straddle_research_class

index
/home/jirong/Desktop/github/straddle/straddle_research_class.py

```
Created on Thu Dec 10 10:38:04 2020

@author: jirong
```

Modules

sklearn.datasets	os	random	yfinance
datetime	pandas	re	
sklearn.model_selection	matplotlib.pyplot	time	
numpy	quandl	util	

Classes

builtins.object

StraddleResearch

class **StraddleResearch**(builtins.object)

Methods defined here:

__init__(self, path, ticker, date_start, date_end, shift_days, buy_day, expiry_day, fix_capital, cap, otm_put_perdiff=None, include_hedge=1)

```
Constructor for FuturesResearch class

:param path: path to data folder file (e.g. "./trend_following/quantopian_data/futures_incl_2016.csv")
:param ticker: ticker of options chain
:param date_start: Starting date of strategy
:param date_end: Ending date of strategy
:param shift_days: Shifting day from purchase date
:param buy_day: Buy on which day (0: Mon, 6: Sun)
:param expiry_day: Expire on which day
:param fix_capital: Fix notional capital of underlying asset (e.g. 600000)
:param cap: Parameter used to cap forecast strength (see Jupyter notebook)
:param otm_perdiff: Out of the money put. % away from ATM strike.
:param include_hedge: Include OTM put. (1: to include. 0: Not to include)
:return: returns StraddleResearch class
```

block_boostrap_simulations(self, num_simulations=10, test_size=0.6666666666666666, param_space=[0.05, 0.1, 0.15, 0.2, 0.25])

```
Block bootstrapping simulation (not used in study)
```

boostrap_simulations(self, num_samples=100, num_simulations=10, param_space=[0.05, 0.1, 0.15, 0.2, 0.25])

```
Bootstrapping simulation (used in study)
:param num_sample: Number of datapoints per bootstrap sample
:param num_simulations: Number of simulations carried out per parameter space
:param param_space: Parameter space for forecast_cap
:return: returns bootstrap statistics dataframe
```

compute_price_change(self)

```
Options price change
```

execute_flow(self)

```
Execute data analysis flow
```

get_stock_data(self)

obtain_vix_filter(self)

```
Include risk filter
```

populate_strategy_data_frame(self)

```
Populate strategy data-frame with options chain data
```

profits_generation(self)

```
Generating profits
```

read_options_chain_data(self, strike, date_quote, date_liquidate, date_expire)

```
Read options chain data from CBOE
:param strike: Strike price
:param date_quote: Quotation/Purchase date
:param date_liquidate: Liquidation date
:param date_expire: Expiry date
:return: returns options df
```

read_options_chain_otm_data(self, strike, date_quote, date_liquidate, date_expire)

```
Read options chain data from CBOE
:param strike: Strike price
```

```
:param date_quote: Quotation/Purchase date
:param date_liquidate: Liquidation date
:param date_expire: Expiry date
:return: returns options info
```

subset_straddle_data(self, buy_day, expiry_day)

```
Subset stock data according to purchase date and liquidation date
:param buy_day: Buy on which day (0: Mon, 6: Sun)
:param expiry_day: Expire on which day
```

trade_statistic_simulation(self, profits_df, forecast_cap)

```
Profits dataframe for each bootstrap sample.

:param profits_df: Profits data-frame
:param forecast_cap: forecast_cap parameter for forecast strength (refer to Jupyter notebook)
:return: returns trade statistic
```

Data descriptors defined here:

__dict__

```
dictionary for instance variables (if defined)
```

__weakref__

```
list of weak references to the object (if defined)
```