

---

# **quantlib Documentation**

***Release 1.0.1***

**SnowWalkerJ**

**May 20, 2017**



<b>1</b>	<b>quant.analysis</b>	<b>1</b>
1.1	quant.analysis.ic . . . . .	1
<b>2</b>	<b>quant.backtest</b>	<b>3</b>
2.1	Get Started . . . . .	3
<b>3</b>	<b>quant.data</b>	<b>5</b>
3.1	quant.data.wind . . . . .	5
<b>4</b>	<b>quant.transform</b>	<b>9</b>
4.1	quant.transform.distribution . . . . .	9
<b>5</b>	<b>quant.utils</b>	<b>11</b>
5.1	quant.utils . . . . .	11
5.2	quant.utils.calendar . . . . .	12
<b>6</b>	<b>Indices and tables</b>	<b>13</b>
	<b>Python Module Index</b>	<b>15</b>



### quant.analysis.ic

IC correlation score

`quant.analysis.ic.get_ic(table1, table2)`  
IC score

**Parameters** `table1, table2`: `pd.DataFrame`

IC

**Returns** float

IC



### Get Started

:

```
from quant.backtest.stock.strategy import AbstractStrategy

class SimpleStrategy(AbstractStrategy):
    start_date = "2017-03-01"
    def handle(self, today, universe):
        self.change_position({universe[0]: 0.1})
```

2017310% 'SimpleStrategy().run()'





## quant.data.wind

### Functions

`quant.data.wind.get_wind_data` (*table*, *field*, *index=None*, *columns=None*, *parse\_dates=True*)  
Wind

**Parameters** table

SQL, 'quant.data.wind.tables'

**field**

**index**

index

**columns**

columns

**parse\_dates: bool, optional**

indexdatetime

**Returns** pd.DataFrame

### Tables

#### AShareEODPrices

`class quant.data.wind.tables.AShareEODPrices` (*\*\*kwargs*)  
A

### Attributes

object_id	
s_info_windcode	eg. 600030.SH
trade_dt	YYYYMMDD
crncy_code	
s_dq_preclose	
s_dq_open	
s_dq_high	
s_dq_low	
s_dq_close	
s_dq_change	
s_dq_pctchange	
s_dq_volume	
s_dq_amount	
s_dq_adjpreclose	
s_dq_adjopen	
s_dq_adjhigh	
s_dq_adjlow	
s_dq_adjclose	
s_dq_adjfactor	
s_dq_avgprice	
s_dq_tradestatus	
sec_id	
opdata	
opmode	

### AShareEODDerivativeIndicator

```
class quant.data.wind.tables.AShareEODDerivativeIndicator(**kwargs)
```

### AIndexEODPrices

```
class quant.data.wind.tables.AIndexEODPrices(**kwargs)
    A
```

### Attributes

object_id	
s_info_windcode	eg. 600030.SH
trade_dt	YYYYMMDD
crncy_code	
s_dq_preclose	
s_dq_open	
s_dq_high	
s_dq_low	
s_dq_close	
s_dq_change	
s_dq_pctchange	
s_dq_volume	
s_dq_amount	
sec_id	
opdata	
opmode	

### AShareIPO

```
class quant.data.wind.tables.AShareIPO(**kwargs)
    AIPO
```

### Attributes

object_id	
s_info_windcode	eg. 600030.SH
s_ipo_listdate	, YYYYMMDD

### AShareST

```
class quant.data.wind.tables.AShareST(**kwargs)
    AST
```

### Attributes

object_id	
s_info_windcode	eg. 600030.SH
s_type_st	ST
entry_dt	ST YYYYMMDD
remove_dt	ST YYYYMMDD
ann_dt	YYYYMMDD

### AShareTTMHis

```
class quant.data.wind.tables.AShareTTMHis(**kwargs)
```



### quant.transform.distribution

**class** `quant.transform.distribution.Transformer` (*distribution='norm'*)  
**Methods**

---

<code>train(data)</code>
<code>transform(data)</code>

---

`quant.transform.distribution.compute_zscore` (*data*, *axis=-1*, *clip=3.0*, *inplace=False*)  
*Z*

