# 如何运用法玛三因子模型？

法玛三因子到底该如何具体操作，最好能够清晰的介绍法玛三因子

关注者

**11**

被浏览

**11,912**

关注问题​写回答

​邀请回答

​好问题 1

​添加评论

​分享

​

## 你可以邀请下面用户，快速获得回答

## 委托刘看山邀请

开启后为你智能邀请潜在的回答者

开启邀请

[](https://www.zhihu.com/people/vita-23-54)

## [阿拉蕾](https://www.zhihu.com/people/vita-23-54)

可能对该问题感兴趣

邀请回答

[](https://www.zhihu.com/org/cang-hai-shu-ju)

## [沧海数据](https://www.zhihu.com/org/cang-hai-shu-ju)

## [​](https://www.zhihu.com/question/48510028)

TA 已帮助 76 人获得解答

邀请回答

[](https://www.zhihu.com/people/61-63-67-88)

## [人情翻覆起波澜](https://www.zhihu.com/people/61-63-67-88)

TA 已帮助 26 人获得解答

邀请回答

[](https://www.zhihu.com/people/qi-qi-60-41-55)

## [七七有话说](https://www.zhihu.com/people/qi-qi-60-41-55)

TA 已帮助 111 人获得解答

邀请回答

更多推荐结果

#### 3 个回答

默认排序

[](https://www.zhihu.com/people/zhi-wang-zhi-hu-fen-bu-qing)

[量化观察](https://www.zhihu.com/people/zhi-wang-zhi-hu-fen-bu-qing)​

​ 关注

4 人赞同了该回答



## 关于法玛三因子模型

[法玛三因子模型](https://zhida.zhihu.com/search?q=法玛三因子模型&zhida_source=entity&is_preview=1)（Fama-French Three-Factor Model）是一种[资本资产定价模型](https://zhida.zhihu.com/search?q=资本资产定价模型&zhida_source=entity&is_preview=1)（Capital Asset Pricing Model，CAPM）的扩展，用于解释股票回报的变异性。该模型由[尤金·法玛](https://zhida.zhihu.com/search?q=尤金·法玛&zhida_source=entity&is_preview=1)（Eugene Fama）和肯尼斯·法rench（Kenneth French）于1992年提出。

该模型考虑了三个因子对股票回报的影响：市场风险因子、[市值因子](https://zhida.zhihu.com/search?q=市值因子&zhida_source=entity&is_preview=1)和价值因子。市场风险因子指的是股票市场整体的风险，可以用市场指数来代表。市值因子是指公司的市值大小，可以用[市值指数](https://zhida.zhihu.com/search?q=市值指数&zhida_source=entity&is_preview=1)来代表。价值因子是指公司的估值水平，可以用账面市值比（B/M）来代表。这三个因子被认为是影响股票回报的重要因素。

法玛三因子模型认为，股票的预期回报可以通过以下公式计算：

E(Ri) = Rf + βi(Market Risk Premium) + si(SMB) + hi(HML)

其中，E(Ri)表示股票i的预期回报，Rf表示[无风险收益率](https://zhida.zhihu.com/search?q=无风险收益率&zhida_source=entity&is_preview=1)，βi表示股票i的市场风险系数，Market Risk Premium表示市场风险因子的回报率减去无风险收益率，si表示股票i的[市值因子系数](https://zhida.zhihu.com/search?q=市值因子系数&zhida_source=entity&is_preview=1)，SMB表示市值因子的超额回报，hi表示股票i的[价值因子系数](https://zhida.zhihu.com/search?q=价值因子系数&zhida_source=entity&is_preview=1)，HML表示价值因子的超额回报。

通过考虑市值和价值因子，法玛三因子模型可以更好地解释股票回报的变异性。该模型被广泛应用于投资组合管理、风险管理和资产定价等领域。

## Python代码实现

由于上述模型需要使用股票的市值和账面市值比等数据，我们需要先获取这些数据。以下是一种获取中国股票市值和账面市值比数据的方法：

import pandas as pd

import tushare as ts

# 获取股票列表

stock\_list = ts.get\_stock\_basics().index.tolist()

