

CAPM 模型及其 PYTHON 实现

01 基本概念

资本资产定价模型（CAPM）描述了资产的预期回报与市场系统风险之间的关系。它由威廉·夏普（1990 年的诺贝尔经济学奖）等人在 1960 年代提出。

CAPM 被认为是经济学的七个基本理论之一。

CAPM 表示资产的预期收益等于无风险收益加上风险溢价。CAPM 的假设是投资者是理性的，希望获得最大化回报并尽可能降低风险。因此，CAPM 的目标是计算相对于无风险利率的给定风险溢价，投资者可以预期获得的回报。

Risk-Free Rate（无风险利率）

当投资者决定买入股票等高风险资产时，他的目的是为了获得高于无风险资产的收益。

一般认为，银行存款利率和国债收益率都是无风险的。不过，一般同期的国债收益率会高于同期银行存款，所以我们常把国债收益率当成无风险利率。

不同期限的国债利率收益不同，一般对标时，使用与风险资产预期投资时间相同就好。

比如，如果计划投资于流动性好的股票，也不打算长期持有，则我们可以使用一年期国债利率作为无风险利率。如果是不动产，则可以使用 5 年期国债利率作为对标的无风险利率。

下面的代码将演示如何通过 akshare 来获取国债收益率，并以此作为我们的无风险收益率：

```
import akshare as ak
import arrow

now = arrow.now()
start = now.shift(years=-1)
end = f"{now.year}{now.month:02d}{now.day:02d}"
start = f"{start.year}{start.month:02d}{start.day:02d}"

bond = ak.bond_china_yield(start_date=start,
                           end_date=end)
bond.set_index(keys='曲线名称', inplace=True)
bond
```

这样我们就得到了近一年的各种债券收益率。输出结果如下图所示：

	日期	3月	6月	1年	3年	5年	7年	10年	30年
曲线名称									
中债中短期票据收益率曲线(AAA)	2022-10-24	1.8835	2.0222	2.0961	2.5554	2.9111	3.1232	3.3302	NaN
中债商业银行普通债收益率曲线(AAA)	2022-10-24	1.8230	1.9540	2.0759	2.4538	2.8003	2.9842	3.1511	3.7295
中债国债收益率曲线	2022-10-24	1.6723	1.7501	1.7700	2.2794	2.4852	2.7104	2.7153	3.1416
中债国债收益率曲线	2022-10-25	1.7101	1.7601	1.7945	2.2899	2.5031	2.7206	2.7278	3.1486
中债中短期票据收益率曲线(AAA)	2022-10-25	1.9203	2.0524	2.1126	2.5778	2.9112	3.1262	3.3331	NaN
...
中债商业银行普通债收益率曲线(AAA)	2023-10-20	2.3921	2.5367	2.6457	2.8377	2.8949	3.0153	3.1036	3.2407
中债国债收益率曲线	2023-10-20	2.1802	2.2888	2.3107	2.5010	2.5982	2.7150	2.7052	2.9962
中债中短期票据收益率曲线(AAA)	2023-10-23	2.5586	2.6325	2.6915	2.9598	3.1339	3.2636	3.3440	NaN
中债商业银行普通债收益率曲线(AAA)	2023-10-23	2.3910	2.5346	2.6384	2.8195	2.9003	3.0208	3.1087	3.2407
中债国债收益率曲线	2023-10-23	2.1802	2.2688	2.3007	2.4895	2.5887	2.7125	2.7047	2.9903

我们可以把“中债国债收益率曲线”一年期的平均值当作一年期国债收益：

```
rf = bond[bond.index=='中债国债收益率曲线']['1 年'].mean()
print(rf)

rf = rf / 100
```

输出是 2.06。这个值应该解读为 2.06%。

市场回报 r_m

市场回报率表示为 r_m ，包括市场上的所有证券。但一般我们只使用某些指数，比如上证 50，沪深 300。如果投资偏好成长股，则可以使用中证 1000。

贝塔 β

β 是衡量股票相对于整体市场（例如沪深 300 指数）波动性的指标。换句话说， β 代表回归线的斜率，即市场回报与个股回报的关系。

CAPM 中使用 β 来描述系统风险或市场风险与资产预期回报之间的关系。根据定义，我们说整个市场的贝塔值为 1.0，个股根据其相对于市场的波动程度进行排名。

- 如果个股的 $\text{Beta} = 1.0$ ，这意味着其价格与市场完全相关
- 如果 $\text{Beta} < 1.0$ （称为“防御性”），这表明该证券理论上的波动性低于市场
- 如果 $\text{Beta} > 1.0$ 或“激进”，则表明资产价格比市场波动更大

