pool = 'all'

	premium	trade_pct	gross_pnl	net_pnl	net_hedge_pnl	sharpe	order_mean	order_cost	turnover
0	0.0000	0.6840	0.6287	0.5597	0.5597	11.4450	12.5644	1.3435	114.8662
1	-0.0010	0.6625	0.6672	0.6006	0.6006	11.7077	11.8330	1.3375	110.7665
2	-0.0020	0.6264	0.7072	0.6455	0.6455	11.6698	10.7964	1.3104	102.6047
3	-0.0030	0.5756	0.7334	0.6790	0.6790	11.2675	9.6157	1.2560	90.3728
4	-0.0040	0.5350	0.7401	0.6919	0.6919	10.7731	8.5274	1.1970	80.0497
5	-0.0050	0.4876	0.7228	0.6816	0.6816	10.0151	7.5602	1.1216	68.3603
6	-0.0060	0.4549	0.7025	0.6665	0.6665	9.4204	6.6700	1.0513	59.7728
7	-0.0070	0.4176	0.6695	0.6384	0.6384	8.8254	6.0000	0.9891	51.6304
8	-0.0080	0.3691	0.6249	0.5987	0.5987	8.1086	5.5046	0.9401	43.3692
9	-0.0090	0.3341	0.5842	0.5619	0.5619	7.5222	5.0068	0.8880	37.0852
10	-0.0100	0.3047	0.5513	0.5318	0.5318	7.0550	4.6148	0.8487	32.3286

`

pool = 'top1500'

	premium	trade_pct	gross_pnl	net_pnl	net_hedge_pnl	sharpe	order_mean	order_cost	turnover
0	0.0000	0.6682	0.5552	0.4907	0.4907	10.9829	9.9797	1.2857	107.3819
1	-0.0010	0.6445	0.5821	0.5207	0.5207	11.2297	9.4116	1.2680	102.1477
2	-0.0020	0.6084	0.6067	0.5508	0.5508	11.1286	8.5417	1.2217	92.9057
3	-0.0030	0.5497	0.6171	0.5688	0.5688	10.5977	7.6345	1.1675	80.2143
4	-0.0040	0.5068	0.6104	0.5684	0.5684	10.0666	6.7350	1.1032	69.8844
5	-0.0050	0.4481	0.5768	0.5419	0.5419	9.2620	6.0957	1.0348	57.9608
6	-0.0060	0.4131	0.5451	0.5154	0.5154	8.6273	5.4044	0.9572	49.4240
7	-0.0070	0.3691	0.5057	0.4808	0.4808	7.9198	4.9113	0.8972	41.3911
8	-0.0080	0.3228	0.4570	0.4367	0.4367	7.1618	4.4476	0.8322	33.5793
9	-0.0090	0.2833	0.4129	0.3962	0.3962	6.5759	4.1195	0.7835	27.7449
10	-0.0100	0.2540	0.3720	0.3581	0.3581	6.1170	3.7422	0.7241	22.9842

pool = 'CSI800'

	premium	trade_pct	gross_pnl	net_pnl	net_hedge_pnl	sharpe	order_mean	order_cost	turnover
0	0.0000	0.5192	0.3398	0.2975	0.2975	7.4688	6.1435	1.0826	70.2574
1	-0.0010	0.4989	0.3470	0.3077	0.3077	7.5878	5.8552	1.0480	65.3515
2	-0.0020	0.4616	0.3440	0.3110	0.3110	7.4567	5.2518	0.9516	54.9102
3	-0.0030	0.4086	0.3283	0.3020	0.3020	6.9866	4.6409	0.8540	43.6162
4	-0.0040	0.3612	0.3096	0.2883	0.2883	6.4915	4.1688	0.7846	35.4243
5	-0.0050	0.3059	0.2829	0.2662	0.2662	5.8104	3.8450	0.7284	27.8492
6	-0.0060	0.2720	0.2639	0.2500	0.2500	5.3443	3.5021	0.6810	23.1559
7	-0.0070	0.2348	0.2406	0.2291	0.2291	4.8212	3.3077	0.6483	19.0248
8	-0.0080	0.1885	0.2143	0.2051	0.2051	4.3278	3.2814	0.6464	15.2307
9	-0.0090	0.1580	0.1944	0.1867	0.1867	3.9335	3.2286	0.6420	12.6814
10	-0.0100	0.1377	0.1736	0.1673	0.1673	3.5902	3.0164	0.6017	10.3573

pool = 'CSI300'

	premium	trade_pct	gross_pnl	net_pnl	net_hedge_pnl	sharpe	order_mean	order_cost	turnover
0	0.0000	0.3567	0.1310	0.1127	0.1127	3.8754	3.5190	0.6837	30.4820
1	-0.0010	0.3386	0.1333	0.1164	0.1164	3.9613	3.3867	0.6629	28.0566
2	-0.0020	0.2822	0.1300	0.1170	0.1170	3.8735	3.1120	0.6114	21.5653
3	-0.0030	0.2111	0.1209	0.1114	0.1114	3.5171	3.0053	0.5991	15.8048
4	-0.0040	0.1840	0.1130	0.1053	0.1053	3.2379	2.7853	0.5570	12.8100
5	-0.0050	0.1524	0.1035	0.0973	0.0973	2.8797	2.7259	0.5449	10.3787
6	-0.0060	0.1287	0.0937	0.0887	0.0887	2.5160	2.5614	0.5152	8.2858
7	-0.0070	0.1106	0.0853	0.0812	0.0812	2.2010	2.4898	0.5010	6.9273
8	-0.0080	0.0858	0.0749	0.0717	0.0717	1.7454	2.5000	0.5035	5.3984
9	-0.0090	0.0745	0.0680	0.0653	0.0653	1.4432	2.4242	0.4885	4.5485
10	-0.0100	0.0587	0.0581	0.0560	0.0560	0.9745	2.3077	0.4655	3.4149

5. Overnight Strategy

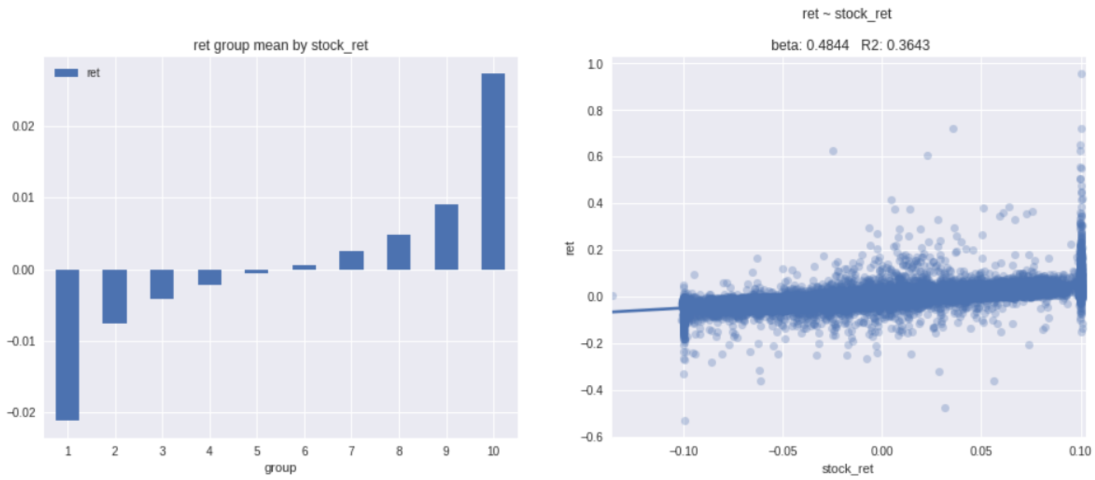
5.1 Exploratory Data Analysis

In this part we further explore the features contributing to the convertible bond future return and discuss the trading ideas.

First we have to decide 'x' and 'y'. As for a daily strategy, close to close return seems to be a default choice for y. However, we will show later in Section 6.3 that it is a much better way to build models for convertible bond intraday return and overnight return separately. Here we start from overnight return as a reasonable choice after the negative premium strategy discussed above.

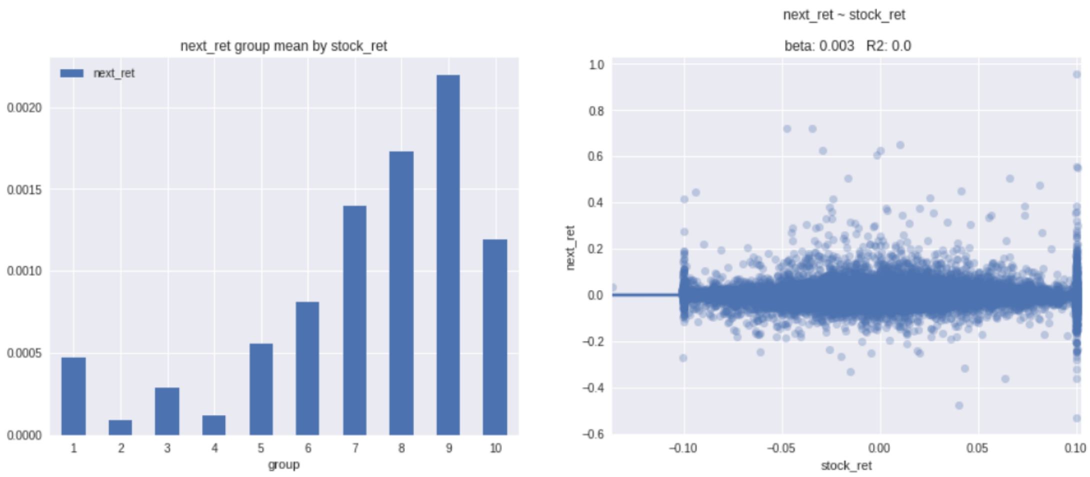
Besides, in order to design a profitable strategy, it is necessary to do some feature engineering and discover the relationship between 'x' and 'y'. Readers can find the detailed feature engineering process in Data Analysis.ipynb. This report will present the most important results with explanation.