# 获取股票市值数据

market\_cap = ts.get\_stock\_basics().loc[:, 'totalAssets']

# 获取股票账面市值比数据

book\_to\_market = ts.get\_stock\_basics().loc[:, 'bvps'] / ts.get\_stock\_basics().loc[:, 'pb']

# 将市值和账面市值比数据合并到一个DataFrame中

data = pd.concat([market\_cap, book\_to\_market], axis=1)

data.columns = ['market\_cap', 'book\_to\_market']

data.index.name = 'code'

data = data.dropna()

接下来，我们可以使用以上获取的市值和账面市值比数据，以及股票收益率数据，来实现法玛三因子模型的计算。以下是一个简单的实现：

import numpy as np

import statsmodels.api as sm

# 获取股票收益率数据

start\_date = '2020-01-01'

end\_date = '2020-12-31'

return\_data = pd.DataFrame()

for code in stock\_list:

try:

stock\_return = ts.pro\_bar(ts\_code=code, start\_date=start\_date, end\_date=end\_date, adj='qfq',

factors=['tor']).set\_index('trade\_date')['pct\_chg']

return\_data[code] = stock\_return

except:

pass

# 计算市场指数收益率

market\_return = ts.pro\_bar(ts\_code='000001.SH', start\_date=start\_date, end\_date=end\_date,

adj='qfq', factors=['tor']).set\_index('trade\_date')['pct\_chg']

market\_return.name = 'market\_return'

# 将股票收益率和市场指数收益率合并到一个DataFrame中

data = pd.concat([return\_data, market\_return], axis=1)

data = data.dropna()

# 计算超额收益率

data = data.sub(data['market\_return'], axis=0)

# 将市值和账面市值比数据合并到一个DataFrame中

data = pd.concat([data, market\_cap, book\_to\_market], axis=1)

data.columns = stock\_list + ['market\_return', 'market\_cap', 'book\_to\_market']

# 计算因子收益率

factor\_data = pd.DataFrame()

factor\_data['market\_factor'] = market\_return - ts.pro\_bar(ts\_code='000016.SH', start\_date=start\_date,

end\_date=end\_date, adj='qfq',

factors=['tor']).set\_index('trade\_date')['pct\_chg']

[factor\_data](https://zhida.zhihu.com/search?q=factor_data&zhida_source=entity&is_preview=1)['size\_factor'] = sm.OLS(np.array(data.mean()), sm.add\_constant(np.log(data['market\_cap']))).fit().resid

factor\_data['value\_factor'] = sm.OLS(np.array(data.mean()), sm.add\_constant(np.log(data['book\_to\_market']))).fit().resid

# 计算法玛三因子模型的参数

X = sm.add\_constant(factor\_data)

model = sm.OLS(np.array(data.mean()), X)

results = model.fit()

print(results.summary())

以上代码中，我们使用了tushare库获取股票数据，并使用statsmodels库进行回归分析。具体来说，我们首先获取了股票收益率、市场指数收益率、股票市值和账面市值比等数据，然后计算了超额收益率，并将这些数据合并到一个DataFrame中。接着，我们计算了市场因子、规模因子和价值因子的收益率，并利用OLS回归分析计算了法玛三因子模型的参数。

需要注意的是，以上代码只是一个简单的示例，实际应用中还需要考虑很多其他因素，如数据处理、缺失值处理、[数据标准化](https://zhida.zhihu.com/search?q=数据标准化&zhida_source=entity&is_preview=1)等。此外，法玛三因子模型也有其局限性，如不能很好地解释一些股票市场现象等。因此，在实际应用中需要根据具体情况进行调整和改进。

