# Empirical Asset Pricing
# Assignment 01

Morteza Aghajanzadeh* Ge Song†

February 13, 2024

## Question 1

(a) Here are the moments and the correlation of the moments:

Table 1: Table of the moments

| | $\Delta c$ | $r_{m,t}$ | $r_{f,t}$ | $r_{e,t}$ |
|---|---|---|---|---|
| $\mu$ | 0.018 | 0.060 | 0.005 | 0.054 |
| $\sigma$ | 0.021 | 0.197 | 0.029 | 0.197 |
| $\rho_1$ | 0.504 | -0.010 | 0.676 | 0.019 |

Table 2: Table of the correlation of the moments

| | $\Delta c$ | $r_{m,t}$ | $r_{f,t}$ | $r_{e,t}$ |
|---|---|---|---|---|
| $\Delta c$ | 0.000 | 0.000 | -0.000 | 0.001 |
| $r_{m,t}$ | 0.000 | 0.039 | 0.000 | 0.038 |
| $r_{f,t}$ | -0.000 | 0.000 | 0.001 | -0.000 |
| $r_{e,t}$ | 0.001 | 0.038 | -0.000 | 0.039 |

(b) Given the moments and correlation that we calculate in the previous question, we can use the equation (2) in the question to estimate the parameters. The equation is as follows:

$$\mathbb{E}[r_{i,t} - r_{f,t}] + \frac{\sigma_i^2}{2} = \gamma \sigma_{ic}$$
$$\Rightarrow \gamma = \frac{\mathbb{E}[r_{i,t} - r_{f,t}] + \frac{\sigma_i^2}{2}}{\sigma_{ic}}$$

i. If we use the sample moments for calculating the parameters, we get that $\gamma_1 = 1.358$.

ii. If we assume that the correlation between excess returns on stocks and consumption growth equals one, we get that $\gamma_2 = 16.585$.

The outcomes differ, as expected, given our assumption of perfect correlation between stock returns and consumption growth in the second scenario. This assumption results in a significantly high value for the risk aversion parameter $\gamma$. The strong correlation implies that stock returns are highly responsive to changes in consumption growth, increasing their riskiness and consequently elevating the value of $\gamma$.

*Department of Finance, Stockholm School of Economics. Email: morteza.aghajanzadeh@phdstudent.hhs.se
†Department of Finance, Stockholm School of Economics. Email: ge.song@phdstudent.hhs.se

(c) Now we need to use the estimated parameters to estimate the time discount factor $\delta$. We can use the equation (3) in the question to estimate the time discount factor. The equation is as follows:

$$r_{f,t} = -\ln(\delta) + \gamma\mathbb{E}[\Delta c_t] - \frac{\gamma^2\sigma_c^2}{2}$$

as we write the equation for the average of the risk-free rate, we get:

$$\begin{aligned}\mathbb{E}[r_{f,t}] &= -\ln(\delta) + \gamma\mathbb{E}[\Delta c_t] - \frac{\gamma^2\sigma_c^2}{2} \\ \Rightarrow \delta &= \exp(-\mathbb{E}[r_{f,t}] + \gamma\mathbb{E}[\Delta c_t] - \frac{\gamma^2\sigma_c^2}{2})\end{aligned}$$

which for given moments and different values of $\gamma$ we get the following values for $\delta$:

i. Base on $\gamma_1$, we get that $\delta_1 = 1.019$ and time preference rate of $-0.019$.

ii. Base on $\gamma_2$, we get that $\delta_2 = 1.267$ and time preference rate of $-0.237$.

(d) Now we need to use the GMM estimator to estimate the parameters in order to have standard errors for the estimators. Let's define the variables as follows:

$$\begin{aligned}f(v_t,\theta) &= \begin{bmatrix} \Delta c_t - \mu_c \\ r_{m,t} - \mu_m \\ r_{m,t} - r_{f,t} + \frac{1}{2}(r_{m,t}-\mu_m)^2 - \gamma(r_{m,t}-\mu_m)(\Delta c_t - \mu_c) \\ r_{f,t} + \ln(\delta) - \gamma\Delta c_t + \frac{1}{2}\gamma^2(\Delta c_t - \mu_c)^2 \end{bmatrix} \quad , \quad \theta = \begin{bmatrix} \mu_c \\ \mu_m \\ \gamma \\ \delta \end{bmatrix} \\ g_T(\theta) &= \frac{1}{T}\sum_{t=1}^{T} f(v_t,\theta)\end{aligned}$$

