APMA1690: Homework # 10 (Due by 11pm December 8)

"Subtle is the Lord, but malicious he is not."

— Albert Einstein

1 Review

I would suggest you go through the review section before going to the problem set.

1.1 Manifold Hypothesis

Every science is based on one or several assumptions. Manifold learning (also known as 'dimension reduction') is not an exception — manifold[1] learning is based on the so-called 'manifold hypothesis:' *high dimensional data tend to lie in the vicinity of a low dimensional manifold* (Fefferman et al., 2016).[2] The 'high dimension' is usually referred to as the 'extrinsic dimension,' and the 'low dimension' is usually referred to as the 'intrinsic dimension.'

1.2 Notations

- d and D are two positive integers satisfying $d < D$. Hereafter, d denotes an intrinsic dimension, and D denotes an extrinsic dimension.
- For any matrix $\boldsymbol{A}$, its transpose is denoted as $\boldsymbol{A}^\intercal$.
- For any $\boldsymbol{x} = (x_1, \ldots, x_D)^\intercal \in \mathbb{R}^D$, its norm is defined as $\|\boldsymbol{x}\| := \sqrt{\sum_{k=1}^D x_k^2}$. Furthermore, if the column vector $\boldsymbol{x}$ is viewed as a D-by-1 matrix, we have

$$\|\boldsymbol{x}\|^2 = \boldsymbol{x}^\intercal \boldsymbol{x}. \tag{1.1}$$

- All vectors in this problem set are viewed as column vectors, which is also a convention widely adopted in the literature.
- $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$ are identically and independently distributed (iid) $\mathbb{R}^D$-valued random variables (i.e., D-dimensional random vectors).

[1] The precise definition of manifolds is quite technical and beyond the scope of APMA 1690. You may just roughly think of a manifold as a surface/curve/hyperplane/line.

[2] The first author Charles Fefferman achieved a full professorship at the University of Chicago at the age of 22, making him the youngest full professor ever appointed in the United States. Fefferman entered the University of Maryland at age 14, graduated with degrees in mathematics and physics at 17, and earned his PhD in mathematics three years later from Princeton University.

1.3 A Glimpse of Manifold Learning

The manifold hypothesis described in Section 1.1 can be mathematically represented as follows:

- Latent d-dimensional random vectors $\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_n$ are generated by 'the Lord' in an iid way. They are not available to us.
- There exists a function $g : \mathbb{R}^d \to \mathbb{R}^D$. The image of g, i.e.,

  $$\mathcal{M} := \left\{ g(z) \,:\, z \in \mathbb{R}^d \right\}, \tag{1.2}$$

  is called a manifold. Neither g nor $\mathcal{M}$ is available to us.
- The data points $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$ that we observe in the high-dimensional space $\mathbb{R}^D$ are generated by the following mechanism

  $$\boldsymbol{X}_i = g(\boldsymbol{Z}_i) + \boldsymbol{\varepsilon}_i, \quad \text{for all } i = 1, 2, \ldots, n, \tag{1.3}$$

  where $\boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_n$ are iid D-dimensional random vectors playing the role of random noise.

The ultimate goal of manifold learning is to learn the low-dimensional manifold $\mathcal{M}$ defined in Eq. (1.2). Then, the observed high-dimensional data points $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$ are reduced to their projections $\pi_{\mathcal{M}}(\boldsymbol{X}_1), \ldots, \pi_{\mathcal{M}}(\boldsymbol{X}_n)$ on the low-dimensional manifold $\mathcal{M}$. See Figure 1 for illustrations.

1.4 Two Branches of Manifold Learning

Dimension reduction/manifold learning refers to a collection of methods reducing the dimensionality of data while preserving most information in the data. There are the following two branches of dimension reduction:

- Linear dimension reduction, e.g., principal component analysis (PCA, Pearson, 1901).
- Nonlinear dimension reduction, e.g., principal curves (Hastie and Stuetzle, 1989), Isomap (Tenenbaum et al., 2000), local linear embedding (Roweis and Saul, 2000; Wu and Wu, 2018), Laplacian eigenmaps (Belkin and Niyogi, 2001), diffusion map (Coifman et al., 2005; Coifman and Lafon, 2006), principal manifolds (Smola et al., 2001; Meng and Eloyan, 2021).

While nonlinear dimension reduction is still developing, linear dimension reduction is relatively well-developed. We focus on the most widely used linear dimension reduction approach — PCA.

1.5 Mathematical Preparations

Materials in this subsection are from Sections 1.4 and 1.5 of Seber and Lee (2012).

For any random vector $\boldsymbol{X} = (X_1, \ldots, X_n)^\intercal$ with any dimension $n = 1, 2, \ldots$, we define its **covariance matrix** $\mathbb{V}(\boldsymbol{X})$ as follows (Seber and Lee, 2012, Definition 1.3)

$$\mathbb{V}(\boldsymbol{X}) := \begin{pmatrix} \operatorname{Cov}(X_1, X_1) & \operatorname{Cov}(X_1, X_2) & \cdots & \operatorname{Cov}(X_1, X_n) \\ \operatorname{Cov}(X_2, X_1) & \operatorname{Cov}(X_2, X_2) & \cdots & \operatorname{Cov}(X_2, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ \operatorname{Cov}(X_n, X_1) & \operatorname{Cov}(X_n, X_2) & \cdots & \operatorname{Cov}(X_n, X_n) \end{pmatrix} = \Big(\operatorname{Cov}(X_i, X_j) \Big)_{1 \le i,j \le n}, \tag{1.4}$$

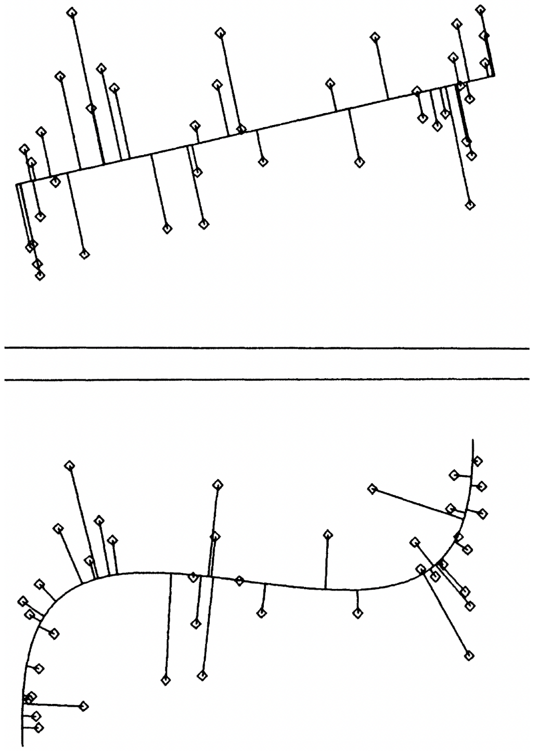


Figure 1: (2-dimensional) data points $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$ and their projections $\pi_{\mathcal{M}}(\boldsymbol{X}_1), \ldots, \pi_{\mathcal{M}}(\boldsymbol{X}_n)$ on (1-dimensional) manifolds. The upper panel illustrates a linear manifold learning result, and the lower panel illustrates a nonlinear manifold learning result. This figure comes from Hastie and Stuetzle (1989).

which is an n-by-n matrix. Obviously, $\mathbb{V}(\boldsymbol{X})$ is a symmetric matrix. Furthermore, we have the following claims

Claim 1.1. *The covariance matrix $\mathbb{V}(\boldsymbol{X})$ defined in Eq. (1.4) is positive semi-definite .*

Proof: The proof is a homework problem.
One may apply the following claim to prove Claim 1.1.