The basic idea for data analysis is linear model. We mainly use two methods to capture the linear relationships between two features. For example, the left image below shows the mean value of 'y'(convertible bond daily return here) grouped by ascending-ordered 'x'(stock return here). It shows that convertible bond return monotonically increases with stock return, which is a reasonable phenomenon. The right image shows the beta and R-square calculated by linear regression y~x and also plots the fitted curve.

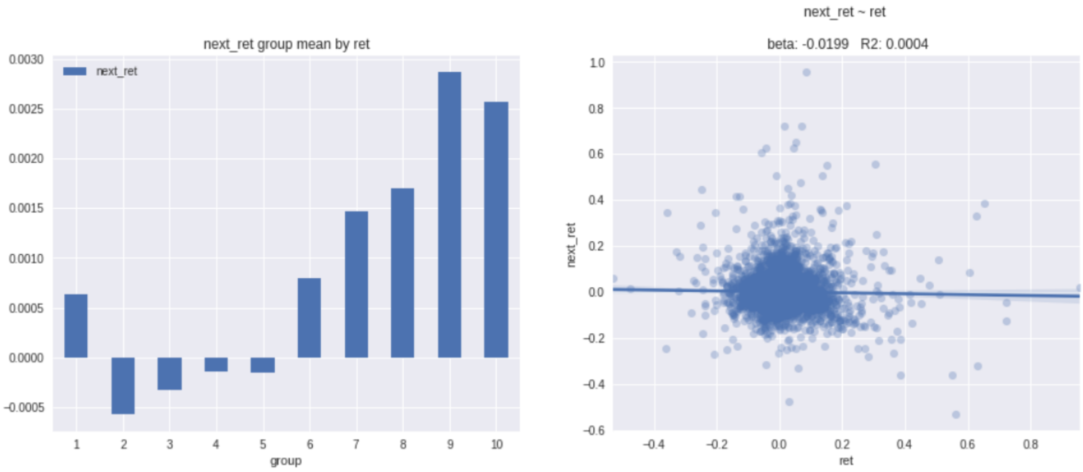


CB_return ~ stock_return

Although the convertible bond daily return has strong relationship with stock return, we can not profit from it because they are the return at the same day. What we interest is the next day convertible bond return. The following 2 images show the relationship of stock return and CB return with next CB return, respectively. The ideas behind them could be recognized as 'spillover effect between two markets' and 'auto-correlation'. However, the pattern seems unclear.

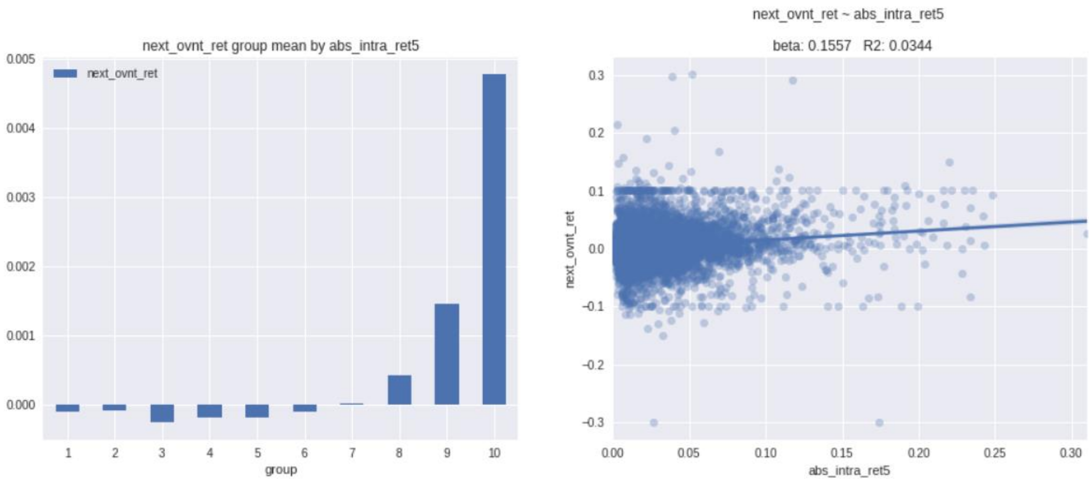


next_CB_return ~ stock_return

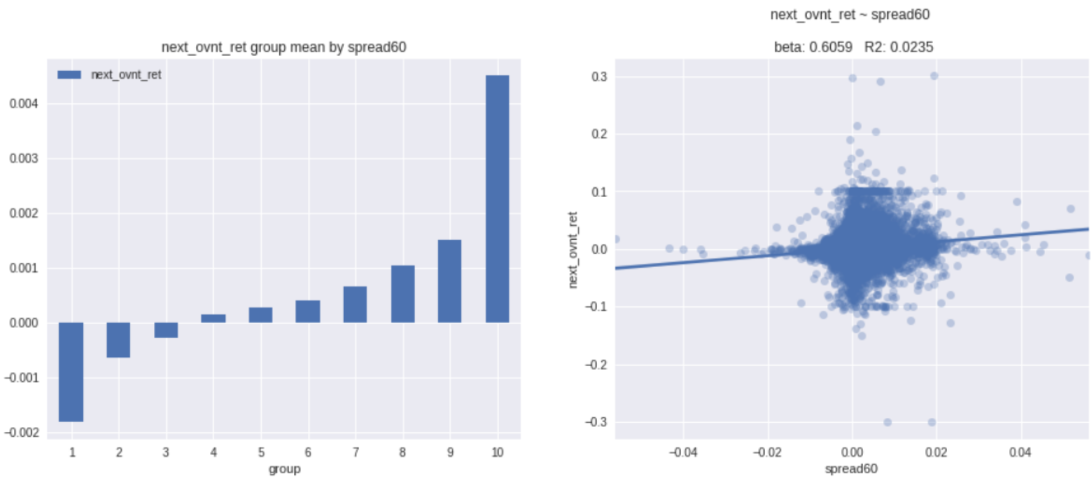


next_CB_return ~ CB_return

After some trials, we find that next overnight return (close at T - open at T+1) is predictable and here we show some interesting findings.



next CB return ~ last 5 days CB absolute intraday return mean



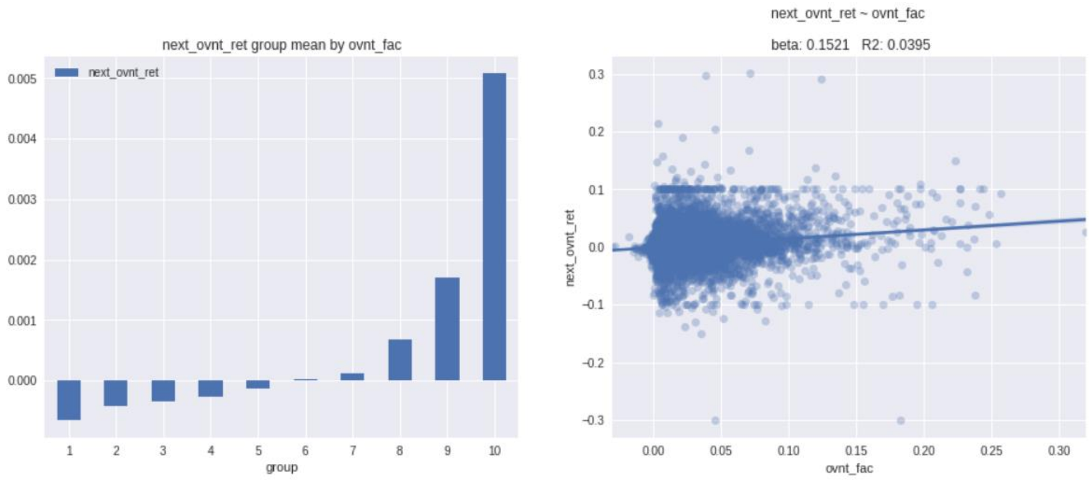
next CB return ~ last 60 days spread mean

The 'spread' above is defined as convertible bond overnight return minus beta times stock overnight return, where the beta is estimated by linear regression on dataset samples during 2017 and 2019. The estimated beta is not sensitive to the time periods.

Other features have strong linear relationship with next CB return includes: atr, abs_ret5, abs_intra_ret10, spread30, etc. Notice that all of them have a positive linear relationship with next CB overnight return, implying a strong momentum effect of CB overnight return. Moreover, the group 10 value is much larger than others, indicating an irrational and inefficient market.

5.2 Overnight Strategy (Linear Model)

We design an overnight strategy using a linear combination of the two factors mentioned above: abs_intra_ret5 and spread60.