## 因子有效性检验

当使用法玛三因子模型时，我们需要确认市场因子、规模因子和价值因子是否有效。以下是几种在Python中进行因子有效性检验的方法：

1 因子收益率的t检验

首先，我们可以使用t检验来检验市场因子、规模因子和价值因子的收益率是否显著不为零。我们可以通过计算每个因子收益率的t统计量，并检验其显著性水平，来判断该因子是否有效。以下是一个简单的示例：

import pandas as pd

import numpy as np

import statsmodels.api as sm

# 获取股票收益率数据

start\_date = '2020-01-01'

end\_date = '2020-12-31'

return\_data = pd.DataFrame()

for code in stock\_list:

try:

stock\_return = ts.pro\_bar(ts\_code=code, start\_date=start\_date, end\_date=end\_date, adj='qfq',

factors=['tor']).set\_index('trade\_date')['pct\_chg']

return\_data[code] = stock\_return

except:

pass

# 计算市场指数收益率

market\_return = ts.pro\_bar(ts\_code='000001.SH', start\_date=start\_date, end\_date=end\_date,

adj='qfq', factors=['tor']).set\_index('trade\_date')['pct\_chg']

market\_return.name = 'market\_return'

# 将股票收益率和[市场指数收益率](https://zhida.zhihu.com/search?q=市场指数收益率&zhida_source=entity&is_preview=1)合并到一个DataFrame中

data = pd.concat([return\_data, market\_return], axis=1)

data = data.dropna()

# 计算超额收益率

data = data.sub(data['market\_return'], axis=0)

# 将市值和账面市值比数据合并到一个DataFrame中

data = pd.concat([data, market\_cap, book\_to\_market], axis=1)

data.columns = stock\_list + ['market\_return', 'market\_cap', 'book\_to\_market']

# 计算因子收益率

factor\_data = pd.DataFrame()

factor\_data['market\_factor'] = market\_return - ts.pro\_bar(ts\_code='000016.SH', start\_date=start\_date,

end\_date=end\_date, adj='qfq',

factors=['tor']).set\_index('trade\_date')['pct\_chg']

factor\_data['size\_factor'] = sm.OLS(np.array(data.mean()), sm.add\_constant(np.log(data['market\_cap']))).fit().resid

factor\_data['value\_factor'] = sm.OLS(np.array(data.mean()), sm.add\_constant(np.log(data['book\_to\_market']))).fit().resid

# 计算t检验的p值

ttest = sm.stats.ttest\_ind(factor\_data, np.zeros(factor\_data.shape), axis=0)

# 打印检验结果

print(ttest)

以上代码中，我们首先计算了每个因子的收益率，然后使用OLS回归分析计算了每个因子的残差，即每个因子收益率中的alpha值。接着，我们计算了每个因子收益率的t统计量和p值，并输出检验结果。

2 因子回归的 $R^2$ 值

除了t检验之外，我们还可以[使用因子](https://zhida.zhihu.com/search?q=使用因子&zhida_source=entity&is_preview=1)回归的 $R^2$ 值来评估市场因子、[规模因子](https://zhida.zhihu.com/search?q=规模因子&zhida_source=entity&is_preview=1)和价值因子的有效性。$R^2$ 值表示模型中因子收益率对总收益率的解释程度，值越高则说明因子对总收益率的解释程度越大。以下是一个简单的示例：

import pandas as pd

import numpy as np

import statsmodels.api as sm

# 获取股票收益率数据

start\_date = '2020-01-01'

end\_date = '2020-12-31'

return\_data = pd.DataFrame()

for code in stock\_list:

try:

stock\_return = ts.pro\_bar(ts\_code=code, start\_date=start\_date, end\_date=end\_date, adj='qfq',

factors=['tor']).set\_index('trade\_date')['pct\_chg']

return\_data[code] = stock\_return

except:

pass

# 计算市场指数收益率

market\_return = ts.pro\_bar(ts\_code='000001.SH', start\_date=start\_date, end\_date=end\_date,

adj='qfq', factors=['tor']).set\_index('trade\_date')['pct\_chg']

market\_return.name = 'market\_return'

# 将股票收益率和市场指数收益率合并到一个DataFrame中

data = pd.concat([return\_data, market\_return], axis=1)

data = data.dropna()