Claim 1.2 (Theorem 1.3 of Seber and Lee (2012))**.** *Let $\boldsymbol{A}$ by any m-by-n deterministic matrix. Then, we have*

$$\mathbb{V}(\boldsymbol{A}\boldsymbol{X}) = \boldsymbol{A}\mathbb{V}(\boldsymbol{X})\boldsymbol{A}^\intercal.$$

Claim 1.3 (Theorem 1.5 of Seber and Lee (2012))**.** *Let $\boldsymbol{A}$ be any n-by-n symmetric matrix. Suppose $\boldsymbol{\Sigma} := \mathbb{V}(\boldsymbol{X})$ and $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)^\intercal = \mathbb{E}\boldsymbol{X}$. Then, we have*

$$\mathbb{E}(\boldsymbol{X}^\intercal \boldsymbol{A}\boldsymbol{X}) = \operatorname{tr}(\boldsymbol{A}\boldsymbol{\Sigma}) + \boldsymbol{\mu}^\intercal \boldsymbol{A}\boldsymbol{\mu}, \tag{1.5}$$

where $\operatorname{tr}(\cdot)$ denotes the trace of a square matrix.

1.6 Principal Component Analysis

Without loss of generality, we assume that the data points $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$ have been centralized. That is, hereafter, we are under the following assumption

Assumption 1. *Data points $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$ satisfy*

$$\mathbb{E}\boldsymbol{X}_i = (0, 0, \ldots, 0)^\intercal =: \mathbf{0}, \qquad \textit{for all } i = 1, 2, \ldots, n. \tag{1.6}$$

Assumption 1 is widely adopted in the PCA literature. It can easily be satisfied as $\mathbb{E}(\boldsymbol{X}_i - \frac{1}{n}\sum_{j=1}^n \boldsymbol{X}_j) = \mathbf{0}$, and we can always update $\boldsymbol{X}_i$ by $\boldsymbol{X}_i \leftarrow \boldsymbol{X}_i - \frac{1}{n}\sum_{j=1}^n \boldsymbol{X}_j$.

The PCA framework assumes that the $\mathcal{M}$ defined in Eq. (1.2) is a d-dimensional hyperplane embedded in $\mathbb{R}^D$ and going through the origin of $\mathbb{R}^D$.[3] Hereafter, we use V to denote d-dimensional hyperplanes.

1.6.1 PCA from the Viewpoint of Fitting Errors

Let V be a d-dimensional hyperplane embedded in $\mathbb{R}^D$ and going through the origin of $\mathbb{R}^D$. For any $\boldsymbol{x} \in \mathbb{R}^D$, its projection on the hyperplane V is denoted as $\boldsymbol{P}_V\boldsymbol{x}$.

PCA is the model claiming that the underlying manifold (see Eq. (1.2)) is the following hyperplane V^* that minimizes the fitting error $\mathbb{E}\left(\|\boldsymbol{X} - \boldsymbol{P}_V\boldsymbol{X}\|^2\right)$

$$V^* = \underset{V}{\operatorname{argmin}} \left\{\mathbb{E}\left(\|\boldsymbol{X} - \boldsymbol{P}_V\boldsymbol{X}\|^2\right) : V \text{ is a } d\text{-dimensional hyperplane going through the origin of } \mathbb{R}^D\right\}. \tag{1.7}$$

Hereafter, all minimizations/minimizations are taken across all d-dimensional hyperplanes V going through the origin of $\mathbb{R}^D$.

1.6.2 PCA from the Viewpoint of Variance

By the Pythagorean theorem, we have $\|\boldsymbol{X} - \boldsymbol{P}_V\boldsymbol{X}\|^2 = \|\boldsymbol{X}\|^2 - \|\boldsymbol{P}_V\boldsymbol{X}\|^2$. Therefore,

$$\min_V \left\{\mathbb{E}\left(\|\boldsymbol{X} - \boldsymbol{P}_V\boldsymbol{X}\|^2\right)\right\} = \mathbb{E}\left(\|\boldsymbol{X}\|^2\right) - \max_V \left\{\mathbb{E}\left(\|\boldsymbol{P}_V\boldsymbol{X}\|^2\right)\right\}.$$

Therefore, we have the following representations

$$\boxed{V^* = \underset{V}{\operatorname{argmin}} \left\{\mathbb{E}\left(\|\boldsymbol{X} - \boldsymbol{P}_V\boldsymbol{X}\|^2\right)\right\} = \underset{V}{\operatorname{argmax}} \left\{\mathbb{E}\left(\|\boldsymbol{P}_V\boldsymbol{X}\|^2\right)\right\}.} \tag{1.8}$$

Eq. (1.8) provides the following two interpretations of the optimal hyperplane V^*:

- V^* is the hyperplane that minimizes the average fitting error $\mathbb{E}\left(\|\boldsymbol{X} - \boldsymbol{P}_V\boldsymbol{X}\|^2\right)$.
- V^* is the hyperplane that makes the projection $\boldsymbol{P}_{V^*}\boldsymbol{X}$ enjoy the largest variance.

Both interpretations indicate that the optimal hyperplane V^* goes through the 'middle' of data points (see the upper panel of Figure 1 for an illustration.).

[3] The 'going through the origin of $\mathbb{R}^D$' corresponds to Assumption 1.

1.6.3 Formulae of V^* and $\boldsymbol{P}_{V^*}$

Since the covariance matrix $\mathbb{V}(\boldsymbol{X})$ is positive semi-definite (see Claim 1.1), we have the following eigen-structure of $\mathbb{V}(\boldsymbol{X})$:

- eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d \geq \ldots \geq \lambda_D \geq 0$;
- the corresponding eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_D$ satsifying $\mathbb{V}(\boldsymbol{X})\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i$ for all $i = 1, 2, \ldots, D$ (the eigenvectors are viewed as column vectors).
- The eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_D$ are orthonormal, i.e.,

$$\boldsymbol{v}_i^\intercal \boldsymbol{v}_j = \begin{cases} 0 & \text{if } i \neq j, \\ 1 & \text{if } i = j. \end{cases} \tag{1.9}$$

Then, we have the following formula of the optimal hyperplane V^* defined in Eq. (1.8)

$$\boxed{V^* = \operatorname{span}\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_d\} = \left\{\sum_{k=1}^{d} \alpha_k \boldsymbol{v}_k \,:\, \alpha_1, \ldots, \alpha_d \in \mathbb{R}\right\},}$$

where span denotes the 'linear span.' We define a D-by-d matrix $\boldsymbol{W}$ as follows

$$\boldsymbol{W} := (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d),$$

i.e., the column vector $\boldsymbol{v}_i$ is the i-th column of the matrix $\boldsymbol{W}$. Then, the projection $\boldsymbol{P}_{V^*}$ can be represented as the following matrix

$$\boxed{\boldsymbol{P}_{V^*} = \boldsymbol{W}\boldsymbol{W}^\intercal,} \tag{1.10}$$

i.e., $\boldsymbol{P}_{V^*}\boldsymbol{X} = \boldsymbol{W}\boldsymbol{W}^\intercal\boldsymbol{X}$.