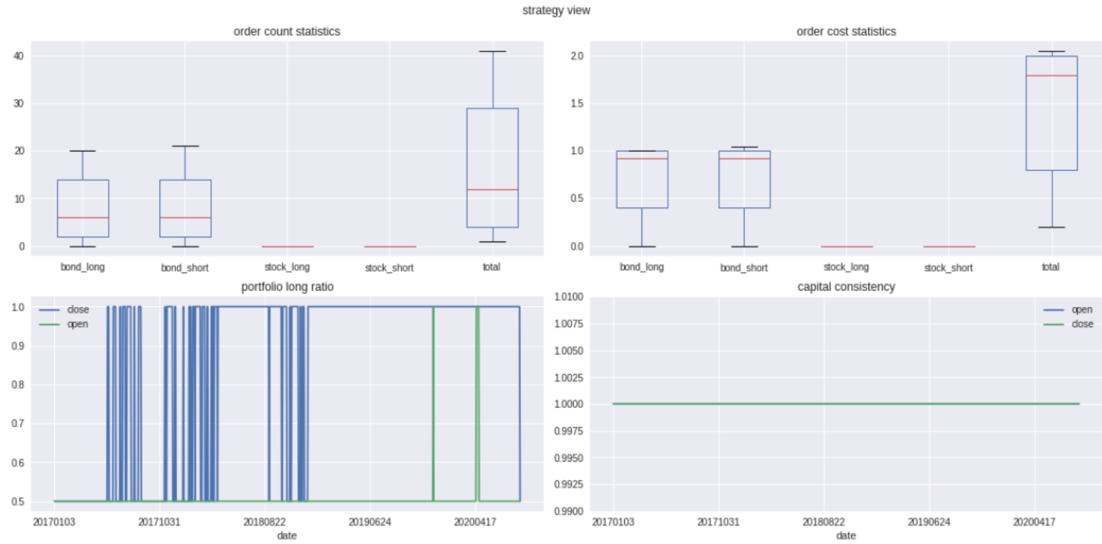


next CB return ~ overnight factor

Signal threshold is set to be 2%, which means the strategy will buy the CB when the factor value is larger than 2%. The pool is 'top1500' and the backtest performance analysis is shown below.

	total_return	annual_return	annual_volatility	sharpe_ratio	win_rate	max_drawdown
pnl	1.6632	0.4693	0.1316	3.2624	0.7754	-0.0524
pnl_open	1.6632	0.4693	0.1316	3.2624	0.7754	-0.0524
pnl_close	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
pnl_trade	1.6632	0.4693	0.1316	3.2624	0.7754	-0.0524
pnl_trans	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
pnl_intra	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
pnl_ovnt	1.6632	0.4693	0.1316	3.2624	0.7754	-0.0524
pnl_hedge	1.3257	0.3741	0.0932	3.5840	0.8002	-0.0403
net	1.5711	0.4433	0.1313	3.0715	0.7325	-0.0547
net_hedge	1.2335	0.3481	0.0929	3.3152	0.7540	-0.0419

	dates	trade_dates	order_sum	order_mean	cover_num	trade_win	conv_win	order_cost	refuse_rate	turnover	port_num	port_value	long_ratio
bond	886	0.7381	11042.0000	16.8838	174	0.5885	0	1.4096	0.0006	130.0590	3.1236	0.2602	0.6783
stock	886	0.7381	0.0000	0.0000	0	0.0000	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.5000
total	886	0.7381	11042.0000	16.8838	174	0.5885	0	1.4096	0.0006	130.0590	3.1236	0.2602	0.6783



The trading idea is not complicated while the backtest performance is good. It can be credited to the target return we choose -- next CB overnight return, instead of next close to close return, to a large extent.

We use the linear model overnight strategy as the benchmark in the further research.

5.3 Overnight Strategy (Machine Learning Model)

Next we adopt machine learning model to improve the overnight strategy. Readers can find the code in ML Model.ipynb. Based on the analysis we discussed above (extreme positive overnight return exists), we assign binomial classification task instead of prediction task for machine learning algorithms. The label 'y' is: CB next overnight return >= 0.1%, and the label ratio is 3:7 (True: False) during 2017 to 2020, 1:3 before 2020 and 1:2 in 2020. The True label ratio is around 25% ~ 35% over different years.

We choose K Nearest Neighbors, Random Forest and XGBoost to train the classification task. The training set is 2017 - 2019 and the testing set is 2020. For each machine learning model, we trained the model using several selected parameters and calculated some strategy-related metrics to measure the performance of each model. Below is the statistics of each three models. The first column is the tuned parameter of the selected models, the 2nd - 5th columns are calculated on training set and the 6th - 9th columns are calculated on testing set. Among them, the total number of signals (sig_num) and the average next overnight return of predicted 'y' (ret_mean) are the two metrics associating with the strategy performance well. Among the three machine learning models, K Nearest Neighbors records worst performance -- signal numbers is sensitive to the parameter and have small average return on testing data. Random Forest is the most stable and profitable ML algorithm -- more signals, higher return and insensitivity to the parameters. In comparison, the performance of XGBoost algorithm seems to be volatile, with more signals than Random Forest but less average return. To conclude, Random Forest is the best machine learning model for this classification task given the project dataset.

	n_neighbors	train_sig_num	train_ret_mean	train_ret_total	train_profit_rate	test_sig_num	test_ret_mean	test_ret_total	test_profit_rate
0	10	2929	0.0046	13.4855	0.7139	2246	0.0035	7.8659	0.5396
1	30	1517	0.0043	6.5476	0.6513	1357	0.0043	5.8527	0.5660
2	50	1135	0.0045	5.0673	0.6608	989	0.0041	4.0267	0.5763
3	100	664	0.0045	2.9742	0.6596	608	0.0038	2.3393	0.5592

KNeighborsClassifier(n_neighbors=n, weights='uniform')

max_depth, min_samples_split	train_sig_num	train_ret_mean	train_ret_total	train_profit_rate	test_sig_num	test_ret_mean	test_ret_total	test_profit_rate
(5, 0.002)	1120	0.0048	5.4081	0.6571	3453	0.0081	27.8984	0.6232
(5, 0.005)	1077	0.0050	5.4091	0.6546	3294	0.0079	26.1438	0.6254
(5, 0.01)	1092	0.0050	5.4761	0.6548	3226	0.0084	27.1955	0.6293
(10, 0.002)	1724	0.0069	11.8770	0.7912	4194	0.0074	30.8654	0.6273
(10, 0.005)	1569	0.0060	9.3649	0.7298	4201	0.0075	31.4784	0.6258
(10, 0.01)	1392	0.0054	7.5327	0.6868	3988	0.0078	31.1778	0.6291
(50, 0.002)	2294	0.0079	18.1102	0.8697	4608	0.0072	33.1970	0.6233
(50, 0.005)	1689	0.0060	10.2109	0.7501	4148	0.0075	31.1629	0.6275
(50, 0.01)	1427	0.0054	7.7736	0.6938	4018	0.0074	29.8176	0.6227

RandomForestClassifier(max_depth=d, min_samples_split=s, random_state=100, oob_score=True)

	n_estimators, gamma	train_sig_num	train_ret_mean	train_ret_total	train_profit_rate	test_sig_num	test_ret_mean	test_ret_total	test_profit_rate
0	(10, 10)	2105	0.0043	9.0737	0.6580	4703	0.0062	28.9818	0.6028
1	(10, 20)	1485	0.0042	6.2007	0.6424	4262	0.0069	29.3659	0.6138
2	(10, 50)	1004	0.0041	4.0890	0.6225	2956	0.0069	20.5244	0.6045
3	(20, 10)	2745	0.0045	12.3206	0.6838	5564	0.0054	30.1976	0.6001
4	(20, 20)	1971	0.0040	7.9100	0.6342	5066	0.0062	31.4399	0.6068
5	(20, 50)	1057	0.0043	4.5318	0.6187	2783	0.0068	18.9034	0.5994
6	(50, 10)	3552	0.0048	17.1623	0.7280	6408	0.0050	32.0783	0.5886
7	(50, 20)	2317	0.0041	9.4673	0.6470	5165	0.0056	29.0854	0.6033
8	(50, 50)	1057	0.0043	4.5318	0.6187	2783	0.0068	18.9034	0.5994

XGBClassifier(n_estimators=n, gamma=g, learning_rate=0.3, min_child_weight=100, max_depth=10, alpha = 1, subsample=0.8, colsample_bytree=0.8,random_state=100, use_label_encoder=False)

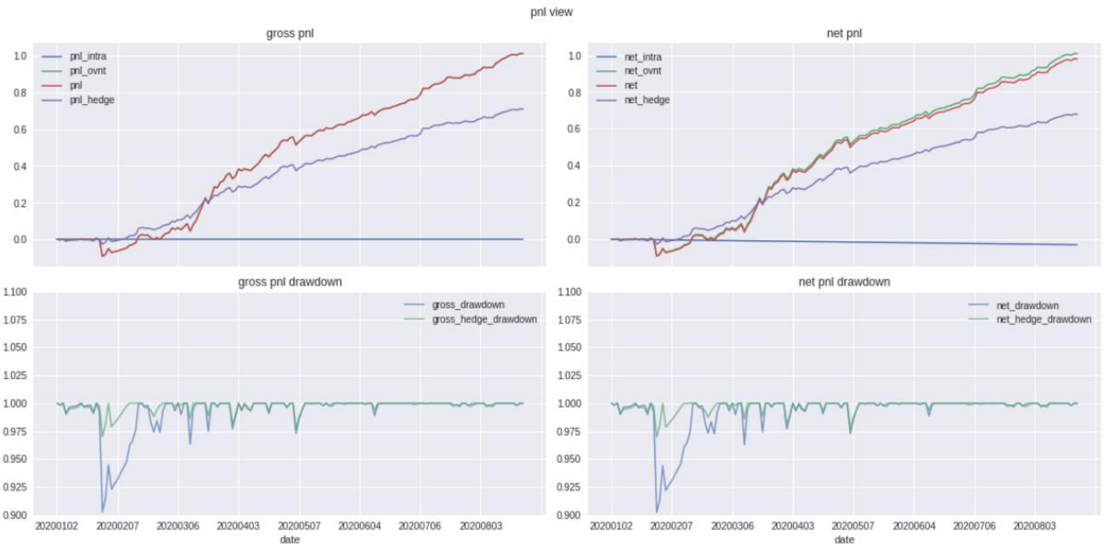
Next, we run all the models' prediction on testing set(2020) as strategy signals. The following table shows the strategy statistics of each model. You can find the model 'benchmark' in 13th place, which is the overnight strategy (linear model) we implement in Section 5.2.

