Lecture 9: Estimation of CAPM

Douglas Chung

National Chengchi University
19 November 2021

1 Estimation of CAPM

This note will demonstrate how to estimate CAPM with linear regression. We first obtain returns of beta-sorted portfolios, the market risk premium, and the risk-free rate from Kenneth R. French's Data Library.

[1]:
```
import numpy as np
import pandas as pd
import pandas_datareader as pdr

factor = pdr.data.DataReader('F-F_Research_Data_Factors', 'famafrench',␣
↪start='1-1-1926')[0]
data = pdr.data.DataReader('Portfolios_Formed_on_BETA', 'famafrench',␣
↪start='1-1-1926')
print(data['DESCR'])
```

```
Portfolios Formed on BETA
-------------------------

This file was created by CMPT_BETA_RETS using the 202106 CRSP database. It
contains value- and equal-weighted returns for portfolios formed on BETA. The
portfolios are constructed at the end of June. Beta is estimated using monthly
returns for the past 60 months (requiring at least 24 months with non-missing
returns). Beta is estimated using the Scholes-Williams method. Annual returns
are from January to December. Missing data are indicated by -99.99 or -999. The
break points include utilities and include financials. The portfolios include
utilities and include financials. Copyright 2021 Kenneth R. French

  0 : Value Weighted Returns -- Monthly (696 rows x 15 cols)
  1 : Equal Weighted Returns -- Monthly (696 rows x 15 cols)
  2 : Value Weighted Returns -- Annual from January to December (57 rows x 15
cols)
  3 : Equal Weighted Returns -- Annual from January to December (57 rows x 15
cols)
  4 : Number of Firms in Portfolios (696 rows x 15 cols)
  5 : Average Firm Size (696 rows x 15 cols)
  6 : Value-Weighted Average of Prior Beta (58 rows x 15 cols)
```

1.1 Preview the data

[2]:
```
print(data[0].head(1))
print(factor.head(1))
```

```
         Lo 20  Qnt 2  Qnt 3  Qnt 4  Hi 20  Lo 10  Dec 2  Dec 3  Dec 4  Dec 5  \
Date
1963-07   1.13  -0.08  -0.97  -0.94  -1.41   1.35   0.77   0.08  -0.24  -0.69

         Dec 6  Dec 7  Dec 8  Dec 9  Hi 10
Date
1963-07   -1.2  -0.49  -1.39  -1.94  -0.77
         Mkt-RF  SMB   HML   RF
Date
1926-07    2.96 -2.3 -2.87  0.22
```

1.2 Prepare the data

Next, we have to select the sampling period and combine data sets. In particular, we only need returns of 10 beta sorted decile portfolios, the market excess returns, and the risk free rates over the pre-specified period.

[3]:
```
df = data[0]
start = '1926'
finish = '2021'
df = df.loc[start:finish,"Lo 10":"Hi 10"]
df = df.join(factor[['Mkt-RF','RF']])
df.head()
```

[3]:

Date	Lo 10	Dec 2	Dec 3	Dec 4	Dec 5	Dec 6	Dec 7	Dec 8	Dec 9	Hi 10
1963-07	1.35	0.77	0.08	-0.24	-0.69	-1.20	-0.49	-1.39	-1.94	-0.77
1963-08	3.52	3.89	4.29	5.25	5.23	7.55	7.57	4.91	9.04	10.47
1963-09	-3.09	-2.24	-0.54	-0.97	-1.37	-0.27	-0.63	-1.00	-1.92	-3.68
1963-10	1.25	-0.12	2.00	5.12	2.32	1.78	6.63	4.78	3.10	3.01
1963-11	-0.91	-0.15	1.60	-2.05	-0.94	-0.69	-1.32	-0.51	-0.20	0.52

Date	Mkt-RF	RF
1963-07	-0.39	0.27
1963-08	5.07	0.25
1963-09	-1.57	0.27
1963-10	2.53	0.29
1963-11	-0.85	0.27

1.3 Linear regression with ordinary least squares

To estimate CAPM, we have to run the following time-series regression (TSR) by regressing the excess returns of the asset i on the market excess returns m.