1.6.4 Choice of the Intrinsic Dimension d

Claim 1.4. *Let $\boldsymbol{X} = (X_1, \ldots, X_D)^\intercal$ be a D-dimensional random vector satisfying $\mathbb{E}X_i = 0$ for all $i = 1, 2, \ldots, D$. The projection matrix $\boldsymbol{P}_{V^*}$ is defined by Eq. (1.10). Then, we have*

1. $\mathbb{E}\left(\|\boldsymbol{X}\|^2\right) = \sum_{k=1}^{D} \lambda_k$.
2. $\mathbb{E}\left(\|\boldsymbol{P}_{V^*}\boldsymbol{X}\|^2\right) = \sum_{k=1}^{d} \lambda_k$.

Proof: The proof is a homework problem.

The following ratio is interpreted as 'the proportion of variance explained by the PCA.'

$$\boxed{r_d := \frac{\mathbb{E}\left(\|\boldsymbol{P}_V\boldsymbol{X}\|^2\right)}{\mathbb{E}\left(\|\boldsymbol{X}\|^2\right)} = \frac{\sum_{k=1}^{d} \lambda_k}{\sum_{k=1}^{D} \lambda_k}.}$$

This ratio r_d provides criteria for choosing the intrinsic dimension d in applications. A widely adopted criterion for choosing d is the following:[4]

- $r_d > 95\%$,
- and $r_{d-1} < 95\%$.

[4] The percentage 95% can be replaced with any other percentage you like.

1.6.5 Further Interpretation of r_d

Suppose the data $\boldsymbol{X}$ is generated by the following mechanism (which is a special case of Eq. (1.3)):

$$\boldsymbol{X} = \boldsymbol{L}\boldsymbol{Z} + \boldsymbol{\varepsilon},$$

- $\boldsymbol{Z}$ is a latent d-dimensional random vector, unavailable (available to 'the Lord' rather than us);
- $\boldsymbol{L}$ is a deterministic D-by-d matrix, unavailable to us;
- $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_D)^\intercal$ is a D-dimensional random vector, playing the role of random noise; we assume $\varepsilon_1, \ldots, \varepsilon_D \overset{iid}{\sim} N(0, \sigma^2)$, which implies

  $$\mathbb{V}(\boldsymbol{\varepsilon}) = \begin{pmatrix} \sigma^2 & & \\ & \ddots & \\ & & \sigma^2 \end{pmatrix}, \tag{1.11}$$

  which is a D-by-D diagonal matrix;
- The latent random variable $\boldsymbol{Z}$ and noise $\boldsymbol{\varepsilon}$ are independent;
- The D-dimensional random vector $\boldsymbol{X}$ represents the data we observe.

Claim 1.2, together with the independence between $\boldsymbol{Z}$ and $\boldsymbol{\varepsilon}$, implies

$$\mathbb{V}(\boldsymbol{X}) = \boldsymbol{L}\mathbb{V}(\boldsymbol{Z})\boldsymbol{L}^\intercal + \mathbb{V}(\boldsymbol{\varepsilon}). \tag{1.12}$$

We have the following matrix rank estimation

$$\operatorname{rank}(\boldsymbol{L}\mathbb{V}(\boldsymbol{Z})\boldsymbol{L}^\intercal) \leq \operatorname{rank}(\mathbb{V}(\boldsymbol{Z})) \leq d,$$

where the last inequality comes from that $\mathbb{V}(\boldsymbol{Z})$ is a d-by-d matrix. Then, the matrix $\boldsymbol{L}\mathbb{V}(\boldsymbol{Z})\boldsymbol{L}^\intercal$ has at most d non-zero eigenvalues $\tilde{\lambda}_1, \ldots, \tilde{\lambda}_d > 0$. Specifically, we have the following eigen decomposition

$$\boldsymbol{L}\mathbb{V}(\boldsymbol{Z})\boldsymbol{L}^\intercal = \boldsymbol{U} \begin{pmatrix} \tilde{\lambda}_1 & & & & & \\ & \ddots & & & & \\ & & \tilde{\lambda}_d & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{pmatrix} \boldsymbol{U}^\intercal, \tag{1.13}$$

where $\boldsymbol{U}$ is an orthogonal matrix (i.e., $\boldsymbol{U}\boldsymbol{U}^\intercal = \boldsymbol{U}^\intercal\boldsymbol{U} = \boldsymbol{I}_D =$ the D-by-D identity matrix). Combining Eq. (1.11), (1.12), and (1.13), we have

$$\mathbb{V}(\boldsymbol{X}) = \boldsymbol{U} \begin{pmatrix} \tilde{\lambda}_1 + \sigma^2 & & & & & \\ & \ddots & & & & \\ & & \tilde{\lambda}_d + \sigma^2 & & & \\ & & & \sigma^2 & & \\ & & & & \ddots & \\ & & & & & \sigma^2 \end{pmatrix} \boldsymbol{U}^\intercal. \tag{1.14}$$

Therefore, the eigenvalues $\{\lambda_k\}_{k=1}^D$ of $\mathbb{V}(\boldsymbol{X})$ are the following

$$\begin{aligned}
&\lambda_1 = \tilde{\lambda}_1 + \sigma^2,\\
&\vdots\\
&\lambda_d = \tilde{\lambda}_d + \sigma^2,\\
&\lambda_{d+1} = \sigma^2,\\
&\vdots\\
&\lambda_D = \sigma^2.
\end{aligned} \tag{1.15}$$

If noise is very small (i.e., $\sigma^2 \approx 0$), we have

$$\lambda_1 \geq \ldots \geq \lambda_d \geq \lambda_{d+1} \approx \lambda_{d+2} \approx \ldots \approx \lambda_D \approx 0.$$

Furthermore, we have

$$r_d = \frac{\sum_{k=1}^d \lambda_k}{\sum_{k=1}^D \lambda_k} = \frac{\sum_{k=1}^d \tilde{\lambda}_k + d \cdot \sigma^2}{\sum_{k=1}^d \tilde{\lambda}_k + D \cdot \sigma^2}. \tag{1.16}$$

If the noise is very small (i.e., $\sigma^2 \approx 0$), Eq. (1.16) implies

$$1 \approx r_D \approx r_{D-1} \approx \ldots \approx r_{d+1} \approx r_d.$$

1.6.6 Computation in Applications

The only input for PCA is the covariance matrix $\mathbb{V}(\boldsymbol{X})$. However, the precise covariance matrix $\mathbb{V}(\boldsymbol{X})$ is unavailable in applications. In practical applications, when working with observed data points $\{\boldsymbol{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,D})^\intercal\}_{i=1}^n \subseteq \mathbb{R}^D$, the following sample covariance matrix $\widehat{\mathbb{V}(\boldsymbol{X})}$ is a substitute for the precise covariance matrix (thanks to the Law of Large Numbers)

$$\begin{aligned}
&\widehat{\mathbb{V}(\boldsymbol{X})} := \begin{pmatrix} \widehat{\operatorname{Cov}}(X_1, X_1) & \widehat{\operatorname{Cov}}(X_1, X_2) & \cdots & \widehat{\operatorname{Cov}}(X_1, X_D) \\ \widehat{\operatorname{Cov}}(X_2, X_1) & \widehat{\operatorname{Cov}}(X_2, X_2) & \cdots & \widehat{\operatorname{Cov}}(X_2, X_D) \\ \vdots & \vdots & \ddots & \vdots \\ \widehat{\operatorname{Cov}}(X_D, X_1) & \widehat{\operatorname{Cov}}(X_D, X_2) & \cdots & \widehat{\operatorname{Cov}}(X_D, X_D) \end{pmatrix},\\
&\text{where } \widehat{\operatorname{Cov}}(X_i, X_j) := \frac{1}{n-1} \sum_{w=1}^n \left[\left(x_{i,w} - \frac{1}{n} \sum_{k=1}^n x_{i,k} \right) \left(x_{jw} - \frac{1}{n} \sum_{l=1}^n x_{j,l} \right) \right].
\end{aligned} \tag{1.17}$$

1.6.7 A Numerical Example

In this subsection, we provide a naive numerical example illustrating the aforementioned theoretical discussions.