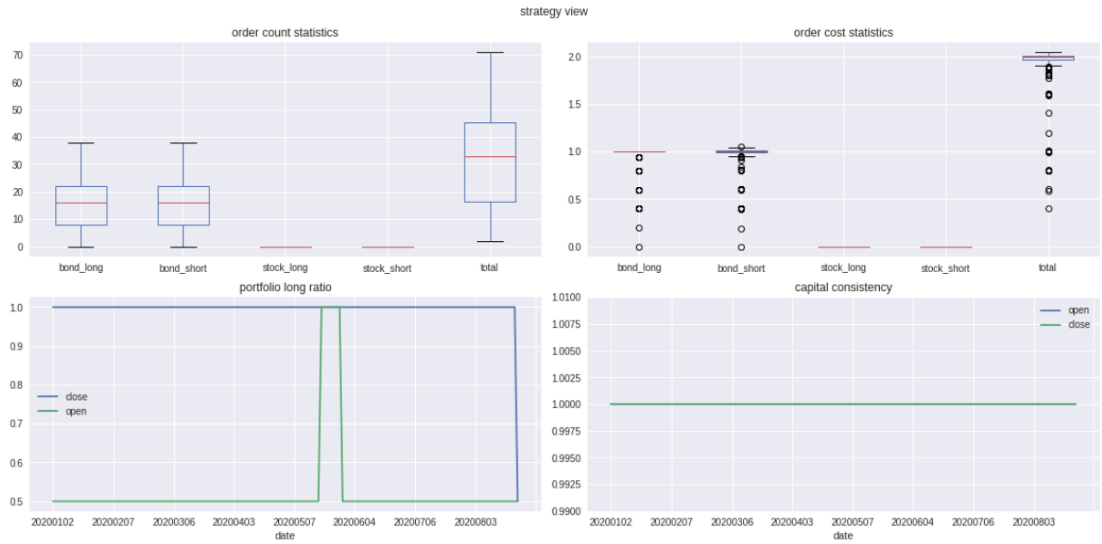
	model	trade_pct	gross_pnl	net_pnl	net_hedge_pnl	sharpe	maxdrawdown	cover_num	order_mean	order_cost	win_rate	turnover	port_num
0	rf5_0.01	1.0000	1.6292	1.5823	1.0965	7.1182	-0.0302	88	31.1484	1.8744	0.6487	234.3024	7.8323
1	rf5_0.002	1.0000	1.6014	1.5543	1.0640	7.1663	-0.0317	92	33.1484	1.8828	0.6469	235.3548	8.3323
2	rf5_0.005	1.0000	1.5916	1.5446	1.0579	6.9683	-0.0273	90	32.0516	1.8800	0.6441	234.9971	8.0581
3	rf50_0.005	1.0000	1.5469	1.4982	1.0018	7.8731	-0.0314	123	38.9935	1.9466	0.6519	243.3268	9.7935
4	rf10_0.002	1.0000	1.5272	1.4787	0.9825	7.8618	-0.0312	133	39.1613	1.9388	0.6537	242.3484	9.8355
5	rf10_0.01	1.0000	1.5123	1.4646	0.9719	7.2861	-0.0307	105	37.7032	1.9057	0.6492	238.2094	9.4710
6	xgb10_50	1.0000	1.4771	1.4309	0.9657	6.3910	-0.0287	130	27.1484	1.8482	0.6231	231.0290	6.8323
7	rf10_0.005	1.0000	1.4984	1.4500	0.9534	7.1600	-0.0317	114	39.8581	1.9368	0.6507	242.0997	10.0097
8	rf50_0.01	1.0000	1.4831	1.4352	0.9404	7.1178	-0.0309	100	38.5290	1.9159	0.6457	239.4839	9.6774
9	xgb50_50	1.0000	1.4313	1.3846	0.9114	6.0738	-0.0309	130	25.8323	1.8663	0.6179	233.2845	6.5032
10	xgb20_50	1.0000	1.4313	1.3846	0.9114	6.0738	-0.0309	130	25.8323	1.8663	0.6179	233.2845	6.5032
11	rf50_0.002	1.0000	1.4509	1.4017	0.9065	7.4063	-0.0281	144	41.8194	1.9662	0.6517	245.7781	10.5000
12	xgb10_20	0.9935	1.4015	1.3538	0.8598	6.4051	-0.0288	133	39.9221	1.9200	0.6376	238.4479	9.9613
13	benchmark	1.0000	1.3675	1.3179	0.8226	4.4290	-0.0691	125	38.4903	1.9830	0.5883	247.8715	9.6548
14	xgb20_20	1.0000	1.3144	1.2655	0.7700	6.6090	-0.0328	145	46.7226	1.9527	0.6382	244.0853	11.7258
15	xgb10_10	1.0000	1.3135	1.2643	0.7675	6.5266	-0.0207	145	43.4065	1.9685	0.6293	246.0597	10.8968
16	knn30	1.0000	1.1960	1.1529	0.7134	5.8002	-0.0231	127	14.6194	1.7234	0.6055	215.4190	3.6548
17	xgb20_10	1.0000	1.2436	1.1941	0.6960	7.0348	-0.0194	160	49.8452	1.9839	0.6363	247.9865	12.5065
18	xgb50_10	1.0000	1.2088	1.1593	0.6613	7.1376	-0.0158	176	55.3935	1.9812	0.6268	247.6509	13.8935
19	xgb50_20	1.0000	1.1786	1.1295	0.6322	6.3303	-0.0261	164	47.2129	1.9654	0.6335	245.6789	11.8484
20	knn50	1.0000	0.9393	0.9015	0.5073	4.5426	-0.0285	115	11.0710	1.5133	0.6154	189.1646	2.7677
21	knn10	1.0000	0.9908	0.9419	0.4437	4.5450	-0.0200	172	22.0903	1.9547	0.5911	244.3407	5.5226
22	knn100	0.9548	0.5542	0.5273	0.2443	1.9181	-0.0567	95	7.3649	1.1266	0.6018	134.4601	1.7581

Additionally, we present the detailed backtesting analysis of the best model in the table above, which is 'rf5_0.01'. The performance like annual return, sharpe ratio and max draw down are good.

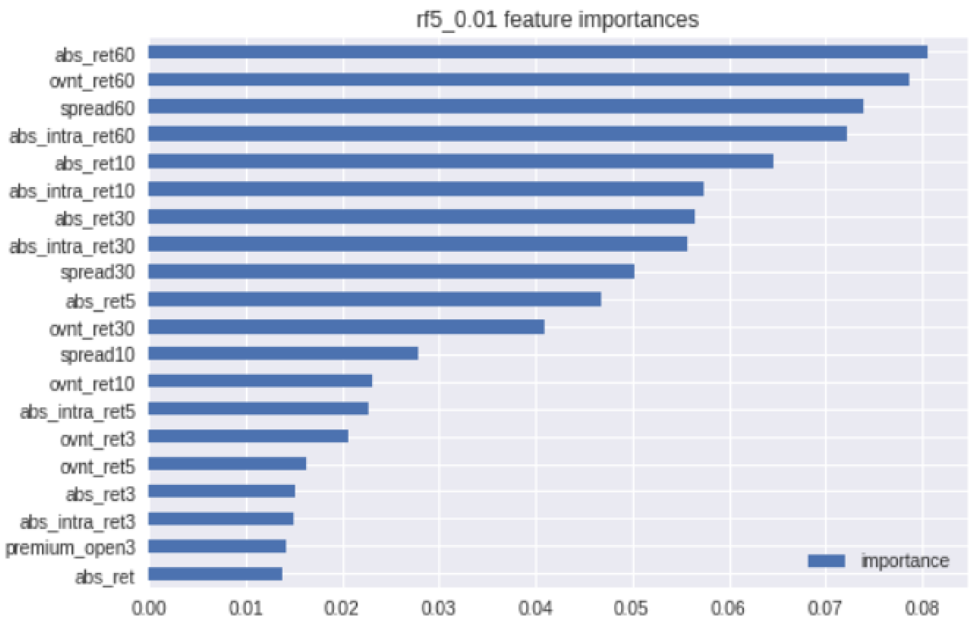
	total_return	annual_return	annual_volatility	sharpe_ratio	win_rate	max_drawdown
pnl	1.0101	1.6292	0.2547	6.2406	0.7613	-0.0973
pnl_open	1.0101	1.6292	0.2547	6.2406	0.7613	-0.0973
pnl_close	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
pnl_trade	1.0101	1.6292	0.2547	6.2406	0.7613	-0.0973
pnl_trans	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
pnl_intra	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
pnl_ovnt	1.0101	1.6292	0.2547	6.2406	0.7613	-0.0973
pnl_hedge	0.7089	1.1434	0.1485	7.4303	0.7355	-0.0298
net	0.9810	1.5823	0.2546	6.0581	0.7548	-0.0978
net_hedge	0.6799	1.0965	0.1484	7.1182	0.7161	-0.0302

	dates	trade_dates	order_sum	order_mean	cover_num	trade_win	conv_win	order_cost	refuse_rate	turnover	port_num	port_value	long_ratio
bond	155	1.0000	4828.0000	31.1484	88	0.6487	0	1.8744	0.0014	234.3024	7.8323	0.4696	0.7597
stock	155	1.0000	0.0000	0.0000	0	0.0000	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.5000
total	155	1.0000	4828.0000	31.1484	88	0.6487	0	1.8744	0.0014	234.3024	7.8323	0.4696	0.7597





The feature importance of 'rf5_0.01' model is shown below. The top contributors includes abs_ret60, ovnt_ret60, spread60 and abs_intra_ret60.



5.4 Up Limit Strategy

In this section we discuss another trading idea inspired by the inefficiency between convertible bond and stock market. The idea is, buy the convertible bond when the stock reaches to up limit at market close, and sell it tomorrow morning.

