In a market consisting of N stocks, we denote the dividend-adjusted return on stock i at trading day t by $r_{i,t}$. We adopt a factor model for stock return,

$$r_t - r_f = \beta_t F_t + \epsilon_t, \quad t = 1, 2, \ldots, T \tag{1}$$

Here, $r_t = \{r_{i,t}\}_{i=1}^N \in \mathbb{R}^N$ are the dividend-adjusted daily return, $r_f \in \mathbb{R}$ is the risk-free rate, $F_t \in \mathbb{R}^{K\times 1}$ are the underlying factors, $\beta_t \in \mathbb{R}^{N\times K}$ are the corresponding loadings on K factors, and $\epsilon_t \in \mathbb{R}^N$ are the residual returns. Factor candidates varies widely, ranging from economical-driven factors such as the Fama-French factors, to statistically-driven factors derived from PCA. In our approach, factors are selected as the leading eigenvectors in PCA. The number of factors K is chosen based on the eigenvalue spectrum of the empirical correlation of daily returns.

Without loss of generality, these factors can be interpreted as portfolios of stocks,

$$F_t = \omega_t \left(r_t - r_f\right) \tag{2}$$

where $\omega_t \in \mathbb{R}^{K\times N}$ contains corresponding portfolio weights. Combining eq. (1) and eq. (2) yields

$$r_t - r_f = \beta_t \omega_t \left(r_t - r_f\right) + \epsilon_t \Rightarrow \epsilon_t = \left(I - \beta_t \omega_t\right)\left(r_t - r_f\right) := \Phi_t \left(r_t - r_f\right) \tag{3}$$

Here,

$$\Phi_t := \left(I - \beta_t \omega_t\right) \tag{4}$$

defines a linear transformation from r_t to ϵ_t. More importantly, $\epsilon_{i,t}$ can be viewed as the return of a tradable portfolio with weights specified by the i-th row of Φ_t. Consequently, the investing universe spanned by r_t is termed as name equity space, and that spanned by ϵ_t as name residual space.

We denote the portfolio weights in name equity space as $w_t^{R,\text{ name}}$ and portfolio weights in name residual space as $w_t^{\epsilon,\text{ name}}$. These weights are related by

$$w_t^{R,\text{ name}} = \Phi_t^T w_t^{\epsilon,\text{ name}} \tag{5}$$

, directly following the equality in portfolio return,

$$\left(w_t^{\epsilon\text{ name}}\right)^T \epsilon_t = \left(w_t^{\epsilon,\text{ name}}\right)^T \Phi_t \left(r_t - r_f\right) = \left(w_t^{R,\text{ name}}\right)^T \left(r_t - r_f\right) \tag{6}$$

For factors derived by PCA, we have

$$\Phi_t \beta_t = 0 \Longrightarrow \left(w_t^{R,\text{ name}}\right)^T \beta_t = \left(w_t^{\epsilon,\text{ name}}\right)^T \Phi_t \beta_t = 0, \quad \forall w_t^{\epsilon,\text{ name}} \tag{7}$$

with proof given in the appendix. It means that for any $w_t^{\epsilon,\text{name}}$, the $w_t^{R,\text{name}}$ calculated by eq. (5) satisfy,

$$\left(w_t^{R,\text{ name}}\right)^T (r_t - r_f) = (w_t^{\epsilon,\text{ name}})^T \Phi_t (\beta_t F_t + \epsilon_t) = (w_t^{\epsilon,\text{ name}})^T \Phi_t \epsilon_t = \left(w_t^{R,\text{ name}}\right)^T \epsilon_t \tag{8}$$

It suggests that the return of our statistical arbitrage portfolios is independent of market factors and relies solely on residual returns, a property usually termed as market neutrality. Ideally, portfolios are also desired to have a zero net value, known as dollar neutrality. Empirical evidence suggests that market-neutral portfolios are also approximately dollar-neutral.