- $Z \sim N(0,1)$;
- $\boldsymbol{L} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$;

- $\boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{pmatrix}$, where $\varepsilon_1, \varepsilon_2, \varepsilon_3 \overset{iid}{\sim} N(0,\, 0.04)$;
- $\boldsymbol{X} \overset{\text{def}}{=} \boldsymbol{L}Z + \boldsymbol{\varepsilon} = \begin{pmatrix} Z + \varepsilon_1 \\ Z + \varepsilon_2 \\ Z + \varepsilon_3 \end{pmatrix}$.

Then, we have

- $\mathbb{V}(\boldsymbol{L}Z) = \boldsymbol{L}\boldsymbol{L}^\intercal = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$, whose eigenvalues are $\tilde{\lambda}_1 = 3, \tilde{\lambda}_2 = \tilde{\lambda}_3 = 0$ (you have learned how to compute eigenvalues in your linear algebra class);
- Eq. (1.15) implies that the eigenvalues of $\mathbb{V}(\boldsymbol{X})$ are

$$\begin{aligned} \lambda_1 &= 3 + 0.04 = 3.04, \\ \lambda_2 &= \lambda_3 = 0.04. \end{aligned}$$

The results of a numerical experiment conducted using R are presented as follows, and they are compatible with our theoretical discussion.

```
> n=10000000
> sigma=0.2
>
> Z=rnorm(n)
> L=matrix(1, nrow = 3, ncol = 1)
> e=cbind(rnorm(n, sd=sigma), rnorm(n, sd=sigma), rnorm(n, sd=sigma))
> X=t(L%*%Z)+e
>
> covariance_matrix=var(X)
> eigenstructure=eigen(covariance_matrix)
> eigenstructure
eigen() decomposition
$values
[1] 3.04140759 0.04002077 0.03997760

$vectors
          [,1]       [,2]       [,3]
[1,] 0.5773425  0.1238912  0.8070481
[2,] 0.5772954 -0.7609286 -0.2961717
[3,] 0.5774129  0.6368977 -0.5108382
```

2 Problem Set

1. (2 points) Prove Claim 1.1.

   Claim: The covariance matrix

$$\mathbb{V}(X) = \begin{pmatrix} \mathrm{Cov}(X_1, X_1) & \mathrm{Cov}(X_1, X_2) & \cdots & \mathrm{Cov}(X_1, X_n) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Cov}(X_2, X_2) & \cdots & \mathrm{Cov}(X_2, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(X_n, X_1) & \mathrm{Cov}(X_n, X_2) & \cdots & \mathrm{Cov}(X_n, X_n) \end{pmatrix} = \Big(\mathrm{Cov}(X_i, X_j) \Big)_{1 \leq i,j \leq n}$$

   is positive semi-definite.

   Proof: By definition, a matrix M is positive semi-definite if it is symmetric and if $z^T M z$ is nonnegative for every nonzero real column vector z.

   Since

$$\mathrm{Cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])] = \mathbb{E}[(Y - \mathbb{E}[Y])(X - \mathbb{E}[X])] = \mathrm{Cov}(Y, X)$$

   the covariance matrix is symmetric.

   Then let z be any nonzero real n-by-1 column vector. Then by Claim 1.2

$$z^T \mathbb{V}(X) z = \mathbb{V}(z^T X)$$

   but since $z^T X$ is a scalar,

$$\mathbb{V}(z^T X) = \mathrm{Cov}(z^T X, z^T x) = \mathrm{Var}(z^T X) \geq 0$$

   so $z^T \mathbb{V}(X) z \geq 0$ and we are done. ■

2. (3 points) Prove Claim 1.4. (Hint: apply Eq. (1.1), Claim 1.3, properties of trace, Eq. (1.9), and Eq. (1.10).)

   Claim: Let $\boldsymbol{X} = (X_1, \ldots, X_D)^\intercal$ be a D-dimensional random vector satisfying $\mathbb{E} X_i = 0$ for all $i = 1, 2, \ldots, D$. The projection matrix $\boldsymbol{P}_{V^*}$ is defined by Eq. (1.10). Then, we have

   (a) $\mathbb{E}\left(\|\boldsymbol{X}\|^2\right) = \sum_{k=1}^{D} \lambda_k$.

   (b) $\mathbb{E}\left(\|\boldsymbol{P}_{V^*}\boldsymbol{X}\|^2\right) = \sum_{k=1}^{d} \lambda_k$.

   Proof:

   By Eq. (1.1), if $\vec{x}$ is a D-by-1 column vector, we have

$$\|\boldsymbol{x}\|^2 = \boldsymbol{x}^\intercal \boldsymbol{x}.$$

   so

$$\mathbb{E}[||X||^2] = \mathbb{E}[X^T X]$$

   Claim 1.3 says, for any n-by-n symmetric matrix $\boldsymbol{A}$, $\boldsymbol{\Sigma} := \mathbb{V}(\boldsymbol{X})$, and $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)^\intercal = \mathbb{E}\boldsymbol{X}$, we have

$$\mathbb{E}(\boldsymbol{X}^\intercal \boldsymbol{A} \boldsymbol{X}) = \mathrm{tr}(\boldsymbol{A}\boldsymbol{\Sigma}) + \boldsymbol{\mu}^\intercal \boldsymbol{A} \boldsymbol{\mu}, \tag{2.1}$$

Thus, we can write $X^TX = X^TIX$ and

$$\mathbb{E}[X^TX] = \mathbb{E}[X^TIX] = \operatorname{tr}(I\Sigma) + \mu^TI\mu = \operatorname{tr}(\Sigma) + \mu^T\mu = \operatorname{tr}(\mathbb{V}(X)) + (\mathbb{E}X)^T(\mathbb{E}X)$$

By definition of trace,

$$\operatorname{tr}(\mathbb{V}(X)) = \sum_{i=1}^{D} \lambda_i$$

and since $\mathbb{E}X = 0$,

$$(\mathbb{E}X)^T(\mathbb{E}X) = 0$$

so

$$\mathbb{E}\left(\|\boldsymbol{X}\|^2\right) = \sum_{i=1}^{D} \lambda_i$$

and we are done. □

Now for the second part, using Eq. (1.10),

$$\begin{aligned}
\mathbb{E}[||P_V^* X||^2] &= \mathbb{E}[\left|\left|WW^TX\right|\right|^2] \\
&= \mathbb{E}[(WW^TX)^T(WW^TX)] \\
&= \mathbb{E}[X^TWW^TWW^TX]
\end{aligned}$$

where W is a D-by-d matrix defined by $W = (v_1, \dots, v_d)$.