First we label the stock with 'up' -- counting the number of consecutive limit up days before. The first table below shows the sample counts and the percentage of CB next overnight return >= 1% in each 'up' label group. There are about 1% (1030 / 97K) samples reach up limit in stock market, and the extreme convertible bond overnight return are more likely to happen with the increase of 'up' label. The second table shows basic statistics of convertible bond overnight return grouped by 'up'. The distribution of overnight return changed a lot with different 'up' and the conditional mean and variance increased. Note that the more consecutive limit up before, the more the downside risk is.

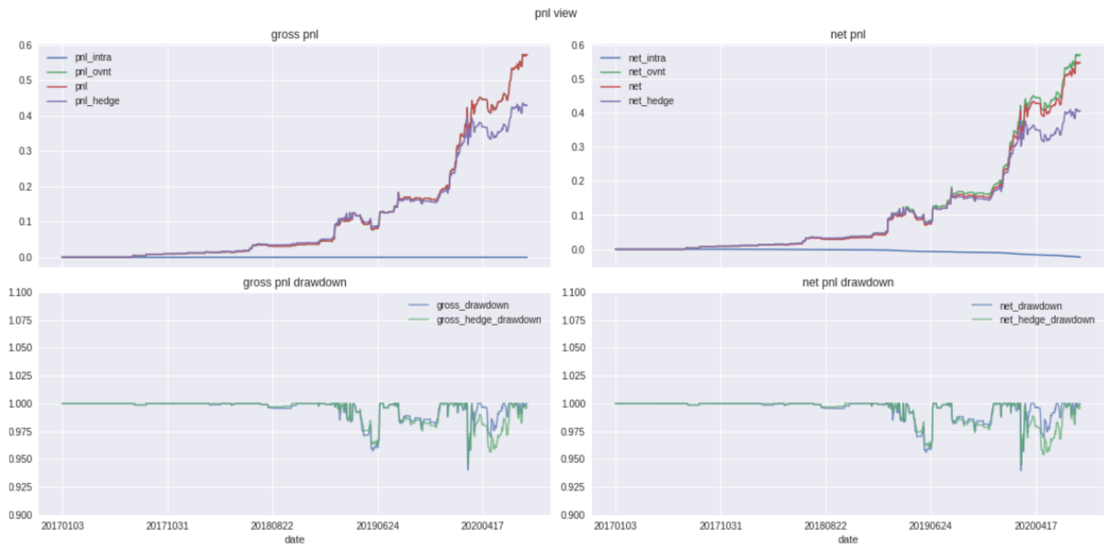
	up	sample	pct
0	0	95905	0.0586
1	1	1030	0.2728
2	2	144	0.2917
3	3	49	0.4082
4	4	33	0.5152

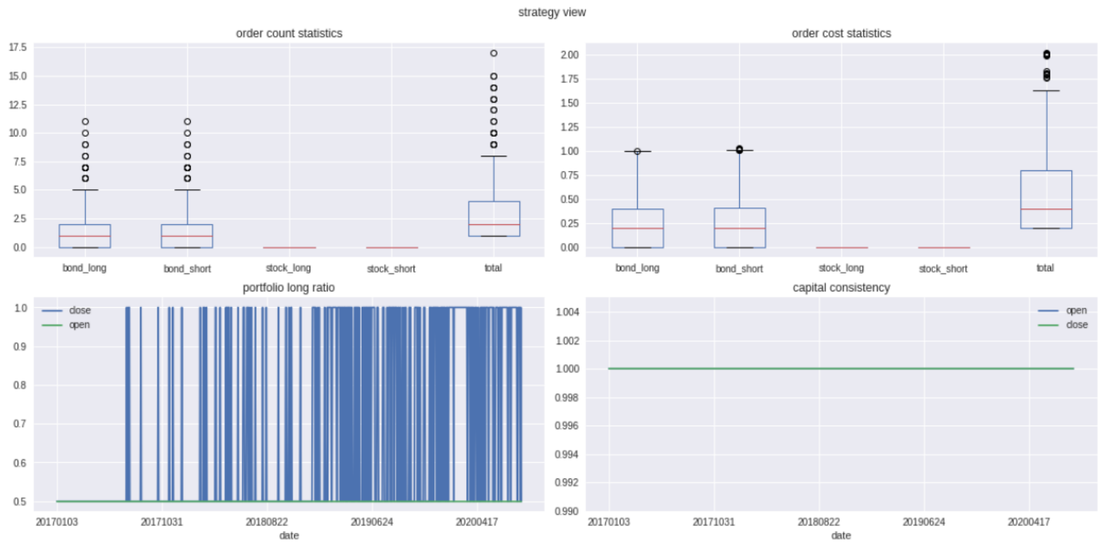
	up_0	up_1	up_2	up_3	up_4
count	95905.0000	1030.0000	144.0000	49.0000	33.0000
mean	0.0005	0.0052	0.0023	0.0040	0.0117
std	0.0095	0.0279	0.0468	0.0366	0.0720
min	-0.3000	-0.1019	-0.1277	-0.1000	-0.3000
25%	-0.0017	-0.0048	-0.0154	-0.0122	-0.0020
50%	0.0000	0.0010	-0.0012	0.0004	0.0160
75%	0.0016	0.0119	0.0138	0.0200	0.0369
max	0.2969	0.3004	0.2902	0.1000	0.1678

Next we design the strategy and test it in our backtest framework. The signal triggered when the stock reached up limit consecutively in the past n days, n is set to be 1, 2, 3 or 4. In this strategy, we adjust the weight by sqrt(5 - n), aiming to control the downside risk exposed to the trading opportunities with higher n. In the backtest below, the pool is 'top1500' and n is [1, 2, 3, 4].

	total_return	annual_return	annual_volatility	sharpe_ratio	win_rate	max_drawdown
pnl	0.5721	0.1614	0.0919	1.3208	0.8860	-0.0596
pnl_open	0.5721	0.1614	0.0919	1.3208	0.8860	-0.0596
pnl_close	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
pnl_trade	0.5721	0.1614	0.0919	1.3208	0.8860	-0.0596
pnl_trans	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
pnl_intra	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
pnl_ovnt	0.5721	0.1614	0.0919	1.3208	0.8860	-0.0596
pnl_hedge	0.4295	0.1212	0.0814	0.9981	0.8837	-0.0550
net	0.5492	0.1550	0.0918	1.2519	0.7494	-0.0603
net_hedge	0.4066	0.1147	0.0813	0.9196	0.7562	-0.0557

	dates	trade_dates	order_sum	order_mean	cover_num	trade_win	conv_win	order_cost	refuse_rate	turnover	port_num	port_value	long_ratio
bond	886	0.4391	1226.0000	3.1517	169	0.5808	0	0.5885	0.0000	32.3003	0.3459	0.0644	0.5784
stock	886	0.4391	0.0000	0.0000	0	0.0000	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.5000
total	886	0.4391	1226.0000	3.1517	169	0.5808	0	0.5885	0.0000	32.3003	0.3459	0.0644	0.5784





Notice that this strategy is unstable over time because of the nature of signals —— 1) total number is relatively small, and becomes smaller (about 50% of total signals) when the pool is limited in 'top1500' 2) more likely to happen when the market is over traded / extremely irrational. In conclusion, this trading idea works but highly depends on the market condition. Possible improvement can be made by detecting the market sentiment and design customized trading strategy for it.

6. Convertible Bond Pricing Model

As we know, convertible bond is a hybrid asset including one fix-income security and one option on the stock. Hence, it is reasonable to expect a theoretical fair price of convertible bond. This part we spotlight the implementation of convertible bond pricing model and try to find alphas from the gap between fair price and market price. First we refer to *Convertible Bond Pricing: A Monte Carlo Approach* and implement three convertible bond models step by step in Section 6.1. Readers can find the implementation code in PricingModel.ipynb. Section 6.2 mainly focus on the parameters formatting and model price calculation. The code is in Run Model.ipynb. Last, we design the trading idea based on the modeling price and further discuss some insights behind.

6.1 Model Implementations

There are 3 typical models to price the convertible bond, namely, Black-Scholes stochastic model, Tree methods, and Monte Carlo methods. We will briefly introduce each of them below. Readers can refer to the book mentioned above for more technical details.

- **BS model**: Black-Scholes model is broadly used in option pricing. It provides closed-form formula for European convertible bonds (investors can only convert the bond to stock at maturity) without any items like callability or putability. The formula is as follows:

$$CB^* = e^{-rT}B(T) + n(T)\underbrace{e^{-rT}\mathbb{E}^{\mathbb{Q}}(\max(S(T) - \frac{B(T)}{n(T)}, 0))}_{(ii)}$$

$$\Rightarrow CB^* = e^{-rT}B(T) + n(T)S_0e^{-qT}\Phi(d_1) - B(T)e^{-rT}\Phi(d_2)$$

$$\begin{aligned} d_1 &= \frac{1}{\sigma\sqrt{T}}\left(log(\frac{N(T)S_0}{B(T)}) + ((r-q) + \frac{\sigma^2}{2})T\right) \\ d_2 &= d_1 - \sigma\sqrt{T} \\ \Phi(x) &= \int_{-\infty}^{x} e^{-\frac{1}{2}z^2}dz \end{aligned}$$

$$CB^* = e^{-rT}B(T) + n(T)\underbrace{e^{-rT}\mathbb{E}^{\mathbb{Q}}(\max(S(T) - \frac{B(T)}{n(T)}, 0))}_{(ii)}$$

$$\Rightarrow CB^* = e^{-rT}B(T) + n(T)S_0e^{-qT}\Phi(d_1) - B(T)e^{-rT}\Phi(d_2)$$

$$\begin{aligned} d_1 &= \frac{1}{\sigma\sqrt{T}}\left(log(\frac{N(T)S_0}{B(T)}) + ((r-q) + \frac{\sigma^2}{2})T\right) \\ d_2 &= d_1 - \sigma\sqrt{T} \\ \Phi(x) &= \int_{-\infty}^{x} e^{-\frac{1}{2}z^2}dz \end{aligned}$$

- **Tree Method**: The basic idea of tree method is to 'grow a tree' to simulate the possible paths of underlying asset price movement, and estimate the probability of each path. Then walk through the tree backward to calculate the discounted value till the start point. Tree method works for American convertible bonds with items except for soft call, soft put and contigent conversion (computational expensive if do so).