As we can see, the system is exactly identified, since the number of parameters is equal to the number of moments. So, we can use the GMM estimator to estimate the parameters.

$$\begin{aligned} g_T(\theta) &= 0 \\ \Rightarrow \frac{1}{T}\sum_{t=1}^{T} f(v_t,\theta) &= 0 \\ \Rightarrow \frac{1}{T}\sum_{t=1}^{T}\begin{bmatrix} \Delta c_t - \mu_c \\ r_{m,t} - \mu_m \\ r_{m,t} - r_{f,t} + \frac{1}{2}(r_{m,t}-\mu_m)^2 - \gamma(r_{m,t}-\mu_m)(\Delta c_t - \mu_c) \\ r_{f,t} + \ln(\delta) - \gamma\Delta c_t + \frac{1}{2}\gamma^2(\Delta c_t - \mu_c)^2 \end{bmatrix} &= 0 \end{aligned}$$

$$\begin{aligned} \Rightarrow \begin{bmatrix} \frac{1}{T}\sum_{t=1}^{T} \Delta c_t - \mu_c \\ \frac{1}{T}\sum_{t=1}^{T} r_{m,t} - \mu_m \\ \frac{1}{T}\sum_{t=1}^{T} r_{m,t} - r_{f,t} + \frac{1}{2}(r_{m,t}-\mu_m)^2 - \gamma(r_{m,t}-\mu_m)(\Delta c_t - \mu_c) \\ \frac{1}{T}\sum_{t=1}^{T} r_{f,t} + \ln(\delta) - \gamma\Delta c_t + \frac{1}{2}\gamma^2(\Delta c_t - \mu_c)^2 \end{bmatrix} &= 0 \\ \Rightarrow \begin{bmatrix} \mathbb{E}[\Delta c_t] - \mu_c \\ \mathbb{E}[r_{m,t}] - \mu_m \\ \mathbb{E}[r_{m,t} - r_{f,t}] + \frac{1}{2}\mathbb{E}[(r_{m,t}-\mu_m)^2] - \gamma\mathbb{E}[(r_{m,t}-\mu_m)(\Delta c_t - \mu_c)] \\ \mathbb{E}[r_{f,t}] + \ln(\delta) - \gamma\mathbb{E}[\Delta c_t] + \frac{1}{2}\gamma^2\mathbb{E}[(\Delta c_t - \mu_c)^2] \end{bmatrix} &= 0 \\ \Rightarrow \begin{bmatrix} \mu_c - \mathbb{E}[\Delta c_t] \\ \mu_m - \mathbb{E}[r_{m,t}] \\ \gamma - \frac{\mathbb{E}[r_{m,t} - r_{f,t}] + \hat{\sigma}_m^2/2}{\hat{\sigma}_{mc}} \\ \delta - exp(-\mathbb{E}[r_{f,t}] + \gamma\mathbb{E}[\Delta c_t] - \frac{1}{2}\gamma^2\sigma_c^2) \end{bmatrix} &= 0 \end{aligned}$$

$$\rightarrow \theta = \begin{bmatrix} \mathbb{E}[\Delta c_t] \\ \mathbb{E}[r_{m,t}] \\ \frac{\mathbb{E}[r_{m,t}-r_{f,t}]+\hat{\sigma}_m^2/2}{\hat{\sigma}_{mc}} \\ exp(-\mathbb{E}[r_{f,t}] + \gamma\mathbb{E}[\Delta c_t] - \frac{1}{2}\gamma^2\sigma_c^2) \end{bmatrix}$$

where $\hat{\sigma}_m^2$ is the sample variance of $r_{m,t}$ and $\hat{\sigma}_{mc}$ is the sample covariance of $r_{m,t}$ and $\Delta c_t$.

As we can see it is the same method as we used for estimating in the first method of previous question. The only difference is that now we can estimate the variance of the estimator by using the equation for the variance of the GMM estimator. The Newey-West adjusted variance of the estimator is as follows:

$$\hat{S}_T = \frac{1}{T}\sum_{t=1}^{T} f(v_t,\hat{\theta})f(v_t,\hat{\theta}) + \frac{1}{2}(\hat{\Gamma}_1 + \hat{\Gamma}_1')$$

where $\hat{\theta}$ is the estimated parameter vector.