**Parameters** *data*

**axis:** `int`  
 features  
**clip:** `float`, optional  
 clip  
**inplace:** `bool`, optional

`quant.transform.distribution.find_extreme_values` (*data*, *distribution='norm'*, *alpha*,  
*pha=0.975*)

**Parameters** *data:* `array_like`

**distribution:** `str`

- :
- norm
- expon
- uniform

**alpha: float**

alpha

**Returns** `pd.Series`

```
quant.transform.distribution.get_residual(y: pandas.core.series.Series, x: pandas.core.series.Series,
                                         estimate_start=None, estimate_end=None, remove_alpha=True)
yx
```

**Parameters** `y: pd.Series`

**x: pd.Series**

**estimate\_start: str (YYYY-MM-DD), optional**

**estimate\_end: str (YYYY-MM-DD), optional**

**remove\_alpha: bool, optional**

True

## quant.utils

Utils that may be used outside the *quantlib* package

```
class quant.utils.LocalizeWrapper (path)
    @localizer.wrap(filename)pandas pd.read_hdf(filename, key)key
```

### Methods

---

```
wrap([filename, update, exclude])
```

---

**wrap** (*filename=None*, *update='overwrite'*, *exclude=None*)

**filename:** str, optional

h5

**update:** {'overwrite', 'append'}, optional

**exclude:** list, optional

Parameters

### Examples

```
@LOCALIZER.wrap ("data")
def get_data (code) :
    ...
```

code="data.h5"

**class** `quant.utils.ConfigManager` (*path*='config.cfg')

#### Methods

<i>add_argument</i> (*args, **kwargs)	'argparse.ArgumentParser.add_argument'
<i>items</i> ()	
<i>keys</i> ()	
<i>update</i> ()	

```

add_argument (*args, **kwargs)
    'argparse.ArgumentParser.add_argument'

items ()

keys ()

update ()

```

## quant.utils.calendar

**class** `quant.utils.calendar.TradingCalendar`

#### Attributes

<i>TradingDay</i>	pd.tseries.offsets.CustomBusinessDay
<i>holidays</i>	

#### Methods

<i>get_holidays</i> ()
------------------------

```

TradingDay
    pd.tseries.offsets.CustomBusinessDay

static get_holidays ()

holidays
    type: List[str]

```



## CHAPTER 6

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



### q

`quant.analysis.ic`, [1](#)  
`quant.transform.distribution`, [9](#)  
`quant.utils`, [11](#)  
`quant.utils.calendar`, [12](#)



## A

`add_argument()` (`quant.utils.ConfigManager` method), 12  
`AIndexEODPrices` (class in `quant.data.wind.tables`), 6  
`AShareEODDerivativeIndicator` (class in `quant.data.wind.tables`), 6  
`AShareEODPrices` (class in `quant.data.wind.tables`), 5  
`AShareIPO` (class in `quant.data.wind.tables`), 7  
`AShareST` (class in `quant.data.wind.tables`), 7  
`AShareTTMHis` (class in `quant.data.wind.tables`), 7

## C

`compute_zscore()` (in module `quant.transform.distribution`), 9  
`ConfigManager` (class in `quant.utils`), 11

## F

`find_extreme_values()` (in module `quant.transform.distribution`), 9

## G

`get_holidays()` (`quant.utils.calendar.TradingCalendar` static method), 12  
`get_ic()` (in module `quant.analysis.ic`), 1  
`get_residual()` (in module `quant.transform.distribution`), 10  
`get_wind_data()` (in module `quant.data.wind`), 5

## H

`holidays` (`quant.utils.calendar.TradingCalendar` attribute), 12

## I

`items()` (`quant.utils.ConfigManager` method), 12

## K

`keys()` (`quant.utils.ConfigManager` method), 12

## L

`LocalizeWrapper` (class in `quant.utils`), 11

## Q

`quant.analysis.ic` (module), 1  
`quant.transform.distribution` (module), 9  
`quant.utils` (module), 11  
`quant.utils.calendar` (module), 12

## T

`TradingCalendar` (class in `quant.utils.calendar`), 12  
`TradingDay` (`quant.utils.calendar.TradingCalendar` attribute), 12  
`Transformer` (class in `quant.transform.distribution`), 9

## U

`update()` (`quant.utils.ConfigManager` method), 12

## W

`wrap()` (`quant.utils.LocalizeWrapper` method), 11