- **Monte Carlo Method:** Monte Carlo simulation is a generalized method used for asset pricing. The basic idea is to simulate many trials (observations) of the asset price movement. The difference between Monte Carlo and tree model is that each trial in Monte Carlo can be regarded as a possible time series observation/sample while there exists many paths to reach one point in tree method. Therefore, Monte Carlo has advantages dealing with complicated path-dependent items like soft call, soft put and contigent conversion. Monte Carlo methods also need forward and backward calculation. We implemented two discretization methods for forward calculation called Euler schemes and Milstein schemes, and two methods for backward calculation called Least Square Monte Carlo and Hedged Monte Carlo. The key formulas are shown below.

- Forward deduction: Euler Schemes

$$X_{i+1} = X_i + b(t_i, X_i)(t_{i+1} - t_i) + \sigma(t_i, X_i)(W_{i+1} - W_i)$$

- Forward deduction: Milstein Schemes

$$\begin{aligned} X_{i+1} = X_i + b(t_i, X_i)(t_{i+1} - t_i) + \sigma(t_i, X_i)(W_{i+1} - W_i) + \\ + \frac{1}{2}\sigma(t_i, X_i)\sigma_x(t_i, X_i)((W_{i+1} - W_i)^2 - (t_{i+1} - t_i)) \end{aligned}$$

- Backward deduction: Least Square Monte Carlo

$$\alpha = \underset{\alpha \in \mathbb{R}^n}{\text{argmin}}\ \mathbb{E}((C(t + \Delta t, S_{t+\Delta t})e^{-r\Delta t} - \alpha \cdot \chi(S_t))^2)$$

- Backward deduction: Hedged Monte Carlo

$$\alpha = \underset{\alpha \in \mathbb{R}^n}{\text{argmin}}\ \mathbb{E}((C(t + \Delta t, S_{t+\Delta t})e^{-r\Delta t} - \alpha \cdot \chi(S_t) - \alpha \cdot \chi'(S_t)(S_{t+\Delta t}e^{-r\Delta t} - S_t))^2)$$

- value of convertible price

$$V(t) = \begin{cases} max(H, S \cdot C_r) & \text{if } t \notin \Omega_{Call} \text{ and } t \notin \Omega_{Put} \\ max(min(H, K_c^{USD} + c_{USD}), S \cdot C_r) & \text{if } t \in \Omega_{Call} \text{ and } t \notin \Omega_{Put} \\ max(H, K_p^{USD} + c_{USD}, S \cdot C_r) & \text{if } t \notin \Omega_{Call} \text{ and } t \in \Omega_{Put} \\ max(min(H, K_c^{USD} + c_{USD}), K_p^{USD} + c_{USD}, S \cdot C_r) & \text{if } t \in \Omega_{Call} \text{ and } t \in \Omega_{Put} \end{cases}$$

The implementation code of the three models share the same parameter formatting as follows. The modelling prices are the same (or have minor difference) with the examples in the reference book. In further practice, we use Monte Carlo methods because most of the convertible bonds traded in the market have items like soft call and soft put.

```python
para = {
    # bond
    'maturity': 2,
    'face_value': 100,
    'coupon': dict(zip([0.5, 1, 1.5, 2], [0.05, 0.05, 0.05, 0.05])),
    # rdmpt
    'redemption': 1,
    # conv
    'conv_ratio': 1,
    'conv_price': 100,
    'conv_date':(0, 2),
    'conv_adj':None,
    # call
    'call':110,
    'call_date':(0, 2),
    'call_soft':(20, 20, 1.1),
    # put
    'put': 98,
    'put_date':(0, 2),
    'put_soft':None,
    # stock
    'r': 0.05,
    'S': 100,
    'sigma': 0.4,
    'q': 0.1,
    # credit spread
    'cr':0.05,
}
```

6.2 Price Calculation

Before the price calculation, we translate the convertible bond items text into the formatting parameters as above. Besides, the volatility of underlying assets is estimated with different methods and thus generated 3 versions of modelling prices.

Version 1 is to use rolling 100 days historical volatility. It is the easier way while may encounter some problems that a gap between modelling price and market price of some convertible bonds is not negligible and the gap level is also not stationary over time. In that case, the bias of modelling price bring difficulties to design profitable factors.

Version 2 is to estimate the hidden volatility at the beginning and then use the ratio (historical vol at t0 / hidden vol at t0) to adjust the rolling historical volatility. This method mitigate the gap at the beginning but can not capture the changes of the ratio in the future.

Version 3 is to dynamically estimate the hidden volatility and adjust the rolling historical volatility when the bias between modelling price and market price is larger than the threshold. However, it is not always the case that we can decrease the gap by only adjusting the hidden volatility because the convertible bond items acts like a 'barrier option' and the modelling price itself has a boundary naturally. Therefore, we label those samples that failed to adjust for more than 2 days with gap over 30% (around 15% in total samples) and drop these labeled samples in the later factor design to minimize the effect of model failure.

The sample data after the price calculation is as follows, where column 'model' is the modelling price and the column 'overvalue' is calculated as the difference between market price and modelling price in percentage. Higher overvalue implies a higher mean reversion of the market price in the future.

	date	bond_ticker	stock_ticker	vol	close	model	overvalue
0	20170103	110030.SH	600185.SH	0.5442	114.0400	114.7751	-0.0064
1	20170104	110030.SH	600185.SH	0.5393	114.2300	110.1316	0.0372
2	20170105	110030.SH	600185.SH	0.5258	114.9500	113.9838	0.0085
3	20170106	110030.SH	600185.SH	0.5208	114.8600	114.1824	0.0059
4	20170109	110030.SH	600185.SH	0.5880	115.5300	112.4799	0.0271

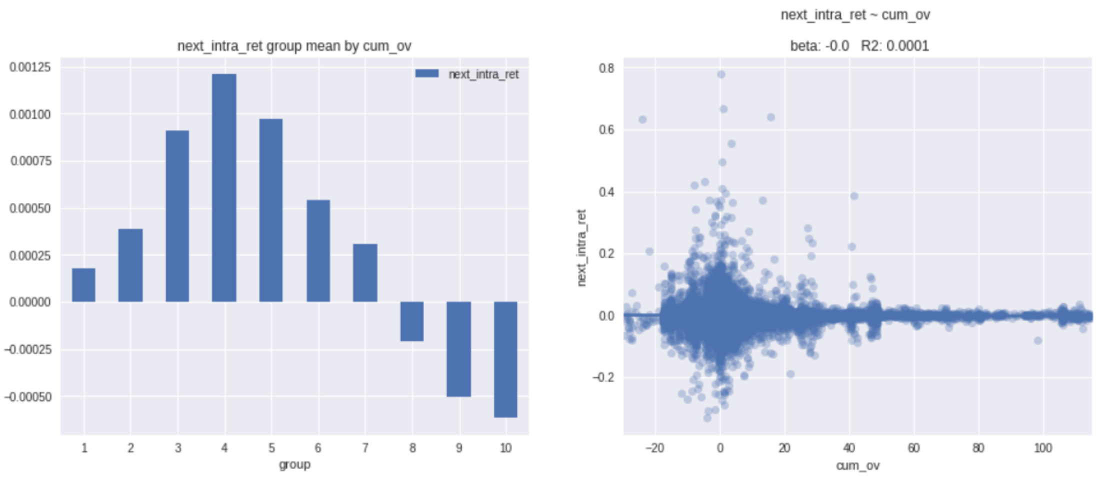
6.3 Cumulative Overvalue Strategy (Close to Close)

So far, we have finished all the data preparations. We start from the nature of modelling price —— it is reasonable to expect that the market price should move around the modelling price. In that case, mean reversion effect is the trading opportunities what we want to capture by factor design.

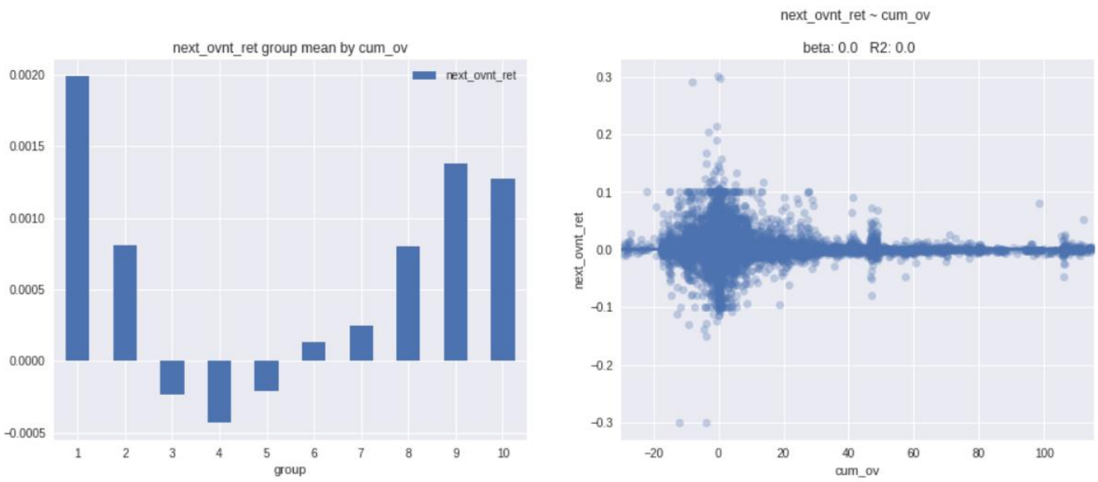
Here we display some interesting findings given by the linear analysis tool. The first image below is the linear relationship between the convertible bond next return (close to close) and its cumulative overvalue. The first group, which is the group with lowest cumulative overvalue, has the largest next return and significantly differs from others.

