1. Model estimation

The following standadizes various asset covariance model estimates. The main objective an estimate of a $p \times p$ covariance matrix Σ.

The input is always a data matrix of returns to assets over time, i.e.,

INPUT: A $p \times n$ matrix Y of asset returns and a list of options.

Here, n is the number of observations of the return $y \in \mathbb{R}^p$. These can be at different frequencies (daily, weekly, monthly, etc).

The naive (MLE) estimate of Σ is the sample covariance matrix $\mathrm{S} = \mathrm{Y}\mathrm{Y}^\top/n$. Principal Component Analysis (PCA) regularizes this estimate by assuming a low dimensional approximation that captures the maximum variance in the data.

PCA may be viewed as a method of estimating a canonical factor model (see `am_standards`) for which the covariance matrix Σ takes the form

$$\Sigma = \Pi\Psi\Pi^\top + \Omega = \sigma^2\beta\beta^\top + \Gamma\Lambda\Gamma^\top + \Omega \tag{1}$$

- $\Pi = (\beta, \Gamma)$ is a $p \times (1+q)$ matrix of factor exposures.
- There are $(q+1)$ factors (i.e. at least one factor).
- β is a p vector of market factor exposures,

$$\text{requirement}: \mathrm{m}(\beta) = 1\,.$$

- σ is the variance of the market factor.
- $\Lambda = \mathrm{diag}(\lambda_1^2, \ldots, \lambda_q^2)$ are the variances of non-market factors.
- $\Gamma = (\gamma^1, \ldots, \gamma^q)$ is a $p \times q$ matrix of non-market factor exposures,

$$\text{requirement}: \mathrm{m}(\gamma^k) \geq 0 \text{ and } \mathrm{s}^2(\gamma^k) = 1\,.$$

- Ω is the $p \times p$ matrix of specific risks.

2. PCA prototype

The following definitions in `numpydoc` (https://numpydoc.readthedocs.io/en/latest/example.html) style define the prototypes for the input and outut dictionaries (here treated as classes with attributes) used to construct a PCA model.

```
Python v3.9.7
src/asset_models/PCA.py

class return_data():
    """ Specification for data input to PCA

    Attributes
    ----------
    source : str
        Data source path and time stamp.
    n : int
        The number of observations.
    p : int
        The number of assets.
    data : numpy.array
        n x p matrix of returns to assets.
    freq : str
        Data observation frequency in {'day', 'week', 'month',
        'quarter', 'year'}.
    start : time
        Start of observation period.
    end  : time
        End of observation period.

class pca_options(data):
    """ Options for PCA analysis for the input data

    Attributes
    ----------
    number_factors : int
        Number of factors if greater than 0; estimate otherwise.
    exposure_adjustments : list
        List of adjustements for exposures to factors (e.g. the
        James-Stein or correlation matrix based methods).
    variance_adjustments : list
        List of adjustements for factors variances (e.g. shrinkage
        estimators, random matrix theory based corrections.)
    specific_adjustments : list
        List of adjustments to specific risk estimates.

    """


class exposure_adjustment(id)
    """ Intructions to adjust factor exposures
```

```
    Attributes
    ----------
    factor_id : int
    type : str
        Possible types are {'JS', 'COR'}
    """


class variance_adjustment(id)
    """ Intructions to adjust factor variances

    Attributes
    ----------
    factor_id : int
    type : str
        Possible types are {'RMT', 'MP'}
    """

class factor_model(id)
    """ An estimated asset model

    Attributes
    ----------
    p : int
        The number of assets.
    n : int
        The number of observations.
    q : int
        The number of factors.
    method : str
        The method used to construct the model in {'PCA', ...}
    code : str
        Code version used to estimate the model
    options : dict
        Options passed to the method in {'pca_options', ...}
    exposures : numpy.array
        The p x q matrix of factor exposures <am_standards>
    variances : numpy.array
        The q vector of factor variances <am_standards>
    specific : numpy.array
        Either a p vector of specific risks for each asset or a
        p x p covariance matrix of the specific returns.
```

```
def pca_model(data, options)
""" Main routing for generating a PCA model

    Parameters
    ----------
    data : return_data
        The returns data dictionary
    options : pca_options
        Specification for a particular models (default?)

    Returns
    -------
    model : dict
        Estimated model.

    Notes
    -----

    References
    ----------
    Path to documents <am_standards>.

    Examples
    --------
```

3. PCA recipes

The recipe for a standardized PCA model is as follows.

INPUT: Y and a number q.

Step 1. Form the sample covariance $\mathrm{S} = \mathrm{YY}^\top / n$

Step 2. Extract q eigenvectors $h^{(1)}, \ldots, h^{(q)}$ from S along with their eigenvalues $s_1^2 \geq \cdots \geq s_q^2$ (largest q eigenvalues of S where $n > q$).