$r_{i,t} - r_{f,t} = \alpha_i + \beta_i(r_{m,t} - r_{f,t}) + \varepsilon_{i,t}$

Let $R_{i,t}$ be the excess returns, then:

$R_{i,t} = \alpha_i + \beta_i R_{m,t} + \varepsilon_{i,t}$

We will demonstrate how to run regressions with the "statsmodels" package. If we want to state the regression equation explicitly, we can use the "smf.ols" method.

[4]:
```
import statsmodels.formula.api as smf

Y = df['Hi 10'] - df['RF']
X = df['Mkt-RF']

reg_df = pd.DataFrame([Y, X], index=['R', 'MKT']).T

TSR = smf.ols('R ~ 1 + MKT', data=reg_df).fit()
YHat = TSR.fittedvalues
print(TSR.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      R   R-squared:                       0.812
Model:                            OLS   Adj. R-squared:                  0.812
Method:                 Least Squares   F-statistic:                     2999.
Date:                Sun, 12 Sep 2021   Prob (F-statistic):          4.01e-254
Time:                        17:57:11   Log-Likelihood:                -1845.5
No. Observations:                 696   AIC:                             3695.
Df Residuals:                     694   BIC:                             3704.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.1922      0.131     -1.463      0.144      -0.450       0.066
MKT            1.6020      0.029     54.762      0.000       1.545       1.659
==============================================================================
Omnibus:                       49.683   Durbin-Watson:                   1.887
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               92.503
Skew:                           0.470   Prob(JB):                     8.19e-21
Kurtosis:                       4.519   Cond. No.                         4.53
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
```

specified.

[27]:
```
import matplotlib.pyplot as plt

plt.style.use('seaborn')
fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(1,1,1)
plt.axhline(0, color='xkcd:gray', linestyle='-', label='_nolegend_', linewidth=1)
plt.axvline(0, color='xkcd:gray', linestyle='-', label='_nolegend_', linewidth=1)
plt.plot(X,Y,'o', alpha=0.5, markersize=5)
plt.plot(X,YHat, color='xkcd:orange', alpha=0.75)
plt.ylabel('Excess returns of an asset')
plt.xlabel('Excess returns of the market portfolio')
plt.title('Scatter plot of data points and the regression line')
plt.legend(['Data points', 'Regression line'], loc='best')
plt.show()
```

1.4 Estimate betas for all portfolios

Alternatively, we can run regressions with the “sm.OLS” method. This method does not include an intercept by default. Therefore, we have to add it by the “sm.add_constant” function. Now, let’s estimate betas for the 10 beta-sorted portfolios. Moreover, we need to compute these portfolios’ average excess returns:

$\mathbb{E}[R_i] = \bar{R}_i = \frac{R_{i,t}}{N}$

as well as their excess returns implied by their CAPM betas:

$\mathbb{E}[R_{i,CAPM}] = \hat{\beta}_i \mathbb{E}[R_m]$

[6]:
```
import statsmodels.api as sm

MKT = df['Mkt-RF']
RF = df['RF']

AvgMkt = MKT.mean()
AvgRF = RF.mean()

BETA = []
AvgR = []
AvgRHat = []

for c in range(10):
    R = df.iloc[:,c].subtract(RF, axis=0)
    TSR = sm.OLS(R, sm.add_constant(MKT)).fit()
    BETA.append(TSR.params[1])

    AvgR.append(R.mean())
    AvgRHat.append(AvgMkt * TSR.params[1])
```

1.5 Empirical security market line

Once we have all the estimated betas, we can estimate the empirical security market line (SML) as implied by the historical data. To do so, we have to regress the average excess returns $\bar{R}_i$ of these portfolios on their respective estimated betas $\hat{\beta}_i$. This is also known as the cross-sectional regression (CSR):

$\bar{R}_i = \gamma + \hat{\beta}_i \lambda + e$

The estimated $\hat{\lambda}$ is the price of risk (risk premium) of the CAPM beta and it is the slope of the empirical SML.