CB next return ~ cumulative overvalue

However, it is not the whole picture. The pattern of convertible bond overnight return and intraday return are exactly opposite from each other and have a non-linear relationship with cumulative overvalue. Also, it is worth to mention that the results are pretty similar when we replace the cumulative overvalue by conversion premium, and the correlation of them is more than 0.3.



CB next intraday return ~ cumulative overvalue



CB next overnight return ~ cumulative overvalue

Next, we test the trading strategy based on cumulative overvalue. The key strategy parameters include the entry/exit threshold of cumulative overvalue and the maximum holding period of the asset. The convertible bond will be sold either when the exit threshold is triggered or reaches the maximum holding period.

	total_return	annual_return	annual_volatility	sharpe_ratio	win_rate	max_drawdown
pnl	1.3597	0.3837	0.2385	1.4411	0.6896	-0.1805
pnl_open	1.0735	0.3029	0.1060	2.4791	0.7619	-0.0721
pnl_close	0.2862	0.0808	0.2057	0.1981	0.6783	-0.1940
pnl_trade	1.3597	0.3837	0.2452	1.4016	0.7054	-0.1762
pnl_trans	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
pnl_intra	0.2862	0.0808	0.2057	0.1981	0.6783	-0.1940
pnl_ovnt	1.0735	0.3029	0.1060	2.4791	0.7619	-0.0721
pnl_hedge	0.8693	0.2453	0.1641	1.2508	0.6986	-0.1410
net	1.2529	0.3535	0.2384	1.3154	0.6862	-0.1912
net_hedge	0.7625	0.2151	0.1640	1.0678	0.6930	-0.1630

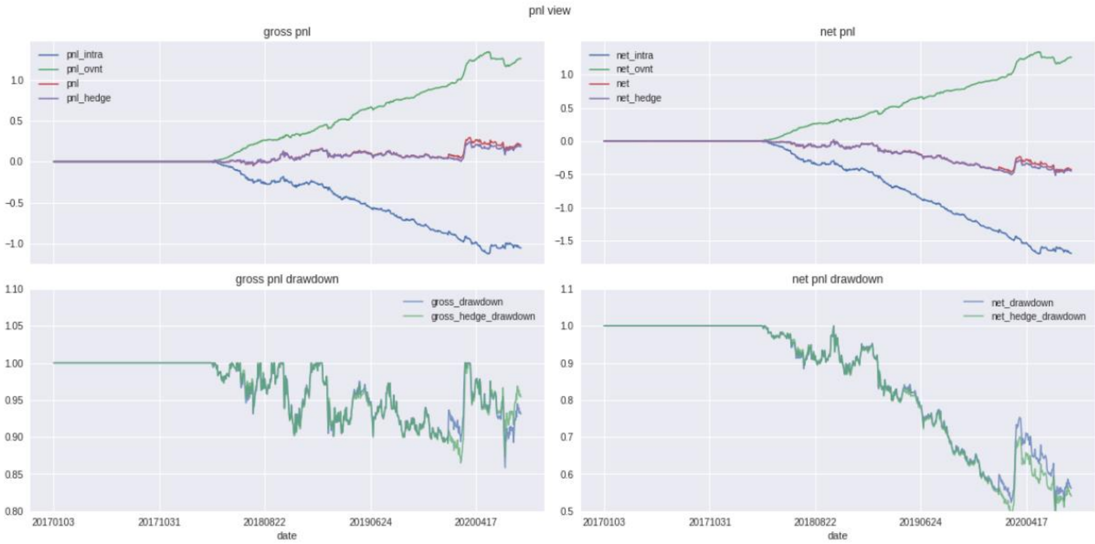
	dates	trade_dates	order_sum	order_mean	cover_num	trade_win	conv_win	order_cost	refuse_rate	turnover	port_num	port_value	long_ratio
bond	886	0.6637	11610.0000	19.7449	49	0.5192	0	1.8160	0.0007	150.6496	6.5468	0.6026	0.8310
stock	886	0.6637	0.0000	0.0000	0	0.0000	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.5000
total	886	0.6637	11610.0000	19.7449	49	0.5192	0	1.8160	0.0007	150.6496	6.5468	0.6026	0.8310

The performance is not that good. But the result provides us deeper understanding about the relationship between convertible bond overnight return and intraday return. Notice that in the image above, the blue line represents the intraday P&L(worse) while the green one represents the overnight P&L(better). The highlighted statistics also tells that cumulative return preforms much better during overnight.

Further, we change the hedging tool from 'top1500' to stocks (matched with convertible bond). The results differ a lot from above. The hedged overnight return outperforms the hedged intraday return for mainly two reasons: 1) cumulative overvalue factor itself performs better during overnight, 2) the average intraday return is much higher than the average overnight return in stock market, which is opposite with convertible bond market.

	total_return	annual_return	annual_volatility	sharpe_ratio	win_rate	max_drawdown
pnl	0.2069	0.0584	0.1561	0.1177	0.6682	-0.1414
pnl_open	1.2622	0.3561	0.0843	3.7498	0.8104	-0.0771
pnl_close	-1.0553	-0.2978	0.1331	-2.5383	0.5858	-1.1268
pnl_trade	0.2069	0.0584	0.1776	0.1035	0.6885	-0.1298
pnl_trans	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
pnl_intra	-1.0553	-0.2978	0.1331	-2.5383	0.5858	-1.1268
pnl_ovnt	1.2622	0.3561	0.0843	3.7498	0.8104	-0.0771
pnl_hedge	0.1851	0.0522	0.1533	0.0798	0.6625	-0.1351
net	-0.4274	-0.1206	0.1563	-1.0277	0.6230	-0.4978
net_hedge	-0.4492	-0.1267	0.1535	-1.0865	0.6208	-0.5244

	dates	trade_dates	order_sum	order_mean	cover_num	trade_win	conv_win	order_cost	refuse_rate	turnover	port_num	port_value	long_ratio
bond	886	0.6637	11336.0000	19.2789	49	0.5221	0	1.8064	0.0007	149.8526	6.3928	0.5994	0.8310
stock	886	0.6637	11182.0000	19.0170	48	0.4919	0	1.7762	0.0198	147.3452	6.4656	0.6076	0.1690
total	886	0.6637	22518.0000	38.2959	97	0.5071	0	3.5825	0.0103	297.1977	12.8584	1.2070	0.4988



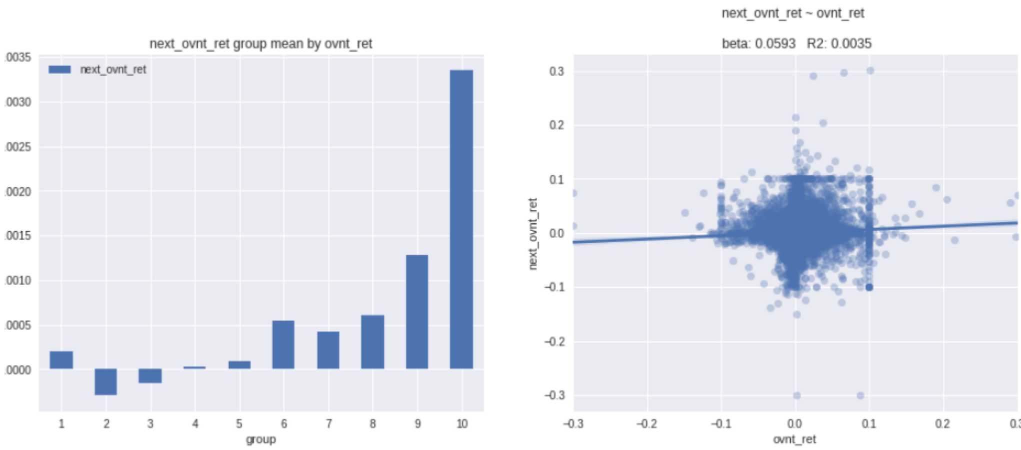
The insights behind the analysis above are:

- The relationship between convertible bond overnight return and intraday return is like a 'jigsaw puzzle' —— looks ordinary as a whole but differs a lot from each other.
- It's a good idea to hedge the convertible bond overnight return by stock, but **be careful when it comes to intraday return**.
- Really hard for one strategy performing well on both the convertible bond overnight return and intraday return. Building models for each of them is highly recommended.

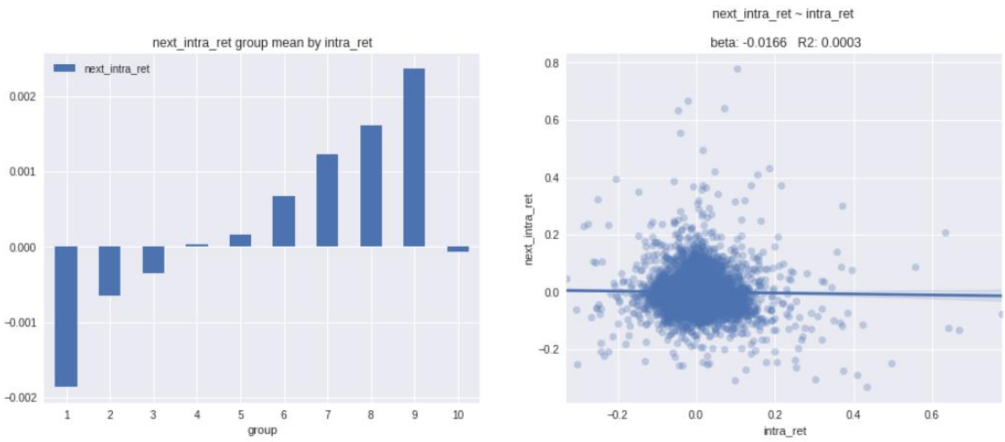
6.4 Intraday Strategy

We have designed an overnight strategy for convertible bond in Section 5.2 and 5.3. In this part we turn to the interday return and build a similar model for it.