Algorithm 1. Market decomposition (PCA) [Fig. 5, panel (c1, c2)]

Require: r_t, $r_{f,t}$, K
Ensure: ϵ_t, Φ_t

```
function MARKET_DECOMPOSITION(r_t, r_{f,t}, K)
    Perform principal component analysis: r_t − r_{f,t} = UΣV^T
    F_t ← (v_1, v_2, ..., v_K), where v_k is the k-th column of V^T
    Calculate ω_t by solving F_t = ω_t(r_t − r_{f,t})
    Calculate β_t as the coefficient of the linear regression r_t − r_f ∼ F_t
    Φ_t ← I − β_t ω_t
    ε_t ← Φ_t(r_t − r_{f,t})
    return ε_t, Φ_t
end function
```

Input:

- r_t: return in name space or transformed return in rank space.
- $r_{f,t}$: risk-free rate at the end of trading day t.
- K: number of market factors, predetermined by analyzing eigenvalue spectrum of the correlation matrix.

Output:

- ϵ_t: residual returns in name space or rank space.
- Φ_t: transformation between residual space and equity space (Eq. 2.1.1 for name space and Eq. 2.1.10 for rank space).

Note:

- The algorithm realizes the formulation in section 2.1.

- Factors F_t and ω_t are calculated on a 252-day look-back window.
- Loadings β_t are calculated on a 60-day look-back window.
- F_t, ω_t, and β_t are updated daily.
- $K = 5$ for name space and $K = 1$ for rank space based on empirical eigenvalue spectrum of the correlation matrix (Fig. 6(c,d)).

[**Appendix**]: Here, we prove the equality $\Phi_t \beta_t = 0$, crucial relationship for market neutrality. We denote the return matrix $R_t = (r_{t-T+1}, r_{t-T+2}, \dots, r_t) \in \mathbb{R}^{N\times T}$, (where T is a window of 252 days). Assume singular value decomposition of R_t,

$$R_t - R_t^f = U\Sigma V^T$$

where $R_t^f \in \mathbb{R}^{1\times T}$ is the risk-free rate, $U \in \mathbb{R}^{N\times N}, \Sigma \in \mathbb{R}^{N\times T}$, and $V^T \in \mathbb{R}^{T\times T}$. Then, the factors and loadings in Eq. 2.1.1 and ω_t in Eq. 2.1.2 becomes

$$F_t = \begin{pmatrix} v_1^T \\ v_2^T \\ \dots \\ v_K^T \end{pmatrix}, \quad \beta_t = (u_1, u_2, \dots, u_K) \begin{pmatrix} \sigma_1 & & \\ & \dots & \\ & & \sigma_K \end{pmatrix}, \quad \omega_t = \begin{pmatrix} \sigma_1^{-1} & & \\ & \dots & \\ & \dots & \sigma_K^{-1} \end{pmatrix} \begin{pmatrix} u_1^T \\ u_2^T \\ \dots \\ \\ u_K^T \end{pmatrix}$$

where u_i and v_i are the i-th column of matrix U and V. Then, because U and V are orthogonal matrix,

$$\beta_t \omega_t = I$$
$$\implies \Phi_t \beta_t = (I - \beta_t \omega_t)\, \beta_t = 0$$

$$\begin{aligned}
\underbrace{F_t}_{K\times T} &= \underbrace{V_K}_{K\times T} \\
\underbrace{w_t}_{K\times N} &= \underbrace{\Sigma_K^+}_{K\times K}\underbrace{U_K^T}_{K\times N} = \underbrace{F_t}_{K\times T}\underbrace{R_t^+}_{T\times N} \\
\underbrace{\beta_t}_{N\times K} &= \underbrace{U_K}_{N\times K}\underbrace{\Sigma_K}_{K\times K} = \underbrace{(F_t^T F_t)^+}_{K\times K}\underbrace{F_t}_{K\times T}\underbrace{r_t}_{T\times 1}
\end{aligned}$$

Potential Typo: If $\beta_t\omega_t = I$, then $\Phi_t := (I - \beta_t\omega_t) = I - I = 0$, which doesn't make sense.