Thus WW^T is a D-by-D matrix and clearly, it is symmetric:

$$(WW^T)^T = (W^T)^TW^T = WW^T$$

so we can apply claim 1.3 to get

$$\mathbb{E}[X^TWW^TWW^TX] = \operatorname{tr}(WW^TWW^T\mathbb{V}(X)) + (\mathbb{E}X)^T(WW^T)^2(\mathbb{E}X) = \operatorname{tr}(WW^TWW^T\mathbb{V}(X))$$

Then using the cyclic property of the trace, we have

$$\operatorname{tr}(WW^TWW^T\mathbb{V}(X)) = \operatorname{tr}(WW^T\mathbb{V}(X)WW^T)$$

We can calculate WW^T as:

$$\begin{aligned}
WW^T &= \begin{pmatrix} \vec{v}_1 & \vec{v}_2 & \dots & \vec{v}_d \end{pmatrix} \begin{pmatrix} \vec{v}_1^T \\ \vec{v}_2^T \\ \vdots \\ \vec{v}_d^T \end{pmatrix} \\
&= v_1v_1^T + v_2v_2^T + \cdots + v_dv_d^T \\
&= \sum_{i=1}^{d} v_iv_i^T
\end{aligned}$$

And since $\mathbb{V}(X)$ is symmetric, we can write

$$\mathbb{V}(X) = Q\Lambda Q^T = \sum_{i=1}^{D} \lambda_i v_i v_i^T$$

where $Q = (v_1, v_2, \dots, v_D)$ and Λ is a diagonal matrix of the eigenvalues of $\mathbb{V}(X)$. So

$$\begin{aligned}
\text{tr}(WW^T\mathbb{V}(X)WW^T) &= \text{tr}(\left(\sum_{i=1}^{d} v_i v_i^T\right)\left(\sum_{j=1}^{D} \lambda_j v_j v_j^T\right)\left(\sum_{k=1}^{d} v_k v_k^T\right)) \\
&= \text{tr}(\left(\sum_{i=1}^{d}\sum_{j=1}^{D}\sum_{k=1}^{d} v_i v_i^T \lambda_j v_j v_j^T v_k v_k^T\right)) \\
&= \text{tr}(\left(\sum_{i=1}^{d}\sum_{j=1}^{D}\sum_{k=1}^{d} \lambda_j v_i v_i^T v_j v_j^T v_k v_k^T\right))
\end{aligned}$$

But since the eigenvectors $v_1, \dots,\ v_D$ are orthonormal:

$$v_i^T v_j = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}$$

All of these terms go to 0 unless $i = j = k$. However, when $d \leq j \leq D$, i and k are at most d so the only terms that survive are when $i = j = k$ and $1 \leq j \leq d$. Thus, we have

$$\text{tr}(WW^T\mathbb{V}(X)WW^T) = \begin{pmatrix} \lambda_1 & & & & & & \\ & \lambda_2 & & & & & \\ & & \ddots & & & & \\ & & & \lambda_d & & & \\ & & & & 0 & & \\ & & & & & \ddots & \\ & & & & & & 0 \end{pmatrix}$$

So

$$\mathbb{E}[||P_V^* X||^2] = \text{tr}(WW^T\mathbb{V}(X)WW^T) = \sum_{i=1}^{d} \lambda_i \quad \blacksquare$$

3. (5 points) Please conduct the following procedures:

   (a) Set $n = 100$.

   (b) Generate n random numbers $z_1, z_2, \dots, z_n \overset{iid}{\sim} N(0, 1)$.

(c) Generate 2-dimensional random vectors $\boldsymbol{\varepsilon}_i = \begin{pmatrix} \varepsilon_{i,1} \\ \varepsilon_{i,2} \end{pmatrix}$ for all $i = 1, 2, \ldots, n$, where

$$\varepsilon_{1,1}, \varepsilon_{2,1}, \ldots, \varepsilon_{n,1}, \varepsilon_{1,2}, \varepsilon_{2,2}, \ldots, \varepsilon_{n,2} \overset{iid}{\sim} N(0, 0.04).$$

(d) Compute $\boldsymbol{x}_i = \begin{pmatrix} x_{i,1} \\ x_{i,2} \end{pmatrix} \overset{\text{def}}{=} \begin{pmatrix} z_i + \varepsilon_{i,1} \\ z_i + \varepsilon_{i,2} \end{pmatrix}$ for all $i = 1, 2, \ldots, n$.

(e) Compute the sample covariance matrix $\widehat{\mathbb{V}(\boldsymbol{X})}$ (see Eq. (1.17)) using $\{\boldsymbol{x}_i\}_{i=1}^n$. Present the matrix $\widehat{\mathbb{V}(\boldsymbol{X})}$.

(f) Compute eigenvalues λ_1, λ_2 and eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2$ of $\widehat{\mathbb{V}(\boldsymbol{X})}$, making the eigenvalues and eigenvectors satisfy the following

- $\lambda_1 \geq \lambda_2$,
- $\widehat{\mathbb{V}(\boldsymbol{X})}\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i$ for $i = 1, 2$, and
- eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2$ are orthonormal (see Eq. (1.9)).

Present the eigenvalues and eigenvectors. (Small numerical errors are allowed.)

(g) Compute the matrix

$$\boldsymbol{P} = \boldsymbol{v}_1 \boldsymbol{v}_1^\intercal, \tag{2.2}$$

which is a 2-by-2 matrix. Present this matrix.

(h) Plot the 2-dimensional data points $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$.

(i) Plot the following straight line

$$V^* = \{\alpha \boldsymbol{v}_1 \,:\, \alpha \in \mathbb{R}\}. \tag{2.3}$$

(j) Plot the 2-dimensional points $\boldsymbol{P}\boldsymbol{x}_1, \boldsymbol{P}\boldsymbol{x}_2, \ldots, \boldsymbol{P}\boldsymbol{x}_n$, where $\boldsymbol{P}$ is defined in Eq. (2.2).

(k) Plot the straight line segments[5] $(\boldsymbol{x}_1,\, \boldsymbol{P}\boldsymbol{x}_1), \ldots, (\boldsymbol{x}_n,\, \boldsymbol{P}\boldsymbol{x}_n)$.

Overlay all the plots. If you conduct all the procedures correctly, the plot you get should look like Figure 2. Please provide the code (in any programming language that you are comfortable with) for conducting the procedures.

[5] $(\boldsymbol{a}, \boldsymbol{b})$ denotes the straight line segment connecting points $\boldsymbol{a}$ and $\boldsymbol{b}$.