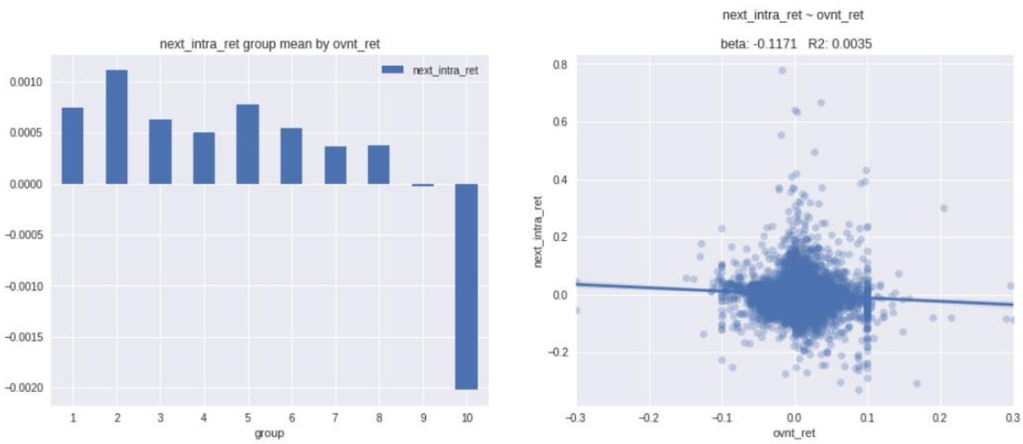
The analysis in Section 6.3 discussed many differences between overnight return and intraday return. Here we present another finding. The first image below shows a strong momentum effect of overnight return, especially for the largest groups. The second image shows a similar pattern of intraday return except for the 10th group —— reverse effect happens in the largest positive intraday return group. The third image tells the same story combining the first and second images, that overnight return has strong reverse effect on next intraday return.



next overnight return ~ overnight return

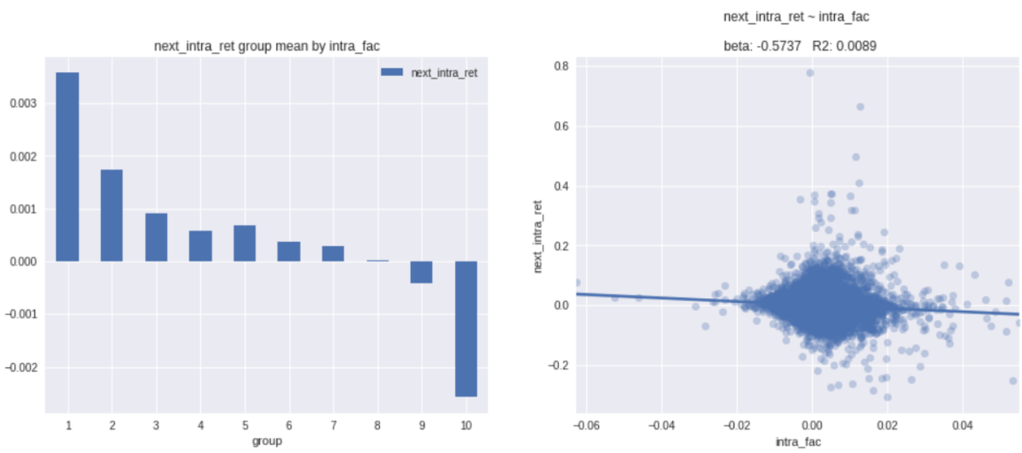


next intraday return ~ intraday return



next intraday return ~ overnight return

After further trials, we find an effective factor to perfectly capture the reverse effect. The image below shows next intraday return is almost monotonically decreasing with the intraday factor, which is calculated as a linear combination of rolling 10 days overnight return and absolute cumulative overvalue. Worth to mention that the result is enhanced by dropping the samples labelled invalid (failed to fit the hidden volatility based on the market price, which implying an unreasonable price range) in Section 6.2.

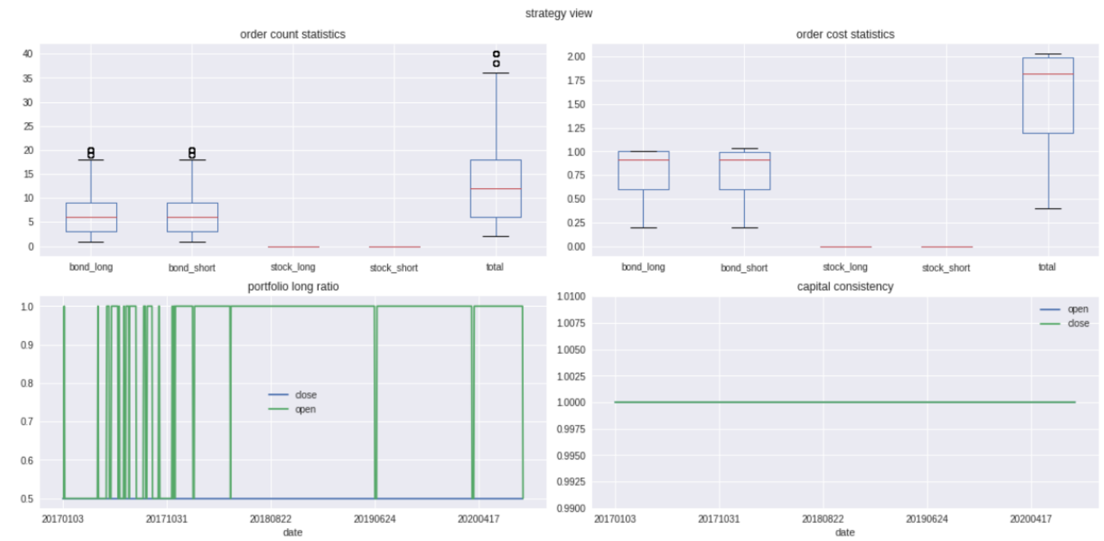


next intraday return ~ intraday factor

Last, we show the intraday strategy bactesting results. The threshold parameter is -0.2%, which means the signal triggers when the rolling 10 overnight return of convertible bond is lower than -0.2%. The samples with invalid label are dropped and the pool is 'top1500'. The performance of this linear intraday strategy nearly reaches the annual return of previous linear overnight model, with better sharpe ratio, lower volatility and smaller max drawdown.

	total_return	annual_return	annual_volatility	sharpe_ratio	win_rate	max_drawdown
pnl	1.5340	0.4328	0.1151	3.4121	0.6964	-0.0452
pnl_open	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
pnl_close	1.5340	0.4328	0.1151	3.4121	0.6964	-0.0452
pnl_trade	1.5340	0.4328	0.1151	3.4121	0.6964	-0.0452
pnl_trans	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
pnl_intra	1.5340	0.4328	0.1151	3.4121	0.6964	-0.0452
pnl_ovnt	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
pnl_hedge	1.1567	0.3264	0.0681	4.2062	0.7009	-0.0124
net	1.4229	0.4015	0.1149	3.1457	0.6907	-0.0472
net_hedge	1.0456	0.2950	0.0679	3.7577	0.6817	-0.0136

	dates	trade_dates	order_sum	order_mean	cover_num	trade_win	conv_win	order_cost	refuse_rate	turnover	port_num	port_value	long_ratio
bond	886	0.8093	9630	13.4310	193	0.5913	0	1.5487	0.0017	156.6578	2.7173	0.3129	0.7023
stock	886	0.8093	0	0.0000	0	0.0000	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.5000
total	886	0.8093	9630	13.4310	193	0.5913	0	1.5487	0.0017	156.6578	2.7173	0.3129	0.7023



7. Conclusion and Future Work

7.1 Report Conclusion

Up to now, this report introduced the main work of the convertible bond project. From data preparation to trading ideas generation, we explored many characteristics of convertible bond, including its items, market statistics, conversion premium, pricing model, and most importantly, the ingredients of daily return —— intraday return plus overnight return. All the data analysis above helped us have a deeper understanding of convertible bond and successfully design two main profitable trading strategies. Hope the work of this project will provide readers some insights about convertible bond market in China.

7.2 Future Work

Convertible bond market in China is developing fast in recent years and there are many trading opportunities for us to explore. The work of this project is only a little taste of it. Here the reporter gives several suggestions for further work:

- **Feature engineering**: In my research work I tried several classic technical indicators but most of them not work, or can not beat existing factors. Researchers may design features to capture the asset volatility and liquidity, etc and hopefully will improve the strategy performance.

- **Modelling Task**: This project implemented classification models to enhance the overnight strategy. Researchers may consider prediction tasks and build models on overnight return and intraday return using time series models, deep learning or reinforcement learning.

- **Data Diversification**: Our dataset only includes the convertible bond items and daily market data, and thus our trading time is limited either at market open or close. Researchers can use market microstructure data like limit orderbook and trade data to design high-frequency strategies. Sentiment data can also be used to capture the heating topics in market to enhance the strategy return(especially the overnight return). Moreover, stock fundamental data and credit data will be helpful to build long-term strategy and control the default risk.

Thanks to my mentors, Xiaoyang Zhao and Ke Gao. Their clear instructions and constructive thoughts help me finish this project.