Note that:

$$
\begin{array}{lllll}
\beta_t & = U_K\Sigma_K & & & \\
\omega_t & = \Sigma_K^{-1}U_K^\top & & & \\
\beta_t\omega_t & = U_K\Sigma_K\Sigma_K^{-1}U_K^\top & = U_KU_K^\top & = \sum_{i=1}^K \vec{u}_i\vec{u}_i^\top & \\
(\beta_t\omega_t)\beta_t & = (U_KU_K^\top)U_K\Sigma_K & = U_K\Sigma_K & & \\
\Phi_t & = (I - \beta_t\omega_t) & = I - U_KU_K^\top & = I_N - (I_N - \sum_{i=K+1}^N \vec{u}_i\vec{u}_i^\top) & = \sum_{i=K+1}^N \vec{u}_i\vec{u}_i^\top \\
\Phi_t\beta_t & = (I - \beta_t\omega_t)\beta_t & = \beta_t - \beta_t\omega_t\beta_t & = \beta_t - \beta_t & = 0
\end{array}
$$

- Since U and V are orthogonal matrices from the Singular Value Decomposition (SVD), they satisfy $U^TU = I \in \mathbb{R}^{N\times N}$ and $V^TV = I \in \mathbb{R}^{T\times T}$.
- $U \in \mathbb{R}^{N\times N}$ is an orthogonal matrix $\implies U^\top U = UU^\top = I_N$, where I_N is the $N\times N$ identity matrix. Hence, the columns (and rows) of U are orthonormal vectors in $\mathbb{R}^N$.
  - $U^\top U = I_N$ can be seen from the inner product of the orthonormal vectors $\vec{u}_i$.

$$
U^\top U = \begin{bmatrix} \vec{u}_1^\top \\ \vec{u}_2^\top \\ \vdots \\ \vec{u}_N^\top \end{bmatrix} \begin{bmatrix} \vec{u}_1 & \vec{u}_2 & \dots & \vec{u}_N \end{bmatrix} = \begin{bmatrix} \vec{u}_1^\top\vec{u}_1 & \vec{u}_1^\top\vec{u}_2 & \dots & \vec{u}_1^\top\vec{u}_N \\ \vec{u}_2^\top\vec{u}_1 & \vec{u}_2^\top\vec{u}_2 & \dots & \vec{u}_2^\top\vec{u}_N \\ \vdots & \vdots & \ddots & \vdots \\ \vec{u}_N^\top\vec{u}_1 & \vec{u}_N^\top\vec{u}_2 & \dots & \vec{u}_N^\top\vec{u}_N \end{bmatrix} = I_N
$$

  - $UU^\top = I_N$ can be seen from the outer product of the orthonormal vectors $\vec{u}_i\vec{u}_i^\top \in \mathbb{R}^N$

$$
UU^\top = \begin{bmatrix} \vec{u}_1 & \vec{u}_2 & \dots & \vec{u}_N \end{bmatrix} \begin{bmatrix} \vec{u}_1^\top \\ \vec{u}_2^\top \\ \vdots \\ \vec{u}_N^\top \end{bmatrix} = \sum_{i=1}^N \vec{u}_i\vec{u}_i^\top = I_N
$$

- The matrix $U_K \in \mathbb{R}^{N\times K}$ is formed by taking the first K columns of U $(K \leq N)$

$$
U_K = \begin{bmatrix} \vec{u}_1 & \vec{u}_2 & \dots & \vec{u}_K \end{bmatrix},
$$

  where $\vec{u}_i \in \mathbb{R}^N$ are the orthonormal columns of U. This means:

$$
\vec{u}_i^\top\vec{u}_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}
$$

- **Computing $U_K^\top U_K$:**

  $U_K^\top \in \mathbb{R}^{K\times N}, U_K \in \mathbb{R}^{N\times K} \implies U_K^\top U_K \in \mathbb{R}^{K\times K}$. Computation:

$$U_K^\top U_K = \begin{bmatrix} \vec{u}_1^\top \\ \vec{u}_2^\top \\ \vdots \\ \vec{u}_K^\top \end{bmatrix} \begin{bmatrix} \vec{u}_1 & \vec{u}_2 & \dots & \vec{u}_K \end{bmatrix} = \begin{bmatrix} \vec{u}_1^\top\vec{u}_1 & \vec{u}_1^\top\vec{u}_2 & \dots & \vec{u}_1^\top\vec{u}_K \\ \vec{u}_2^\top\vec{u}_1 & \vec{u}_2^\top\vec{u}_2 & \dots & \vec{u}_2^\top\vec{u}_K \\ \vdots & \vdots & \ddots & \vdots \\ \vec{u}_K^\top\vec{u}_1 & \vec{u}_K^\top\vec{u}_2 & \dots & \vec{u}_K^\top\vec{u}_K \end{bmatrix} = I_K$$

  since the vectors $\vec{u}_i$ are orthonormal:

- **Computing $U_K U_K^\top$**

  $U_K \in \mathbb{R}^{N\times K}, U_K^\top \in \mathbb{R}^{K\times N} \implies U_K U_K^\top \in \mathbb{R}^{N\times N}$. Computation:

$$U_K U_K^\top = \begin{bmatrix} \vec{u}_1 & \vec{u}_2 & \dots & \vec{u}_K \end{bmatrix} \begin{bmatrix} \vec{u}_1^\top \\ \vec{u}_2^\top \\ \vdots \\ \vec{u}_K^\top \end{bmatrix} = \sum_{i=1}^{K} \vec{u}_i \vec{u}_i^\top$$

  Hence, $U_K U_K^\top$ is not the identity matrix I_N unless $K = N$. Instead, $U_K U_K^\top$ is a projection matrix onto the column space of U_K.

  - **Incomplete Basis**: The set $\{\vec{u}_1, \vec{u}_2, \dots, \vec{u}_K\}$ spans a K-dimensional subspace $\mathcal{S}$ of $\mathbb{R}^N$. It does not form a complete basis for $\mathbb{R}^N$ when $K < N$.
  - **Projection Operator**: $U_K U_K^\top$ is a projection matrix onto the subspace $\mathcal{S}$. This projection does not recover $\vec{x}$ unless $\vec{x} \in \mathcal{S}$. For any $\vec{x} \in \mathbb{R}^N$, the projection onto $\mathcal{S}$ is:

$$U_K U_K^\top \vec{x} = \sum_{i=1}^{K} \vec{u}_i \left(\vec{u}_i^\top \vec{x} \right)$$

  - Note that, since $\sum_{i=1}^{N} \vec{u}_i\vec{u}_i^\top = (\sum_{i=1}^{K} \vec{u}_i\vec{u}_i^\top) + (\sum_{i=K+1}^{N} \vec{u}_i\vec{u}_i^\top) = I_N$, then

$$\sum_{i=1}^{K} \vec{u}_i\vec{u}_i^\top = I_N - \sum_{i=K+1}^{N} \vec{u}_i\vec{u}_i^\top$$

- **Question**: Given: $(\beta_t\omega_t)\,\beta_t = \left(U_K U_K^\top\right) U_K \Sigma_K = U_K \Sigma_K$, does this imply that $U_K U_K^\top = I_N$?

  Answer: No, it does not imply that $U_K U_K^\top = I_N$. The equation shows that $U_K U_K^\top$ acts as an identity only on the vectors in the column space of $U_K \Sigma_K$.

  Key Points:

1. $U_K U_K^\top$ is a Projection Matrix: It satisfies $\left(U_K U_K^\top\right)^2 = U_K U_K^\top$. It projects any vector onto the column space of U_K.
2. Acting on $U_K \Sigma_K$: Since $U_K \Sigma_K \in \mathrm{Col}(U_K)$, projecting it onto the column space throught the projection matrix $U_K U_K^\top$ leaves it unchanged: $(U_K U_K^\top)(U_K \Sigma_K) = U_K \Sigma_K$
3. Not Acting as Identity on Entire $\mathbb{R}^N$: For $\vec{v} \notin \mathrm{Col}(U_K)$, the projection matrix $U_K U_K^\top$ does not act as the identity. *Example: Let $\vec{v}$ be orthogonal to the column space of U_K :*

$$U_K^\top \vec{v} = 0$$

Then:

$$U_K U_K^\top \vec{v} = U_K \left(U_K^\top \vec{v}\right) = U_K(0) = 0 \neq \vec{v}$$

This shows that $U_K U_K^\top$ does not act as the identity on $\vec{v}$.