CAPM 公式

该公式定义如下：

$$r_i = r_f + \beta(r_m - r_f)$$

这里：

- r_i 是证券（个股）的预期回报
- r_f 是无风险利率
- β_i 是证券相对于市场的 β 值
- $r_m - r_f$ 被称为风险溢价

我们通过一个例子来解读这个公式。如果标普 500 的整体回报率是 12.4%，无风险利率为 0%，而 APPL 的 β 为 1.1 的话，则投资者买入 APPL，他期望获得 13.7% 的回报，以补偿承担的额外风险。

要使用 CAPM 模型，核心是计算个股相对于市场组合（指数）的 β 。下面我们就通过 Python 来进行实现。

基于 Python 的 CAPM 实现

我们使用的市场组合是沪深 300，因此，我们也要从中抽取个股。我们将随机抽取 10 支个股来进行计算。

获取数据

为了确保所有人都能拿到数据，我们仍然使用 akshare。

首先，我们通过 akshare 获取过去一年的沪深 300 的行情数据：

```
import akshare as ak

hs300 = ak.stock_zh_index_daily(symbol="sz399300")
print(hs300)
```

我们对过去一年的沪深 300 的收益情况进行速览：

```
now = arrow.now()
year_ago = now.shift(years = -1)

filter = hs300.index >= np.datetime64(year_ago)
year_ago = hs300[filter].index[0]
# PRINT(YEAR_AGO)

# 计算买入并持有的收益（最近一年）
filter = hs300.index == year_ago
buy_price = hs300[filter].iloc[0]["close"]
buy_and_hold = hs300["close"][-1]/buy_price - 1
print(f"买入并持收益: {buy_and_hold:.2%}")

# 通过均值推算年化收益
market_returns = hs300.pct_change()["close"].dropna()
filter = returns.index >= year_ago
market_annual = (1 + returns[filter].mean()) ** 242 - 1
print(f"年化收益: {market_annual:.2%}")
```

可以看出，过去一年里，以买入并持有法计，沪深 300 的收益是-4.22%；如果按每日收益取均值，再年化，则得到收益是-3.31%，两者相差不大。

接下来，我们获取沪深 300 成份股，以便从中抽取个股进行检验：

```
import akshare as ak

index_stock_cons_df = ak.index_stock_cons("399300")
print(index_stock_cons_df)
```

我们将得到如下输出：

	品种代码	品种名称	纳入日期
0	600754	锦江酒店	2023-06-12
1	600732	爱旭股份	2023-06-12
2	688223	晶科能源	2023-06-12
3	601872	招商轮船	2023-06-12
4	601607	上海医药	2023-06-12
..
295	600660	福耀玻璃	2005-04-08
296	600690	青岛海尔	2005-04-08
297	600741	巴士股份	2005-04-08
298	600795	国电电力	2005-04-08
299	600900	长江电力	2005-04-08

接下来，我们随机取 10 支股票，获取行情，并计算每日收益率：

```
np.random.seed(78)

stocks = index_stock_cons_df['品种代码'].to_list()
stocks = random.sample(stocks, 10)

frames = {}

now = arrow.now()
start = now.shift(years = -1)
end = now.format("YYYYMMDD")
start = start.format("YYYYMMDD")

# 获取 10 支股票的行情数据
for code in stocks:
    bars = ak.stock_zh_a_hist(symbol=code,
                               period="daily",
                               start_date=start,
                               end_date=end,
                               adjust="qfq")

    bars.index = pd.to_datetime(bars["日期"])
    frames[code] = bars["收盘"]
```

```
# 与指数行情数据合并
start = np.datetime64(now.shift(years = -1))
frames["399300"] = hs300[hs300.index >= start]["close"]

df = pd.DataFrame(frames)

# 计算每日收益
returns = df.pct_change()

# 如果存在 NAN, 则后面的回归法将无法聚合
returns.dropna(how='any', inplace=True)
returns.style.format('{:,.2%}')
```