Calculate the variance of the estimator analytically is a bit more complicated, but it is possible. I will only drive the variance of the estimator numerically.

Table 3: Newey-West adjusted variance of the estimator

| | $\mu_c$ | $\mu_m$ | $\gamma$ | $\delta$ |
|---|---|---|---|---|
| $\mu_c$ | 0.0007 | 0.0013 | 0.0081 | -0.4310 |
| $\mu_m$ | 0.0013 | 0.0379 | 0.1080 | -0.9938 |
| $\gamma$ | 0.0081 | 0.1080 | 1.0622 | -7.9041 |
| $\delta$ | -0.4310 | -0.9938 | -7.9041 | 507.0663 |

(e) Now we change the target moments to be the following:

$$f(v_t,\theta) = \begin{bmatrix} \exp(\ln(\delta) - \gamma\Delta c_t + r_{m,t}) - 1 \\ \exp(\ln(\delta) - \gamma\Delta c_t + r_{f,t}) - 1 \end{bmatrix} \quad , \quad \theta = \begin{bmatrix} \gamma \\ \delta \end{bmatrix}$$

$$g_T(\theta) = \frac{1}{T}\sum_{t=1}^{T} f(v_t,\theta)$$

again the system is exactly identified, since the number of parameters is equal to the number of moments. So, we can use the GMM estimator to estimate the parameters.

$$\begin{aligned} g_T(\theta) &= 0 \\ \Rightarrow \frac{1}{T}\sum_{t=1}^{T} f(v_t,\theta) &= 0 \\ \Rightarrow \frac{1}{T}\sum_{t=1}^{T} \begin{bmatrix} \exp(\ln(\delta) - \gamma\Delta c_t + r_{m,t}) - 1 \\ \exp(\ln(\delta) - \gamma\Delta c_t + r_{f,t}) - 1 \end{bmatrix} &= 0 \\ \Rightarrow \begin{bmatrix} \frac{1}{T}\sum_{t=1}^{T} \exp(\ln(\delta) - \gamma\Delta c_t + r_{m,t}) - 1 \\ \frac{1}{T}\sum_{t=1}^{T} \exp(\ln(\delta) - \gamma\Delta c_t + r_{f,t}) - 1 \end{bmatrix} &= 0 \\ \Rightarrow \begin{bmatrix} \mathbb{E}[\exp(\ln(\delta) - \gamma\Delta c_t + r_{m,t})] - 1 \\ \mathbb{E}[\exp(\ln(\delta) - \gamma\Delta c_t + r_{f,t})] - 1 \end{bmatrix} &= 0 \end{aligned}$$

As our target moments are non-linear, we can not use the same method as we used for the linear moments. So we need to use the numerical optimization methods to estimate the parameters.

Here we use the FOC of the GMM estimator which is as follows:

$$g_T(\theta) = 0$$

and in the optimization I try to minimize the loss function which is quadratic one.

```python
def f_v(theta,x):
    gamma = theta[0]
    delta = theta[1]
    x_c = x[0]
    x_m = x[1]
    x_e = x[2]
    r_f = x_m - x_e
    f = np.array([
        np.exp(np.log(delta) - gamma * x_c + x_m) - 1,
        np.exp(np.log(delta) - gamma * x_c + r_f) - 1,
        ]).reshape(len(theta),1)
    return f

def FOC(theta,x):
    return sum([f_v(theta,i) for i in x])/len(x)

def loss(theta,x):
    return (FOC(theta,x).T @ FOC(theta,x))[0][0]
```

Listing 1: Python code for defining the moments and the loss function for the non-linear moments

Table 4: Estimation of the parameters for the non-linear moments

| | $\gamma$ | $\delta$ |
|---|---|---|
| $\hat{\theta}$ | 44.69 | 0.91 |

Table 5: Newey-West adjusted variance of the estimator for the non-linear moments

| | $\gamma$ | $\delta$ |
|---|---|---|
| $\gamma$ | 15.6636 | 16.3159 |
| $\delta$ | 16.3159 | 17.1583 |