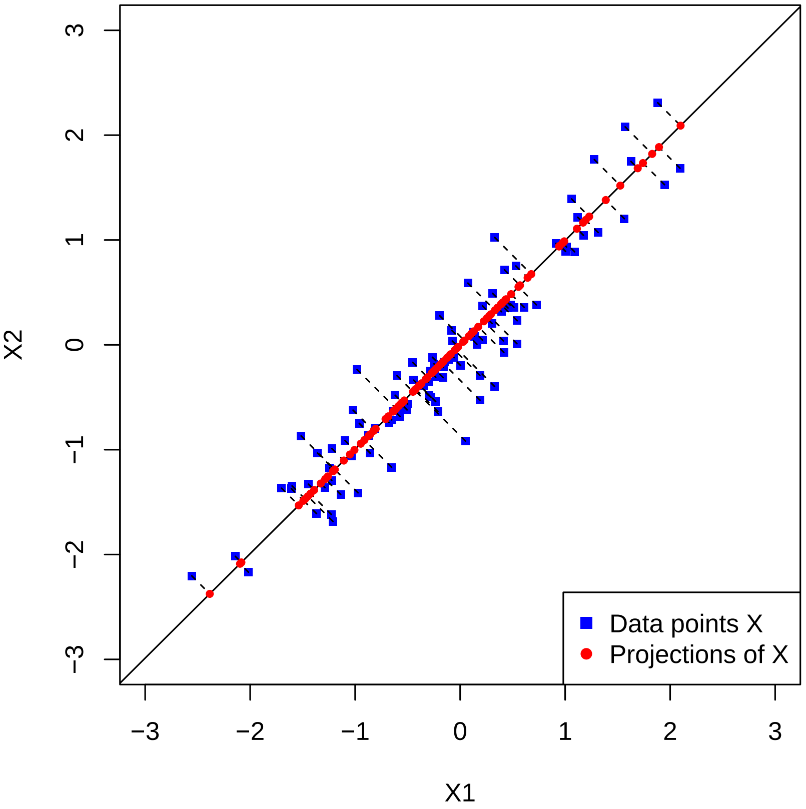
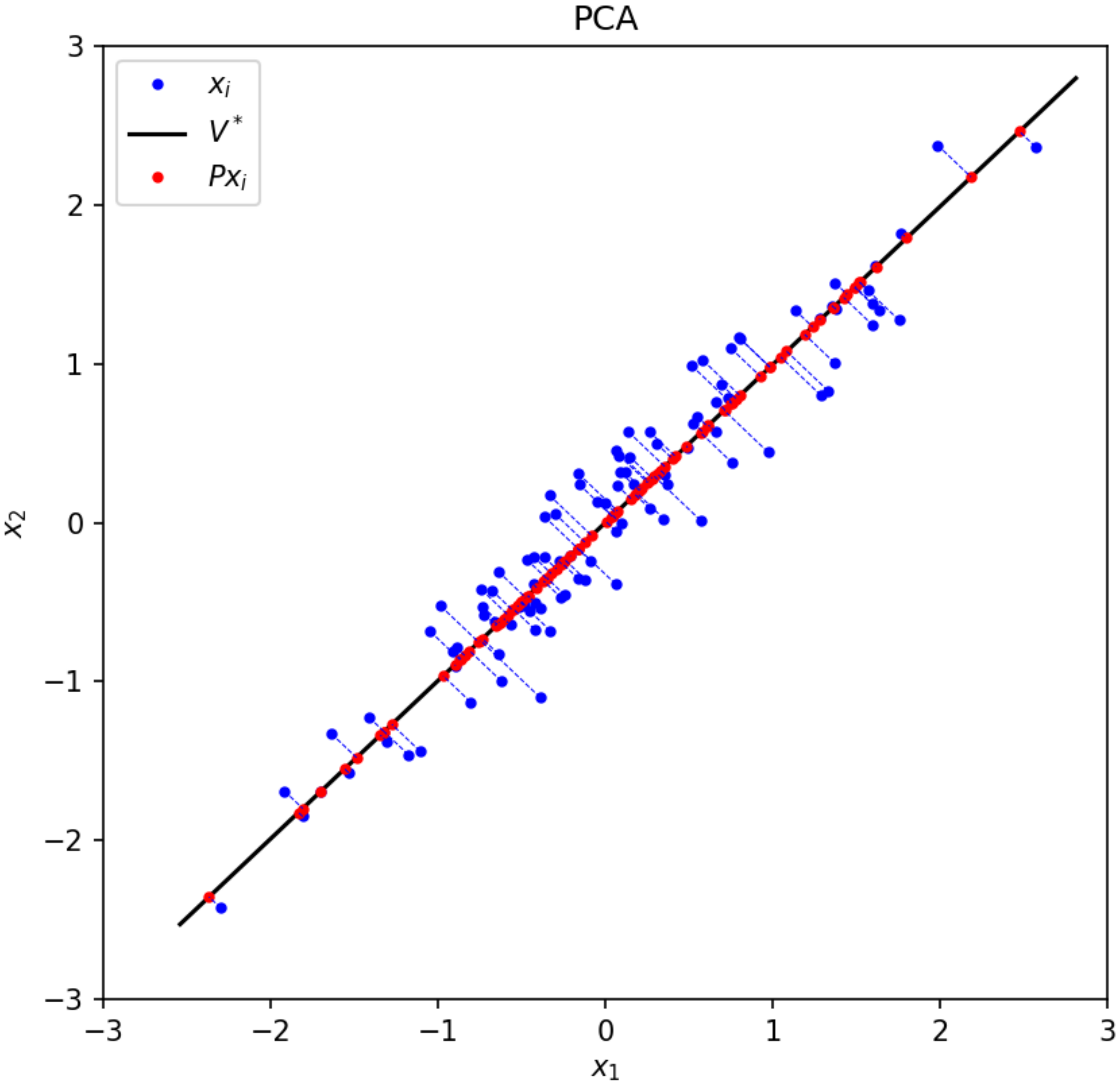
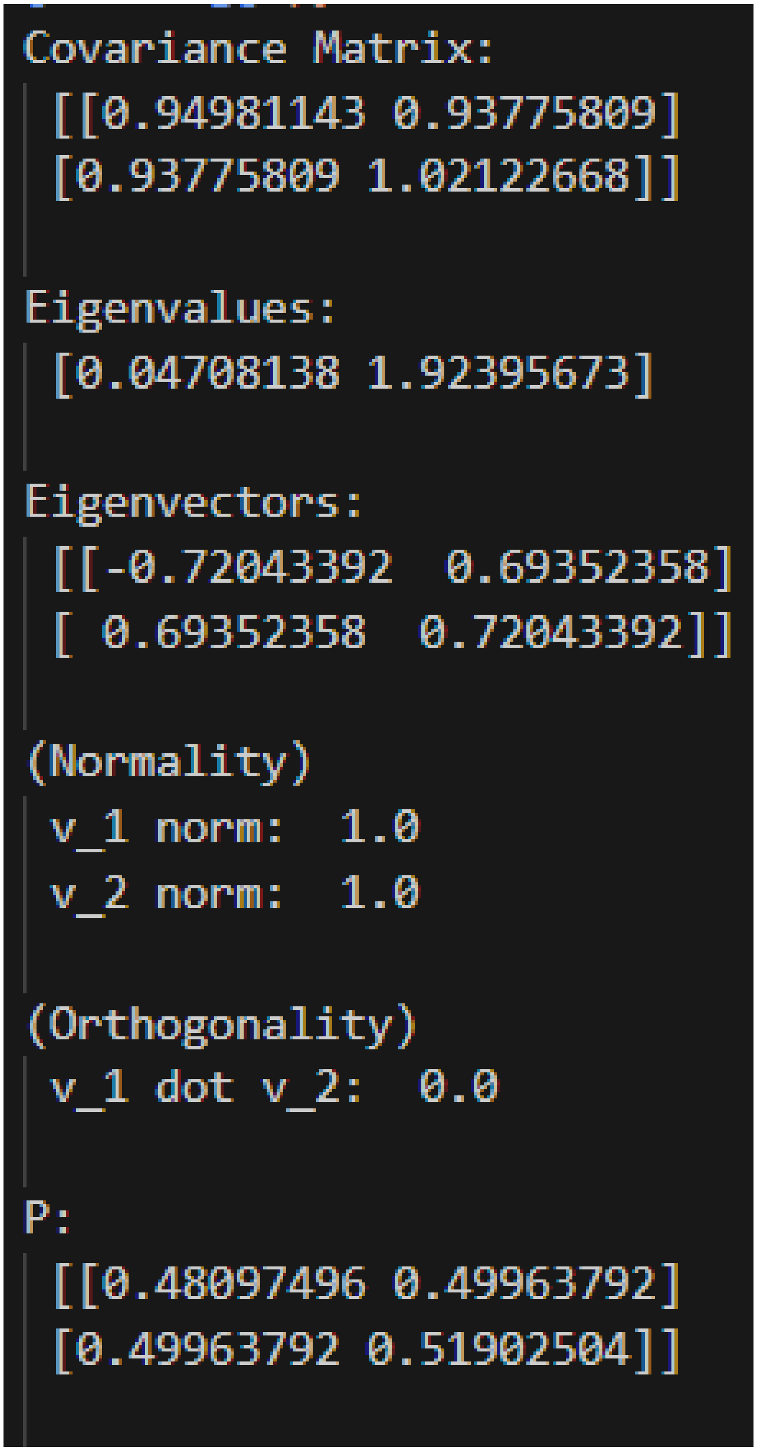