# 计算超额收益率

data = data.sub(data['market\_return'], axis=0)

# 将市值和账面市值比数据合并到一个DataFrame中

data = pd.concat([data, market\_cap, book\_to\_market], axis=1)

data.columns = stock\_list + ['market\_return', 'market\_cap', 'book\_to\_market']

# 计算因子收益率

factor\_data = pd.DataFrame()

factor\_data['market\_factor'] = market\_return - ts.pro\_bar(ts\_code='000016.SH', start\_date=start\_date,

end\_date=end\_date, adj='qfq',

factors=['tor']).set\_index('trade\_date')['pct\_chg']

factor\_data['size\_factor'] = sm.OLS(np.array(data.mean()), sm.add\_constant(np.log(data['market\_cap']))).fit().resid

factor\_data['value\_factor'] = sm.OLS(np.array(data.mean()), sm.add\_constant(np.log(data['book\_to\_market']))).fit().resid

# 计算因子回归的[R2值](https://zhida.zhihu.com/search?q=R2值&zhida_source=entity&is_preview=1)

y = data.mean()

X = sm.add\_constant(factor\_data)

model = sm.OLS(y, X).fit()

rsquared = model.rsquared

# 打印检验结果

print([rsquared](https://zhida.zhihu.com/search?q=rsquared&zhida_source=entity&is_preview=1))

以上代码中，我们首先计算了每个因子的收益率，然后使用OLS回归分析计算了每个因子的残差，即每个因子收益率中的alpha值。接着，我们将股票超额收益率和因子收益率合并到一个DataFrame中，然后计算因子回归的 $R^2$ 值。最后，我们输出了检验结果。

3 [因子相关性分析](https://zhida.zhihu.com/search?q=因子相关性分析&zhida_source=entity&is_preview=1)

除了单独检验每个因子的有效性之外，我们还可以使用因子相关性分析来评估每个因子对投资组合表现的贡献程度。这里我们使用热图来显示每个因子之间的相关性。以下是一个简单的示例：

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

# 获取股票收益率数据

start\_date = '2020-01-01'

end\_date = '2020-12-31'

return\_data = pd.DataFrame()

for code in stock\_list:

try:

stock\_return = ts.pro\_bar(ts\_code=code, start\_date=start\_date, end\_date=end\_date, adj='qfq',

factors=['tor']).set\_index('trade\_date')['pct\_chg']

return\_data[code] = stock\_return

except:

pass

# 计算市场指数收益率

market\_return = ts.pro\_bar(ts\_code='000001.SH', start\_date=start\_date, end\_date=end\_date,

adj='qfq', factors=['tor']).set\_index('trade\_date')['pct\_chg']

market\_return.name = 'market\_return'

# 将股票收益率和市场指数收益率合并到一个DataFrame中

data = pd.concat([return\_data, market\_return], axis=1)

data = data.dropna()

# 计算超额收益率

data = data.sub(data['market\_return'], axis=0)

# 将市值和账面市值比数据合并到一个DataFrame中

data = pd.concat([data, market\_cap, book\_to\_market], axis=1)

data.columns = stock\_list + ['market\_return', 'market\_cap', 'book\_to\_market']

# 计算因子收益率

factor\_data = pd.DataFrame()

factor\_data['market\_factor'] = market\_return - ts.pro\_bar(ts\_code='000016.SH', start\_date=start\_date,

end\_date=end\_date, adj='qfq',

factors=['tor']).set\_index('trade\_date')['pct\_chg']

factor\_data['size\_factor'] = sm.OLS(np.array(data.mean()), sm.add\_constant(np.log(data['market\_cap']))).fit().resid

factor\_data['value\_factor'] = sm.OLS(np.array(data.mean()), sm.add\_constant(np.log(data['book\_to\_market']))).fit().resid

# 计算因子相关性

[corr\_matrix](https://zhida.zhihu.com/search?q=corr_matrix&zhida_source=entity&is_preview=1) = factor\_data.corr()