## Question 2

(a)

## Question 3

I just wrote a function that create the portfolios and calculate the portfolio return based on "Equal" and "Market" weighting. The function is shown in the code 3. The function takes the following inputs:

- **df**: The dataframe that contains the data
- **sorting_car**: The variable that will be used to sort the stocks
- **number_of_portfolios**: The number of portfolios that will be created
- **weighting**: The type of weighting that will be used to calculate the portfolio return. The default is "Equal" weighting.

```python
def get_portfolios(df, sorting_car, number_of_portfolios,weighting = '
    Equal'):
portfoli_df = df.dropna(subset=[sorting_car])[
    ['t', 'permno', sorting_car, 'me', 'ret']
].copy()
portfoli_df['portfolios'] = portfoli_df.groupby('t')[sorting_car].
    transform(lambda x: pd.qcut(x, number_of_portfolios, labels=False))
portfoli_df['portfolios'] = portfoli_df['portfolios'] + 1    # The
    highest value is the highest portfolio
if weighting == 'market':
    portfoli_df['weight'] = portfoli_df.groupby(['t','portfolios'])['me'
        ].transform(lambda x: x/sum(x))
    portfoli_df['ret'] = portfoli_df['ret'] * portfoli_df['weight']
elif weighting == 'Equal':
    portfoli_df['weight'] = portfoli_df.groupby(['t','portfolios'])['me'
        ].transform(lambda x: 1/len(x))
    portfoli_df['ret'] = portfoli_df['ret'] * portfoli_df['weight']
return portfoli_df.groupby(['t','portfolios']).ret.sum().unstack().
    reset_index().rename(columns = {"t":"month"})
```

Listing 2: Python function to create portfolios

(a) Here is the result of the function for the "Equal" weighting. (Figure 1)

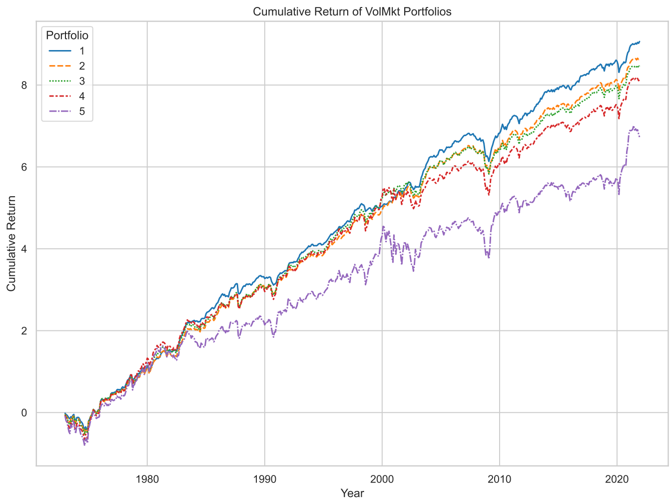


Figure 1: Time series of the average returns of the portfolios based on the "Equal" weighting.

(b) Here is the result of the function for the "Market" weighting. (Figure 2)

(c) Here we create the long-short portfolio. The long-short portfolio is created by taking the difference between the returns of the highest and the lowest portfolio. The result is shown in the figure 3.

Now we can test the CAPM, Fama-French 3 factors and the Fama-French 5 factors, Carhart, and HXZ models. You can find the function that I write to test the null hypothesis that $\alpha_{LS} = 0$. I will get the p-value of the test.

```python
def time_series_regression(portfolios, factors, FactorModel):
    portfolios = portfolios.merge(factors, on='month', how='left')
    portfolios = portfolios.dropna()
   X = portfolios[FactorModel]
   X = sm.add_constant(X)
```

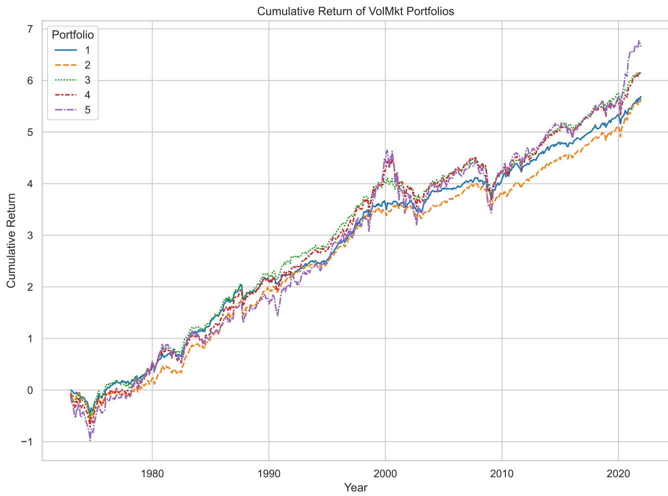


Figure 2: Time series of the average returns of the portfolios based on the "Market" weighting.