Figure 2: The blue squares present the data points $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$, and the red dots present their projections $\boldsymbol{P}\boldsymbol{x}_1, \boldsymbol{P}\boldsymbol{x}_2, \ldots, \boldsymbol{P}\boldsymbol{x}_n$ on V^* (see Eq. (2.3)).

Here is the final plot:

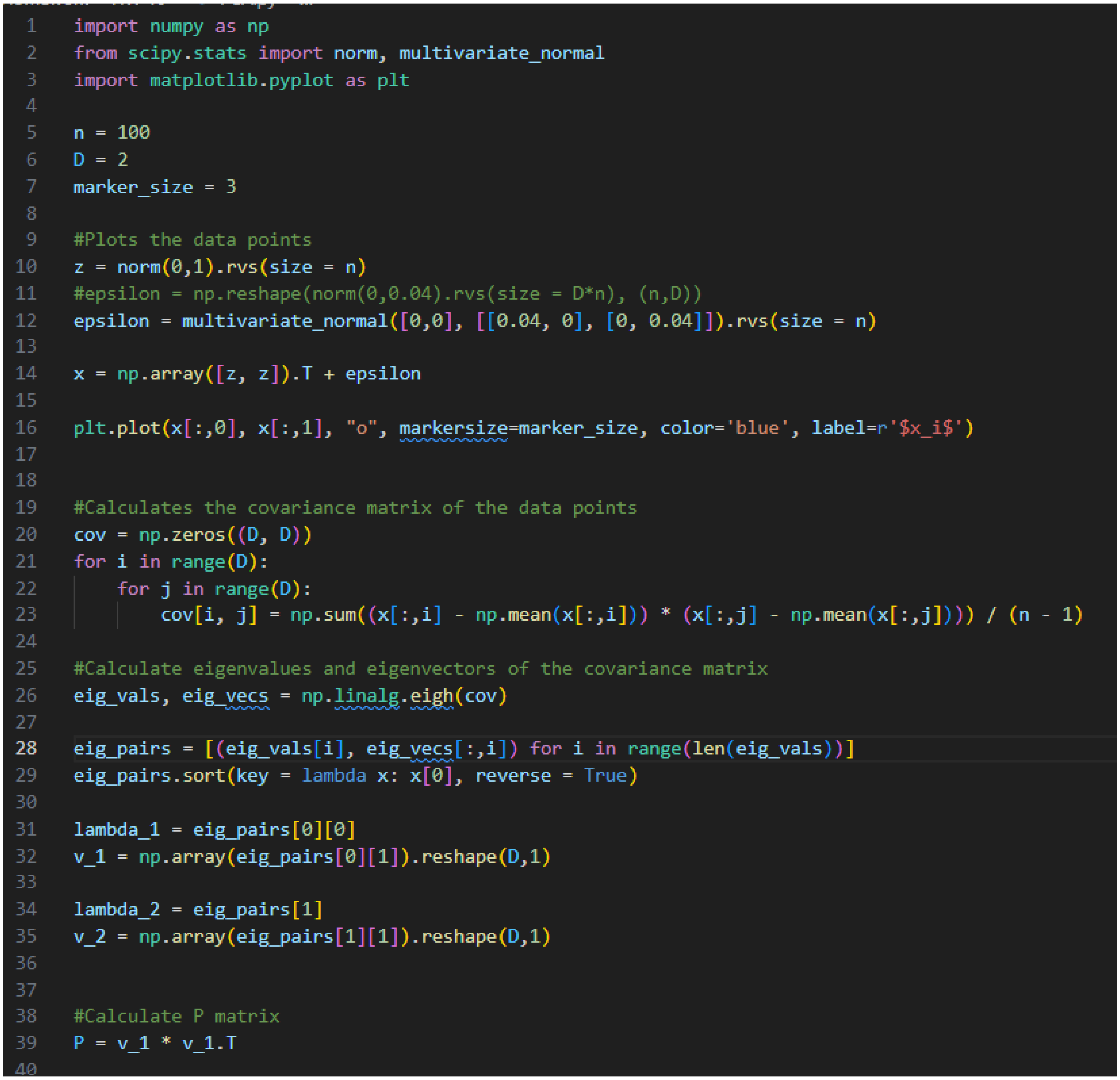


Here are the numerical results

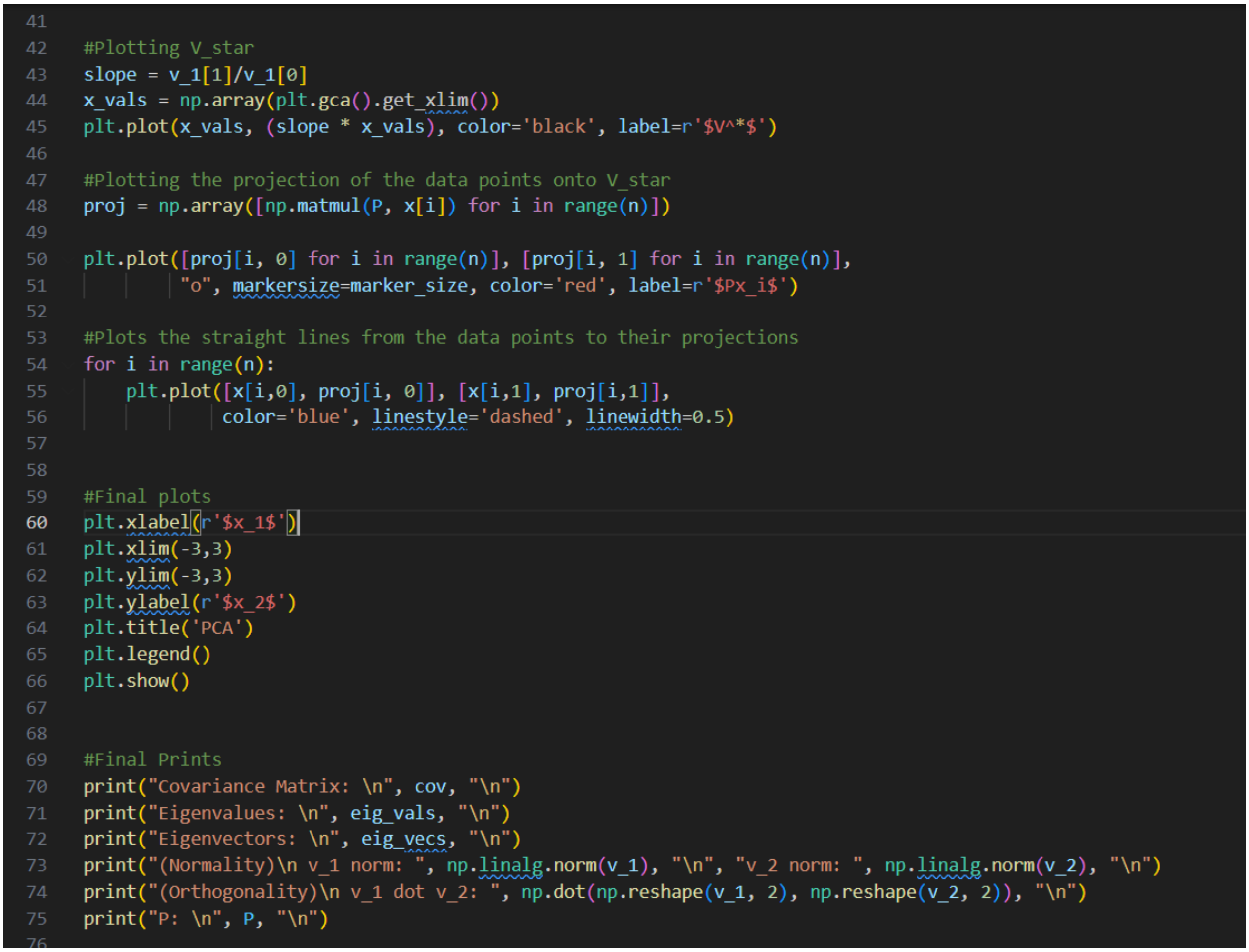


```
Covariance Matrix:
 [[0.94981143 0.93775809]
 [0.93775809 1.02122668]]

Eigenvalues:
 [0.04708138 1.92395673]

Eigenvectors:
 [[-0.72043392  0.69352358]
 [ 0.69352358  0.72043392]]

(Normality)
 v_1 norm:  1.0
 v_2 norm:  1.0

(Orthogonality)
 v_1 dot v_2:  0.0

P:
 [[0.48097496 0.49963792]
 [0.49963792 0.51902504]]
```

And here is the code I used to generate it. Note that I needed to generate the epsilon vector using the multivariate normal distribution to get a graph similar to the provided one. Generating 200 iid normal random variables and then reshaping them into a 100-by-2 matrix did work when $\varepsilon_i \sim N(0, 0.4)$ so I hypothesize the problem is simply one of scaling to be visible on the graph.



```
import numpy as np
from scipy.stats import norm, multivariate_normal
import matplotlib.pyplot as plt

n = 100
D = 2
marker_size = 3

#Plots the data points
z = norm(0,1).rvs(size = n)
#epsilon = np.reshape(norm(0,0.04).rvs(size = D*n), (n,D))
epsilon = multivariate_normal([0,0], [[0.04, 0], [0, 0.04]]).rvs(size = n)

x = np.array([z, z]).T + epsilon

plt.plot(x[:,0], x[:,1], "o", markersize=marker_size, color='blue', label=r'$x_i$')


#Calculates the covariance matrix of the data points
cov = np.zeros((D, D))
for i in range(D):
    for j in range(D):
        cov[i, j] = np.sum((x[:,i] - np.mean(x[:,i])) * (x[:,j] - np.mean(x[:,j]))) / (n - 1)

#Calculate eigenvalues and eigenvectors of the covariance matrix
eig_vals, eig_vecs = np.linalg.eigh(cov)

eig_pairs = [(eig_vals[i], eig_vecs[:,i]) for i in range(len(eig_vals))]
eig_pairs.sort(key = lambda x: x[0], reverse = True)

lambda_1 = eig_pairs[0][0]
v_1 = np.array(eig_pairs[0][1]).reshape(D,1)

lambda_2 = eig_pairs[1]
v_2 = np.array(eig_pairs[1][1]).reshape(D,1)


#Calculate P matrix
P = v_1 * v_1.T

```



```

#Plotting V_star