Step 3 Construct $\hat{\mathrm{B}}$ as follows. The first column and $\hat{\mathrm{V}}_{11}$ is

$$\hat{\beta} = \frac{h^{(1)}}{m(h^{(1)})} \qquad \text{and} \qquad \hat{\mathrm{V}}_{11} = s_1^2 m^2(h^{(1)}).$$

The kth column of $\hat{\mathrm{B}}$ for $1 < k \le q$ and $\hat{\mathrm{V}}_{kk}$ is set to

$$\hat{\gamma}^{(k)} = \frac{h^{(k)}}{m(h^{(1)})} \quad \text{and} \quad \hat{\mathrm{V}}_{kk} = \mathcal{s}_k^2 m^2(h^{(1)}).$$

*Note, the normalization uses $h^{(1)}$, not $h^{(k)}$.

Step 4. Estimate the diagonal specific return covariance as

$$\hat{\Delta} = \mathrm{diag}(\mathrm{S} - \hat{\mathrm{B}}\hat{\mathrm{V}}\hat{\mathrm{B}}^\top).$$

– **Return** $(\hat{\mathrm{B}}, \hat{\mathrm{V}}, \hat{\Delta})$

*Note, $\mathrm{S} - \hat{\mathrm{B}}\hat{\mathrm{V}}\hat{\mathrm{B}}^\top$ may be used as basis for more general estimates of a matrix $\hat{\Omega}$, e.g. eigenvalue truncation, sparsification, etc.

The empirical literature is mixed on how to select the number of factors q. Even for US equitites which has been under active investigation for 60+ years there is disagreement in the empirical literature.[1] However, there are statistical approaches to selecting the estimate $\hat{q}$. The following recipes come from **?**.

INPUT: The sample covariance matrix S.

1. Let $\delta_0 > 0$ be some threshold and $q_{\min}$ and $q_{\max} \le n$ be plausible lower/upper bounds on q. For eigenvaues $\mathcal{s}_1^2 \ge \cdots \ge \mathcal{s}_n^2$ of S,

$$\hat{q} = \max_{1 \le i \le q_{\max}} \left\{ i \, : \, \mathcal{s}_i^2 - \mathcal{s}_{i+1}^2 \ge \delta_0 \right\}$$

2. Let $q_{\min}$ and $q_{\max} \le n$ be plausible lower/upper bounds on q. For eigenvaues $\mathcal{s}_1^2 \ge \cdots \ge \mathcal{s}_n^2$ of S, we take

$$\hat{q} = \mathrm{argmax}_{q_{\min} \le i \le q_{\max}} \left(\frac{\mathcal{s}_i^2 - \mathcal{s}_{i+1}^2}{\mathcal{s}_{i+1}^2 - \mathcal{s}_{i+2}^2} \right).$$

3. Let $q_{\min}$ and $q_{\max} \le n$ be plausible lower/upper bounds on q and set $\nu_i = \sum_{j=i+1}^n \mathcal{s}_j^2$ for eigenvaues $\mathcal{s}_1^2 \ge \cdots \ge \mathcal{s}_n^2$ of S, we take

$$\hat{q} = \mathrm{argmax}_{q_{\min} \le i \le q_{\max}} \left(\frac{\log(\nu_{i-1}/\nu_i)}{\log(\nu_i/\nu_{i+1})} \right).$$

4. Let R be the correlation matrix for S (i.e., $\mathrm{R} = \mathrm{D}^{-1}\mathrm{S}\mathrm{D}^{-1}$ where

[1] Various authors propose evidence for anywhere between one and six factors.

$\mathrm{D} = \mathrm{diag}(\mathrm{S})$ and let $\rho_1^2 \geq \cdots \geq \rho_p^2$ be the eigenvalues of R.

$$\hat{q} = \max_{1 \leq i \leq p} \{ i : \rho_i^2 > 1 \}$$

*A more advanced version of this estimator is in **?**.

The recipe for PCA can be significantly sped up when p is much larger than n, say $p \geq 2n$. The following recipe replaced **Step 2** of the PCA procedure.

INPUT: Y and a number q (assumes $p > n$)

1. Compute the $n \times n$ dual sample covariance matrix $\mathrm{L} = \mathrm{Y}^\top \mathrm{Y}/p$.
2. Extract q eigenvectors $u^{(1)}, \ldots, u^{(q)}$ of L with the largest eigenvalues $\ell_1^2 \geq \cdots \ell_q^2$ and for $1 \leq i \leq q$ set

$$h^{(i)} = \frac{\mathrm{Y} u^{(i)}}{\ell_i \sqrt{p}} \quad \text{and} \quad \mathcal{s}_i^2 = \ell^2 p/n\,.$$

– **Return** $(\mathcal{s}_i^2, h^{(i)})_{1 \leq i \leq q}$ as the eigenpairs of $\mathrm{S} = \mathrm{Y}\mathrm{Y}^\top/n$.