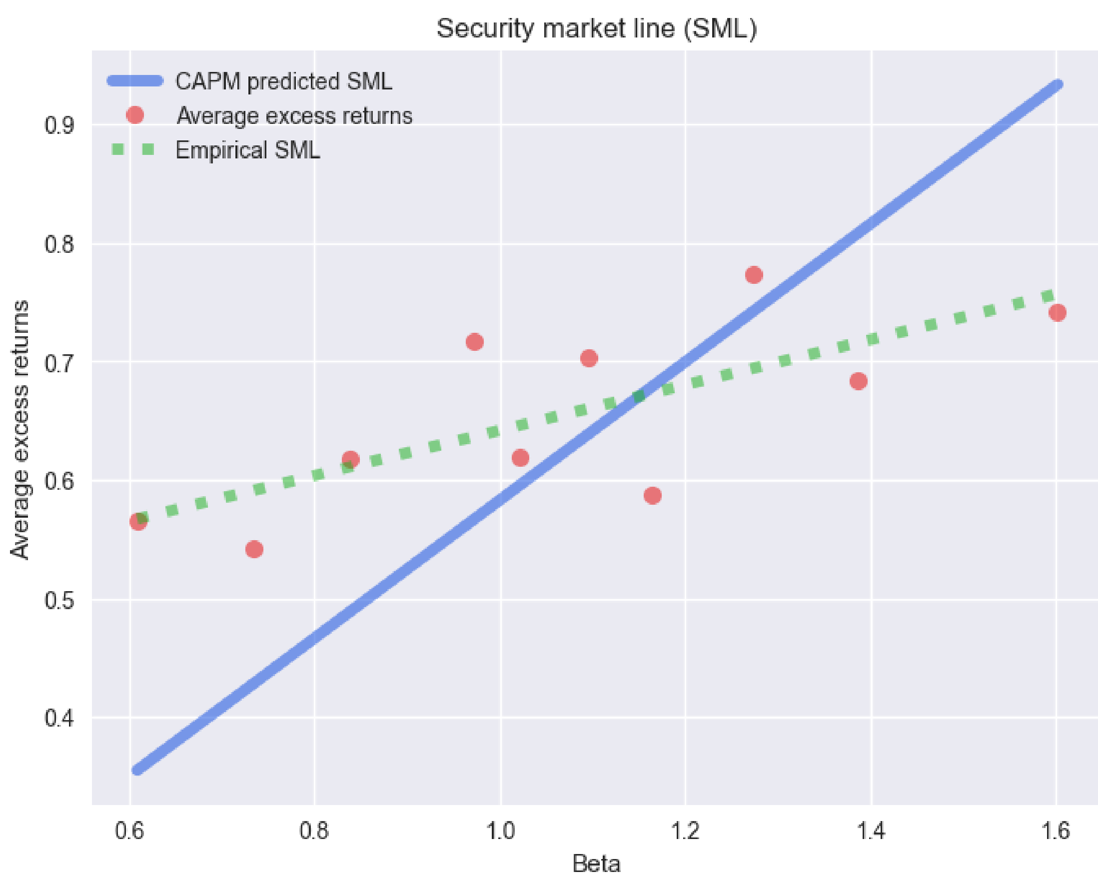
[7]:
```
CSR = sm.OLS(AvgR, sm.add_constant(BETA)).fit()
```

1.6 Compare the empirical SML with the CAPM SML

If the CAPM is true, the average excess returns of assets should equal their excess returns implied by their CAPM betas. However, the empirical SML is flatter than the CAPM predicted SML. In other words, low beta portfolios tend to have positive alphas while high beta portfolios tend to have negative alphas.

[8]:
```
import matplotlib.pyplot as plt

plt.style.use('seaborn')
fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(1,1,1)
plt.plot(BETA, AvgRHat, color='xkcd:blue', linewidth=5, alpha=0.5)
plt.plot(BETA, AvgR, 'o', color='xkcd:red', alpha=0.5, markersize=8)
plt.plot(BETA, CSR.fittedvalues, ':', color='xkcd:green', alpha=0.5, linewidth=5)
plt.ylabel('Average excess returns')
plt.xlabel('Beta')
plt.title('Security market line (SML)')
plt.legend(['CAPM predicted SML', 'Average excess returns', 'Empirical SML'],␣
↪loc='best')
plt.show()
```



References

[1] Eugene Fama and Kenneth French (2004) "The Capital Asset Pricing Model: Theory and Evidence", *Journal of Economic Perspectives*, 18(3): 25–46.