Further Explanation with Mathematical Details

- **Rank considerations**. U_K has dimensions $N \times K$. Then, $U_K^\top U_K = I_K$ because the columns of U_K are orthonormal. However, $U_K U_K^\top$ is an $N \times N$ matrix, and $\mathrm{rank}(U_K U_K^\top) = K$, since it's the product of an $N \times K$ matrix and a $K \times N$ matrix. Since the Identity Matrix I_N has rank N and $\mathrm{rank}(U_K U_K^\top) = K < N$, it's clear that $U_K U_K^\top$ cannot be equal to I_N, as a matrix of rank K cannot equal a matrix of rank N.
- **Projection onto Col(U_K) considerations**. Any vector $\vec{x}$ in $\mathbb{R}^N$ can be decomposed as $\vec{x} = \vec{x}_\parallel + \vec{x}_\perp$, where $\vec{x}_\parallel$ is the component in the column space of U_K, and $\vec{x}_\perp$ is the component orthogonal to the column space of U_K. Applying $U_K U_K^\top$ to $\vec{x}$:

$$U_K U_K^\top \vec{x} = U_K U_K^\top (\vec{x}_\parallel + \vec{x}_\perp) = U_K U_K^\top \vec{x}_\parallel + U_K U_K^\top \vec{x}_\perp = \vec{x}_\parallel$$

because, since $U_K U_K^\top$ projects onto the column space, $U_K U_K^\top \vec{x}_\parallel = \vec{x}_\parallel$ and $U_K U_K^\top \vec{x}_\perp = \overrightarrow{0}$. Therefore, $U_K U_K^\top \vec{x} = \vec{x}_\parallel$. In general, unless there is no component in $\vec{x}$ that is orthogonal to $\mathrm{Col}(U_K)$; that is, if $\vec{x}_\perp = \overrightarrow{0}$ (i.e: $\vec{x}$ is in the column space), $U_K U_K^\top \vec{x} \neq \vec{x}$.

Why is $\omega_t = \Sigma_K^{-1} U_K^\top$ implied by Algorithm 1?

According to Algorithm 1., ω_t can be computed by solving $F_t = \omega_t(r_t - r_{f,t})$.

- **Truncated SVD with Top K Components**. We can approximate $R_t - R_t^f$ using the top K components:

$$R_t - R_t^f \approx U_K \Sigma_K V_K^\top$$

where $U_K \in \mathbb{R}^{N \times K}$ are the first K columns of U, $\Sigma_K \in \mathbb{R}^{K \times K}$ are the top K singular values, and $V_K^\top \in \mathbb{R}^{K \times T}$ are the first K rows of $V^\top$. Define $F_t = V_K^\top \in \mathbb{R}^{K \times T}$.

$$
\begin{aligned}
\dot{R}_t &= U\Sigma V^\top &\approx U_K \Sigma_K V_K^\top \\
F_t &:= V_K &= \omega_t \dot{R}_t &\approx \omega_t U_K \Sigma_K V_K^\top
\end{aligned}
$$

$$
\begin{aligned}
F_t &\approx \omega_t U_K \Sigma_K V_K^\top \\
V_K^T &\approx \omega_t U_K \Sigma_K V_K^\top \\
V_K^T V_K &\approx \omega_t U_K \Sigma_K V_K^\top V_K \\
I &\approx \omega_t U_K \Sigma_K \\
\omega_t &\approx (U_K \Sigma_K)^{-1} \\
&\approx \Sigma_K^{-1} U_K^\top
\end{aligned}
$$

My understanding of the algorithm:
for t in *timeline*:

- $\dot{\mathbf{r}}_{t-w_{pca}+1:t} - \text{TSmean}(\dot{\mathbf{r}}_{t-w_{pca}+1:t}) = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top \in \mathbb{R}^{w_{pca}\times N}$ focus on $\mathbf{V} \in \mathbb{R}^{N\times N}$
  - $\text{TSmean}(\dot{\mathbf{r}}^{(i)}_{t-w_{pca}+1:t}) = \frac{1}{w_{pca}} \sum_{s=t-w_{pca}+1}^{t} \dot{r}^{(i)}_s$ for i in *firms*
- $\boldsymbol{\omega}_t = \mathbf{V}_{[1:K,:]} \in \mathbb{R}^{K\times N}$
- $\mathbf{F}_t = \boldsymbol{\omega}_t[\dot{\mathbf{r}}_t - \text{mean}(\dot{\mathbf{r}}_{t-w_{pca}+1:t})] \in \mathbb{R}^{K\times 1}$
- Solution: Use $\text{TSmean}(\dot{\mathbf{r}}_{t-w_{pca}+1:t})$ to center the returns in the regression
  - $[\dot{\mathbf{r}}^{(i)}_{t-w_{reg}+1:t} - \text{TSmean}(\dot{\mathbf{r}}^{(i)}_{t-w_{pca}+1:t})] = \boldsymbol{\beta}^{(i)}_t \mathbf{F}_t + \boldsymbol{\epsilon}^{(i)}_{t-w_{reg}+1:t} \in \mathbb{R}^{w_{reg}\times 1}$ for each i in *firms*
  - Extract the last TS element: $[\dot{\mathbf{r}}^{(i)}_t - \text{TSmean}(\dot{\mathbf{r}}^{(i)}_{t-w_{pca}+1:t})] = \boldsymbol{\beta}^{(i)}_t \mathbf{F}_t + \boldsymbol{\epsilon}^{(i)}_t \in \mathbb{R}$
  - Stack them XS: $[\dot{\mathbf{r}}_t - \text{TSmean}(\dot{\mathbf{r}}_{t-w_{pca}+1:t})] = \boldsymbol{\beta}_t \mathbf{F}_t + \boldsymbol{\epsilon}_t \in \mathbb{R}^{N\times 1}$

    Hence, now:

$$
\begin{aligned}
\boldsymbol{\epsilon}_t &= [\dot{\mathbf{r}}_t - \text{TSmean}(\dot{\mathbf{r}}_{t-w_{pca}+1:t})] - \boldsymbol{\beta}_t \mathbf{F}_t \\
&= [\dot{\mathbf{r}}_t - \text{TSmean}(\dot{\mathbf{r}}_{t-w_{pca}+1:t})] - \boldsymbol{\beta}_t \boldsymbol{\omega}_t [\dot{\mathbf{r}}_t - \text{mean}(\dot{\mathbf{r}}_{t-w_{pca}+1:t})] \\
&= (\mathbf{I} - \boldsymbol{\beta}_t \boldsymbol{\omega}_t)[\dot{\mathbf{r}}_t - \text{TSmean}(\dot{\mathbf{r}}_{t-w_{pca}+1:t})] \\
&= \boldsymbol{\Phi}_t [\dot{\mathbf{r}}_t - \text{TSmean}(\dot{\mathbf{r}}_{t-w_{pca}+1:t})]
\end{aligned}
$$

Problem: Inconsistent return centering $\implies$ we cannot compute $\boldsymbol{\Phi}_t$

- $[\dot{\mathbf{r}}^{(i)}_{t-w_{reg}+1:t} - \text{TSmean}(\dot{\mathbf{r}}^{(i)}_{t-w_{reg}+1:t})] = \boldsymbol{\beta}^{(i)}_t \mathbf{F}_t + \boldsymbol{\epsilon}^{(i)}_{t-w_{reg}+1:t} \in \mathbb{R}^{w_{reg}\times 1}$ for each i in *firms*

- Extract the last TS element: $[\dot{\mathbf{r}}_t^{(i)} - \text{TSmean}(\dot{\mathbf{r}}^{(i)}_{t-w_{reg}+1:t})] = \boldsymbol{\beta}_t^{(i)}\mathbf{F}_t + \boldsymbol{\epsilon}_t^{(i)} \in \mathbb{R}$
- Stack them XS: $[\dot{\mathbf{r}}_t - \text{TSmean}(\dot{\mathbf{r}}_{t-w_{reg}+1:t})] = \boldsymbol{\beta}_t\mathbf{F}_t + \boldsymbol{\epsilon}_t \in \mathbb{R}^{N\times 1}$