The estimate of the first (market) factor $\hat{\beta}$ provided by PCA is heavily biased. The following procedure is a James-Stein type correction for PCA aimed to remedy this. It is meant as an addon to **Step 3** in the PCA recipe.

INPUT: The first eigenvector $h = h^{(1)}$ of S and eigenvalues $\mathcal{s}_1^2, \ldots, \mathcal{s}_q^2$ (or alternatively the eigenvalues of L in the spedup version $\ell_1^2 \geq \cdots \geq \ell_q^2$).

1. Compute the sample average $m(h) = \sum_{i=1}^p h_i/p$, the sample variance $\mathrm{s}^2(h) = \sum_{i=1}^p (h_i - m(h))^2/p$ and

$$\mathrm{c} = 1 - \frac{\hat{\nu}^2}{\mathcal{s}_1^2 \mathrm{s}^2(h)} \quad \text{where} \quad \hat{\nu}^2 = \left(\frac{\mathrm{tr}(\mathrm{S}) - (\mathcal{s}_1^2 + \cdots + \mathcal{s}_q^2)}{\min(n, p) - q} \right) / p\,.$$

If the input is the eigenvalues $\ell_1^2 \geq \cdots \geq \ell_q^2$ of L, we can set

$$\mathrm{c} = 1 - \frac{\hat{\nu}^2}{\ell_1^2 \mathrm{s}^2(h)} \quad \text{where} \quad \hat{\nu}^2 = \left(\frac{\mathrm{tr}(\mathrm{L}) - (\ell_1^2 + \cdots + \ell_q^2)}{\min(n, p) - q} \right) / p\,.$$

2. Compute the corrected vector

$$\hat{\beta}^{\mathrm{JS}} = \frac{m(h) + \mathrm{c}\,(h - m(h))}{m(h)}\,.$$

*Notation, $u - x = (u_1 - x, \ldots, u_p - x)$ for $u \in \mathbb{R}^p$ and $x \in \mathbb{R}$.

Another PCA variation uses the correlation matrix R to address the issue of bias. In the estimated model $\hat{\Sigma} = \hat{\mathrm{B}}\hat{\mathrm{V}}\hat{\mathrm{B}}^\top + \hat{\Delta}$ this addresses only the estimation of the columns of $\hat{\mathrm{B}}$ and should not be used to adjust the estimate of $\hat{\mathrm{V}}$ in the PCA procedure. Accordingly, it may be used to adjust all or only of the columns of $\hat{\mathrm{B}}$. For example, the first column may (and perhaps should) be James-Stein corrected as above. This also may be nicely combined with the correlation based q estimation recipe.

INPUT: Y and a number q.

1. Compute $\mathrm{D} = \mathrm{diag}(\mathrm{S})$ with $\mathrm{S} = \mathrm{Y}\mathrm{Y}^\top/n$ efficienly.
2. Extract the eigevector h of S with the largest eigenvalue.
3. Let R be the correlation matrix for S (i.e., $\mathrm{R} = \mathrm{D}^{-1}\mathrm{S}\mathrm{D}^{-1}$).
4. Extract q eigenvectors $v^{(1)}, \ldots, v^{(q)}$ from R correspondind to the $q \leq n$ largest eigenvues (sorted in decreasing order).
5. Construct $\hat{\mathrm{B}}$ as follows. The first column and $\hat{\mathrm{V}}_{11}$ is

$$\hat{\beta} = \frac{v^{(1)}}{m(v^{(1)})} \quad \text{and} \quad \hat{\mathrm{V}}_{11} = \mathscr{s}_1^2 m^2(h)\,.$$

The kth column of $\hat{\mathrm{B}}$ for $1 < k \leq q$ and $\hat{\mathrm{V}}_{kk}$ is set to

$$\hat{\gamma}^{(k)} = \frac{v^{(k)}}{m(h)} \quad \text{and} \quad \hat{\mathrm{V}}_{kk} = \mathscr{s}_k^2 m^2(h)\,.$$

*Note, the normalization uses h, not $h^{(k)}$ nor $v^{(k)}$.

Step 4. Estimate the diagonal specific return covariance as

$$\hat{\Delta} = \mathrm{diag}(\mathrm{S} - \hat{\mathrm{B}}\hat{\mathrm{V}}\hat{\mathrm{B}}^\top).$$

– **Return** $(\hat{\mathrm{B}}, \hat{\mathrm{V}}, \hat{\Delta})$