我们得到的每日个股及指数涨跌数据如下：

	601216	000425	002466	600111	601615	601800	601398	002714	002648	300896	399300
2022-10-26 00:00:00	1.34%	2.52%	0.34%	0.75%	-1.67%	-0.15%	-1.49%	-6.49%	6.13%	7.93%	0.81%
2022-10-27 00:00:00	0.26%	1.79%	0.05%	1.48%	-4.89%	0.29%	-0.75%	-2.46%	-0.67%	-5.99%	-0.70%
2022-10-28 00:00:00	-2.63%	-1.98%	-3.46%	-5.31%	-0.95%	-3.06%	0.00%	-4.06%	-3.92%	-1.79%	-2.47%
2022-10-31 00:00:00	0.00%	0.45%	3.48%	-2.32%	-5.38%	-2.85%	-2.53%	-1.60%	-2.59%	-2.25%	-0.92%
2022-11-01 00:00:00	2.97%	2.23%	5.00%	10.07%	2.52%	4.02%	-1.82%	3.69%	5.73%	8.17%	3.58%
2022-11-02 00:00:00	3.67%	0.44%	-0.85%	4.50%	-0.83%	0.15%	-0.26%	-0.48%	0.08%	3.11%	1.20%
2022-11-03 00:00:00	-2.28%	0.87%	-2.74%	-1.16%	-2.44%	-0.89%	-0.27%	-1.37%	2.67%	1.03%	-0.81%
2022-11-04 00:00:00	1.55%	0.86%	8.74%	2.93%	6.56%	1.65%	0.80%	2.35%	0.67%	4.84%	3.27%

计算 β

我们将通过两种方式来计算 β 。一种是回归法，一种是协方差法。

回归法

在 numpy 中有一个 polyfit 函数，可以用来进行多项式拟合。

当我们使用一次项拟合，那么得到的系数就是要求的 β 。

```
cols = df.columns
for name in cols:
    x = returns[name]
    y = returns["399300"]
    beta, alpha = np.polyfit(x, y, deg=1)
    print(name, f"{beta:.2%} {alpha:.2%}")
```

从结果可以看出，有两支股票存在正的 alpha，同时还存在 20%以上的 beta 收益。

现在，我们就来看看，如果买入这两支股票，它们的预期收益应该是多少：

这里要注意，我们使用的 risk_free 是年化收益，因此我们最终计算出来的预期收益，也应该是年化（或者都统一到日化）：

```
code = "601615"
beta = params[code][0]

# 回归法得到的预期收益
er = rf + beta * (market_annual - rf)
print(f"code beta: {beta:.2f}, Er: {er:.2%}")
```

最终我们得到 601615 的 beta 是 0.23，一年期的预期收益是 0.83 %左右。

协方差法

```
params = {}

for name in cols:
    cov = np.cov(returns[name], returns["399300"])
    beta = cov[0,1]/cov[1,1]

    er = rf + beta * (market_annual - rf)
    print(f"{name} beta: {beta:.2%}, Er: {er:.2%}")
    params[name] = beta

code = "601615"
beta = params[code]
```

```
# 协方差法得到的预期收益
er = rf + beta * (market_annual - rf)
print(f"code beta: {beta:.2f}, Er: {er:.2%}")
```

这样得到 601615 的 beta 是 1.03，年化预期收益是-3.46%，与沪深 300 接近。

回顾

内容摘自《大富翁量化课程》。文章重点：

1. 介绍 CAPM 模型及重点概念
2. 分别用回归法和协方差法计算 CAPM 中的 β 参数。

3. 掌握`random.sample`, `stock_zh_a_hist`, `pd.to_datetime`, `np.seed`, `np.datetime64`, `pct_change`, `dropna`, `df.style.format`等方法。为什么要使用这些方法？在我们课程中有讲。

思考

1. 使用协方差与回归法，算出来的 β 值不一样。哪一种方法更鲁棒？
2. 使用CAPM需要满足哪些条件？你认为它的可复现性怎么样？