$$
\begin{aligned}
\boldsymbol{\epsilon}_t &= [\dot{\mathbf{r}}_t - \text{TSmean}(\dot{\mathbf{r}}_{t-w_{reg}+1:t})] - \boldsymbol{\beta}_t\mathbf{F}_t \\
&= [\dot{\mathbf{r}}_t - \text{TSmean}(\dot{\mathbf{r}}_{t-w_{reg}+1:t})] - \boldsymbol{\beta}_t\boldsymbol{\omega}_t[\dot{\mathbf{r}}_t - \text{mean}(\dot{\mathbf{r}}_{t-w_{pca}+1:t})] \\
\\
< \text{if } w = w_{pca} = w_{reg} > \quad &= [\dot{\mathbf{r}}_t - \text{TSmean}(\dot{\mathbf{r}}_{t-w+1:t})] - \boldsymbol{\beta}_t\boldsymbol{\omega}_t[\dot{\mathbf{r}}_t - \text{mean}(\dot{\mathbf{r}}_{t-w+1:t})] \\
&= (\mathbf{I} - \boldsymbol{\beta}_t\boldsymbol{\omega}_t)[\dot{\mathbf{r}}_t - \text{TSmean}(\dot{\mathbf{r}}_{t-w+1:t})] \\
&= \boldsymbol{\Phi}_t[\dot{\mathbf{r}}_t - \text{TSmean}(\dot{\mathbf{r}}_{t-w+1:t})]
\end{aligned}
$$

0.1. Fitting an Ornstein-Uhlenbeck (OU) Process

The objective is to fit an Ornstein-Uhlenbeck (OU) process to the cumulative residual returns x_t^L for each asset. The OU process is a continuous-time stochastic process exhibiting mean-reverting behavior, suitable for modeling financial time series that drift toward a long-term mean.

Definition of the Ornstein-Uhlenbeck Process

The OU process is governed by the stochastic differential equation (SDE):

$$dX_t = \frac{1}{\tau}(\mu - X_t)dt + \sigma dB_t \tag{9}$$

where:

- X_t is the state variable at time t.
- μ is the long-term mean toward which the process reverts.
- τ is the mean-reversion time (the speed of reversion).
- σ is the volatility parameter.
- dB_t is the increment of a standard Brownian motion.

Discretization of the OU Process

To fit this continuous-time process to discrete data, we discretize the SDE using the Euler-Maruyama method with a time step $\Delta t = 1$:

$$X_{t+1} = X_t + \frac{1}{\tau}(\mu - X_t)\Delta t + \sigma\epsilon_t \tag{10}$$

where $\epsilon_t \sim \mathcal{N}(0, \Delta t)$ is a standard normal random variable.
Simplifying with $\Delta t = 1$:

$$X_{t+1} = X_t + \frac{1}{\tau}(\mu - X_t) + \sigma\epsilon_t \tag{11}$$

Rewriting the Equation

Rewriting the equation:

$$X_{t+1} = \left(1 - \frac{1}{\tau}\right) X_t + \frac{\mu}{\tau} + \sigma \epsilon_t \tag{12}$$

Define:

$$a = 1 - \frac{1}{\tau} \tag{13}$$

$$b = \frac{\mu}{\tau} \tag{14}$$

The equation becomes:

$$X_{t+1} = aX_t + b + \sigma \epsilon_t \tag{15}$$

This is a first-order autoregressive (AR(1)) process with an intercept term.

Estimating Parameters Using Linear Regression

We estimate the parameters a and b by performing linear regression on X_t and X_{t+1}:

$$X_{t+1} = aX_t + b + \text{noise} \tag{16}$$

Linear Regression Model

We set up the linear regression model:

$$y_t^{(i)} = a_i x_t^{(i)} + b_i + \epsilon_t^{(i)} \tag{17}$$

where:

- $y_t^{(i)} = x_{\text{curr}}^{(i)}$
- $x_t^{(i)} = x_{\text{lag}}^{(i)}$

Estimating Coefficients

We use the Ordinary Least Squares (OLS) method to estimate a_i and b_i by minimizing the sum of squared residuals:

$$\min_{a_i,b_i} \sum_t \left(y_t^{(i)} - a_i x_t^{(i)} - b_i\right)^2 \tag{18}$$

This can be solved analytically using the normal equations:

$$\beta^{(i)} = \left(X^{(i)\top} X^{(i)}\right)^{-1} X^{(i)\top} y^{(i)} \tag{19}$$

where:

- $y^{(i)} = [x_0^{(i)}, x_1^{(i)}, ..., x_{T-1}^{(i)}]^\top$
- $X^{(i)} = [\mathbf{1}_{T-1}, x^{(i)}]$
- $x^{(i)} = [x_1^{(i)}, x_2^{(i)}, ..., x_T^{(i)}]^\top$
- $\beta^{(i)} = \begin{bmatrix} a_i \\ b_i \end{bmatrix}$
- $X^{(i)}$ is the design matrix for asset i, including an intercept term.