# 绘制热图

sns.heatmap(corr\_matrix, annot=True, cmap='RdYlBu')

plt.show()

以上代码中，我们首先计算了每个因子的收益率和超额收益率，然后将这些数据合并到一个DataFrame中。接着，我们使用OLS回归分析计算了每个因子的残差，即每个因子收益率中的alpha值。然后，我们计算了每个因子之间的相关性，并使用热图可视化了结果。

## 法玛三因子模型的优缺点

法玛三因子模型的优点：

1. 解释力强：相较于CAPM模型，法玛三因子模型的解释力更强，能够更好地解释股票收益率的变化。
2. 考虑了多个因素：相较于单因子模型，法玛三因子模型考虑了市场因子、规模因子和价值因子，更全面地考虑了股票收益率的影响因素。
3. 可解释性强：法玛三因子模型中的三个因子，即市场因子、规模因子和价值因子，都是经济学上有实际意义的因素，因此其结果更容易被解释。

法玛三因子模型的缺点：

1. 忽略了其他因素：法玛三因子模型只考虑了市场因子、规模因子和价值因子，忽略了其他可能对股票收益率有影响的因素，如流动性、动量等。
2. 样本限制：法玛三因子模型的样本通常是历史股票数据，而历史表现并不能保证未来表现，因此其预测能力有限。
3. 可能存在共线性问题：法玛三因子模型中的因子可能存在共线性问题，导致其解释能力下降。
4. 不适用于所有市场：法玛三因子模型的适用范围有限，可能无法适用于所有市场。例如，一些新兴市场可能存在不同的因子影响股票收益率，无法使用法玛三因子模型来解释其表现。

综上所述，法玛三因子模型虽然具有一定的局限性，但在投资组合管理和股票选择方面仍然具有一定的实用性和可靠性。

## 如何改进提升

虽然法玛三因子模型具有较好的解释股票收益率的能力，但是其仍然存在一些缺点和局限性。下面是一些改进和提升三因子模型的方法：

1 添加其他因子：法玛三因子模型只考虑了市场因子、规模因子和价值因子，可以添加其他因子，如动量、流动性等，来提升模型的解释能力。可以使用pyfolio库中的get\_factor\_returns函数获取更多的因子数据，例如动量因子和波动率因子：

import pyfolio as pf

start\_date = '2015-01-01'

end\_date = '2021-12-31'

tickers = ['AAPL', 'MSFT', 'AMZN', 'GOOG', 'FB']

factor\_names = ['market\_beta', 'size\_factor', 'value\_factor', 'momentum\_factor', 'volatility\_factor']

factor\_data = pf.utils.get\_factor\_returns(factor\_names, start\_date=start\_date, end\_date=end\_date)

2 考虑时间变化：股票市场中因子的影响可能会随着时间变化而变化，可以建立时间变化的因子模型，或者采用滚动回归来考虑时间变化对因子的影响。 可以使用rolling函数进行滚动回归，并将时间窗口设置为1年或更长时间：

import pandas as pd

import statsmodels.api as sm

rolling\_window = 252

factor\_data\_rolling = pd.DataFrame(index=factor\_data.index)

for factor\_name in factor\_names:

factor\_data\_rolling[factor\_name] = factor\_data[factor\_name].rolling(window=rolling\_window).apply(lambda x: sm.OLS(x, sm.add\_constant(factor\_data[['market\_beta', 'size\_factor', 'value\_factor']]).loc[x.index]).fit().params)

factor\_data\_rolling = factor\_data\_rolling.dropna()

3 考虑非线性关系：股票收益率和因子之间可能存在非线性关系，可以使用非线性[回归模型](https://zhida.zhihu.com/search?q=回归模型&zhida_source=entity&is_preview=1)来建立因子模型，或者使用机器学习方法来建立预测模型。 可以使用scikit-learn库中的[多项式回归](https://zhida.zhihu.com/search?q=多项式回归&zhida_source=entity&is_preview=1)模型来建立非线性关系的因子模型：

from sklearn.linear\_model import LinearRegression

from sklearn.preprocessing import PolynomialFeatures

poly\_degree = 2

poly\_features = PolynomialFeatures(poly\_degree, include\_bias=False)

