

说明文档

初始化

两种初始化方式，传入参数则可以做多因子模型，不传入参数也可以调用提供的优化接口

```
model = FactorModel(data, factors)
model = FactorModel()
```

参数

- `data`: 数据
- `factors`: 因子列表如 `['ROE', 'ROA', 'ROE_TTM']`，这些因子数据不能出现NA值，不能有多重共线性

一个典型的由原始数据获取 `factor` 的过程

```
data = pd.read_csv('expo_test.csv')

factors = list(data.columns.droplevel([ 'Ticker', 'CompanyCode', 'TickerName', 'SecuCode', 'IndustryName',
                                         'CategoryName', 'Date', 'Month', 'Return', 'PCF'])))

# 去除经过检测后产生多重共线性的因子
factors.remove('AnalystROEAdj')
factors.remove('FreeCashFlow')
```

构建模型

如果要做多因子模型优化，需要调用 `create_factor_model()`，反之则不用，会创建一系列用于多因子模型的私有变量。

```
model.create_factor_model()
```

用 `print_private()` 可以查看这些私有变量的值

```
model.print_private()
```

优化接口

最大化收益

```
max_returns(self, risk, b=None, up=1.0, industry=None, deviate=None,
            returns=None, rs=None)
```

参数:

- `risk`: 年化跟踪误差，或是风险
- `b`: 默认为None，为没有基准，传入时需保证与投资组合成份对齐，长度一致
- `up`: 个股上限，默认为1.0，即100%
- `industry`: 行业哑变量，可以自己传入，也可以通过 `get_industry_dummy()` 获取，自己传入时需要保证对齐
- `deviate`: 行业偏离
- `returns`: 收益向量，不传入则使用内部多因子模型的计算值

- `rs`: 风险结构, 不传入则使用内部多音字模型的计算结果

多因子模型示例一 无基准, 无行业中性

```
model = FactorModel(data, factors)
model.create_factor_model()
print(model.max_returns(0.1, 0.01))
```

多因子模型示例二 有基准, 行业中性

```
model = FactorModel(data, factors)
model.create_factor_model()
B = np.ones(288)/288
industry = model.get_industry_dummy()
print(model.max_returns(0.07, B, 0.01, industry, 0.01))
```

只应用优化接口示例

```
model = FactorModel()
B = your_base
industry = your_industry
returns = your_returns
rs = your_rs
print(model.max_returns(0.07, B, 0.01, industry, 0.01, returns, rs))
```

最小化风险

```
min_risk(self, target_return, b=None, up=1.0, industry=None,
         deviate=None, returns=None, rs=None)
```

- `target_return`: 目标组合收益
- `b`: 默认为None, 为没有基准, 传入时需保证与投资组合成份对齐, 长度一致
- `up`: 个股上限, 默认为1.0, 即100%
- `industry`: 行业哑变量, 可以自己传入, 也可以通过 `get_industry_dummy()` 获取
- `deviate`: 行业偏离
- `returns`: 收益向量, 不传入则使用内部多因子模型的计算结果
- `rs`: 风险结构, 不传入则使用内部多音字模型的计算结果

示例一 无基准, 无行业中性

```
model = FactorModel(data, factors)
model.create_factor_model()
print(model.min_risk(0.1, up=0.01))
```

示例二 有基准, 有行业中性

```
model = FactorModel(data, factors)
model.create_factor_model()
B = np.ones(288)/288
industry = model.get_industry_dummy()
print(model.min_risk(0.1, B, 0.01, industry, 0.01))
```

其他接口

```
set_names(self, freq=None, returns=None, company=None, industry=None,
          factor=None)
```

用于设置需要用到的一系列列名 参数

- `freq`: 用到的时间列名, 默认为 `Month`
- `returns`: 收益列名, 默认为 `Return`
- `company`: 公司或股票列名, 默认为 `CompanyCode`
- `industry`: 行业列名, 默认为 `IndustryName`
- `factor`: 因子列名, 默认为 `None` 或调用构造函数时传入的值

```
get_components(self)
```

用于获取多因子模型中默认用于投资组合的股票成分

```
get_industry_dummy(self)
```

用于获取行业哑变量

```
set_predict_method(self, method, arg=None)
```

用不同方式预测未来因子收益率 参数

- `method`: 预测方法, 包括 `mean`, `ewma`, `hpfilter`
- `arg`: 设置预测方法需要的额外参数

包接口

除了 `FactorModel` 类 提供了优化接口外, `FactorModel.py` 包也提供了优化接口 `min_risk` 及 `max_returns`, 以及 `ewma` 函数, 这些接口可以直接调用。

```
import FactorModel as Fm
Fm.min_risk(your params)
Fm.max_returns(your params)
Fm.ewma()
```