slope = v_1[1]/v_1[0]
x_vals = np.array(plt.gca().get_xlim())
plt.plot(x_vals, (slope * x_vals), color='black', label=r'$V^*$')

#Plotting the projection of the data points onto V_star
proj = np.array([np.matmul(P, x[i]) for i in range(n)])

plt.plot([proj[i, 0] for i in range(n)], [proj[i, 1] for i in range(n)],
         "o", markersize=marker_size, color='red', label=r'$Px_i$')

#Plots the straight lines from the data points to their projections
for i in range(n):
    plt.plot([x[i,0], proj[i, 0]], [x[i,1], proj[i,1]],
             color='blue', linestyle='dashed', linewidth=0.5)


#Final plots
plt.xlabel(r'$x_1$')
plt.xlim(-3,3)
plt.ylim(-3,3)
plt.ylabel(r'$x_2$')
plt.title('PCA')
plt.legend()
plt.show()


#Final Prints
print("Covariance Matrix: \n", cov, "\n")
print("Eigenvalues: \n", eig_vals, "\n")
print("Eigenvectors: \n", eig_vecs, "\n")
print("(Normality)\n v_1 norm: ", np.linalg.norm(v_1), "\n", "v_2 norm: ", np.linalg.norm(v_2), "\n")
print("(Orthogonality)\n v_1 dot v_2: ", np.dot(np.reshape(v_1, 2), np.reshape(v_2, 2)), "\n")
print("P: \n", P, "\n")
```

References

M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in neural information processing systems*, 14, 2001.

R. R. Coifman and S. Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1): 5–30, 2006.

R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the national academy of sciences*, 102(21):7426–7431, 2005.

C. Fefferman, S. Mitter, and H. Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

T. Hastie and W. Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84 (406):502–516, 1989.

K. Meng and A. Eloyan. Principal manifold estimation via model complexity selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 83(2):369–394, 2021.

K. Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901.

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

G. A. Seber and A. J. Lee. *Linear regression analysis*. John Wiley & Sons, 2012.

A. Smola, S. Mika, B. Schölkopf, R. Williamson, et al. Regularized principal manifolds. 2001.

J. B. Tenenbaum, V. d. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

H.-T. Wu and N. Wu. Think globally, fit locally under the manifold setup: Asymptotic analysis of locally linear embedding. 2018.