X\_poly = poly\_features.fit\_transform(factor\_data[['market\_beta', 'size\_factor', 'value\_factor']])

model = LinearRegression().fit(X\_poly, factor\_data['returns'])

或者使用scikit-learn库中的机器学习模型来建立因子模型，例如随机森林模型：

from sklearn.ensemble import RandomForestRegressor

model = RandomForestRegressor(n\_estimators=100, random\_state=0)

model.fit(factor\_data[['market\_beta', 'size\_factor', 'value\_factor']], factor\_data['returns'])

4 解决共线性问题：法玛三因子模型中的因子可能存在共线性问题，可以使用[主成分分析](https://zhida.zhihu.com/search?q=主成分分析&zhida_source=entity&is_preview=1)等方法来减少因子之间的共线性，提高模型的解释能力。 可以使用scikit-learn库中的主成分分析模型来减少因子之间的共线性：

from sklearn.decomposition import PCA

n\_components = 3

pca = PCA(n\_components=n\_components)

X\_pca = pca.fit\_transform(factor\_data[['market\_beta', 'size\_factor', 'value\_factor']])

5 使用更多数据：使用更多的数据来建立因子模型，可以提高模型的预测能力和稳健性。 可以使用Quandl等数据源获取更多的历史数据来建立因子模型：

import quandl

quandl.ApiConfig.api\_key = 'your\_api\_key'

data = quandl.get\_table('SHARADAR/SF1', ticker=tickers, dimension='MRY', qopts={'columns': ['ticker', 'date', 'marketcap', 'roe', 'pb']})

data = data.pivot(index='date', columns='ticker')

data.columns = [f"{col[0]}\_{col[1]}" for col in data.columns]

data = data.dropna()

6 考虑国别和行业因素：股票收益率受到不同国别和行业因素的影响，可以建立考虑国别和行业因素的[多因子模型](https://zhida.zhihu.com/search?q=多因子模型&zhida_source=entity&is_preview=1)，来提高模型的解释能力。 可以使用pyfolio库中的get\_industry\_returns函数获取行业因子数据，并使用alpha\_vantage等数据源获取国别因子数据：

import alpha\_vantage

from alpha\_vantage.timeseries import TimeSeries

import pyfolio as pf

# 使用alpha\_vantage获取国别因子数据

ts = TimeSeries(key='YOUR\_API\_KEY', output\_format='pandas')

data, meta\_data = ts.get\_daily(symbol='SPY', outputsize='full')

data.columns = ['open', 'high', 'low', 'close', 'volume']

data = data[['close']]

data = data.pct\_change().dropna()

data.columns = ['market\_factor']

# 使用pyfolio获取行业因子数据

industry\_data = pf.utils.get\_industry\_returns('morningstar', 'usa')

industry\_data.columns = ['industry\_factor']

# 合并国别和行业因子数据

factor\_data = pd.concat([data, industry\_data], axis=1).dropna()

factor\_data = factor\_data.resample('M').last()

# 运用法玛三因子模型进行分析

...

7 考虑投资组合构建方法：可以使用优化模型来构建投资组合，例如使用CVXPY库中的优化模型：

pythonCopy codeimport cvxpy as cp

weights = cp.Variable(3)

constraints = [cp.sum(weights) == 1, weights >= 0]

expected\_return = factor\_data\_rolling['returns'].mean()

cov\_matrix = factor\_data\_rolling[['market\_beta', 'size\_factor', 'value\_factor']].cov()

risk = cp.quad\_form(weights, cov\_matrix)

objective = cp.Minimize(risk - 0.5 \* gamma \* expected\_return \* cp.quad\_form(weights, cov\_matrix))

problem = cp.Problem(objective, constraints)

problem.solve()