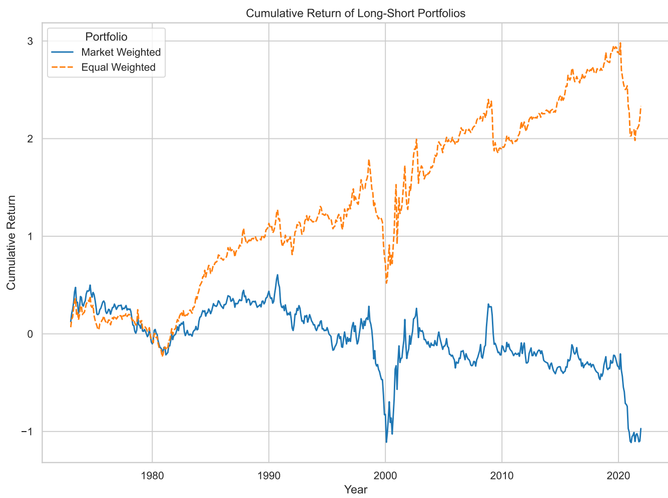


Figure 3: Time series of the average returns of the long-short portfolio.

```
    Y = portfolios['long_short']
    model = sm.OLS(Y, X).fit(cov_type='HAC',cov_kwds={'maxlags':int(len(Y)**0.25)})
    pvalues = model.pvalues
    betas = model.params
    return [betas.iloc[0],pvalues.iloc[0]]
```

Listing 3: Python function to run the test

Table 6: $\alpha$ test for long-short portfolio with different models

Equal Weighted

| | $\alpha$ | *Pvalue* |
|---|---|---|
| CAPM | 0.010 | 0.000 |
| FF3 | 0.008 | 0.000 |
| CAR | 0.003 | 0.167 |
| FF5 | 0.003 | 0.167 |
| HXZ | 0.000 | 0.944 |

Market Weighted

| | $\alpha$ | *Pvalue* |
|---|---|---|
| CAPM | 0.004 | 0.073 |
| FF3 | 0.002 | 0.267 |
| CAR | -0.000 | 0.790 |
| FF5 | -0.003 | 0.061 |
| HXZ | -0.004 | 0.036 |

Table 7: $\alpha$ test long-short portfolio for in and out of sample with equal weighting

Sample period

| | $\alpha$ | *Pvalue* |
|---|---|---|
| CAPM | 0.010 | 0.000 |
| FF3 | 0.008 | 0.000 |
| CAR | 0.006 | 0.008 |
| FF5 | 0.005 | 0.016 |
| HXZ | 0.005 | 0.081 |

Post-publication period

| | $\alpha$ | *Pvalue* |
|---|---|---|
| CAPM | 0.012 | 0.002 |
| FF3 | 0.011 | 0.000 |
| CAR | 0.007 | 0.027 |
| FF5 | 0.005 | 0.132 |
| HXZ | 0.001 | 0.777 |

Table 8: $\alpha$ test long-short portfolio for in and out of sample with market weighting

Sample period

| | $\alpha$ | *Pvalue* |
|---|---|---|
| CAPM | 0.004 | 0.089 |
| FF3 | 0.002 | 0.346 |
| CAR | 0.000 | 0.911 |
| FF5 | -0.000 | 0.896 |
| HXZ | -0.001 | 0.821 |

Post-publication period

| | $\alpha$ | *Pvalue* |
|---|---|---|
| CAPM | 0.005 | 0.115 |
| FF3 | 0.004 | 0.056 |
| CAR | 0.002 | 0.286 |
| FF5 | -0.002 | 0.389 |
| HXZ | -0.003 | 0.198 |

(d)