Computing Residuals and Estimating Volatility

The residuals are computed as:

$$\epsilon_t^{(i)} = y_t^{(i)} - (a_i x_t^{(i)} + b_i) \tag{20}$$

The volatility σ_i is estimated as the standard deviation of the residuals:

$$\sigma_i = \sqrt{\frac{1}{T-2} \sum_{t=1}^{T-1} \left(\epsilon_t^{(i)}\right)^2} \tag{21}$$

Recovering OU Process Parameters

From the estimated a_i and b_i, we recover τ_i and μ_i.

Mean-Reversion Time

$$\tau_i = -\frac{1}{\ln(a_i)} \tag{22}$$

Long-Term Mean

$$\mu_i = \frac{b_i}{1 - a_i} \tag{23}$$

Constraints on Parameter Estimates

Why $a_i \leq 0$ and $a_i \geq 1$ Are Invalid

For the OU process:

- The mean-reversion time $\tau > 0$, implying $a_i = 1 - \frac{1}{\tau} < 1$.
- If $a_i \geq 1$, $\ln(a_i) \geq 0$, leading to $\tau_i \leq 0$, which is not meaningful.
- If $a_i \leq 0$, $\ln(a_i)$ is undefined (complex), and a_i implies explosive or oscillatory behavior, inconsistent with the OU process.

Therefore, valid estimates require $0 < a_i < 1$.

Maximum Likelihood Estimation and Log-Likelihood Optimization

Under the assumption of normally distributed residuals $\epsilon_t^{(i)}$, OLS estimation of the AR(1) model is equivalent to maximizing the log-likelihood function.

Log-Likelihood Function

The likelihood function for the AR(1) model is:

$$L(a_i, b_i, \sigma_i) = \prod_{t=1}^{T-1} \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{\left(y_t^{(i)} - a_i x_t^{(i)} - b_i\right)^2}{2\sigma_i^2}\right) \tag{24}$$

Taking the natural logarithm:

$$\ln L(a_i, b_i, \sigma_i) = -\frac{(T-1)}{2}\ln(2\pi\sigma_i^2) - \frac{1}{2\sigma_i^2}\sum_{t=1}^{T-1}\left(y_t^{(i)} - a_i x_t^{(i)} - b_i\right)^2 \tag{25}$$

Maximizing the log-likelihood with respect to a_i and b_i is equivalent to minimizing the sum of squared residuals, which is what OLS achieves.

Summary of Steps

1. **Data Preparation:** Extract lagged and current values for each asset.
2. **Linear Regression:** Set up the regression model $y = ax + b + \epsilon$.
3. **Estimate Coefficients:** Use OLS to estimate a_i and b_i.
4. **Compute Residuals:** Calculate $\epsilon_t^{(i)}$ and estimate σ_i.
5. **Recover OU Parameters:** Compute τ_i and μ_i from a_i and b_i.
6. **Validate Estimates:** Ensure that $0 < a_i < 1$ for valid OU process parameters.

Conclusion

By discretizing the OU process and fitting an AR(1) model using linear regression, we estimate the parameters τ_i, μ_i, and σ_i for each asset. The OLS estimation implicitly performs log-likelihood optimization under the assumption of normally distributed residuals. The constraints on a_i ensure the estimated OU process parameters are valid and meaningful.

Key Takeaways

- The OU process models mean-reverting behavior suitable for financial time series.
- Discretization allows us to fit the continuous-time OU process to discrete data.
- OLS regression provides maximum likelihood estimates under Gaussian noise assumptions.
- Valid parameter estimates require $0 < a_i < 1$ to ensure meaningful τ_i and μ_i.