Homework 1.

Question 1.

(a) Explain why the sample mean $\hat{\mu}_i = \frac{1}{n}\sum_{j=1}^{n} X_{ji}$ is an "appropriate" estimator of the population mean μ_i.

$$\hat{\mu}_i = \frac{1}{n}\sum_{j=1}^{n} X_{ji}$$
$$E(\hat{\mu}_i) = E(\frac{1}{n}\sum_{j=1}^{n} X_{ji})$$
$$E(\hat{\mu}_i) = \frac{1}{n}\sum_{j=1}^{n} E(X_{ji})$$
$$E(\hat{\mu}_i) = \frac{1}{n} n\mu_i$$
$$E(\hat{\mu}_i) = \mu_i$$

The sample mean, $E(\hat{\mu}_i)$ is an unbiased estimator of poulation mean, μ_i, it is also a **consistent estimator** by the Law of Large Numbers, because of the consistency and unbiasedness, it is an appropriate estimator.

The sample covariance matrix $\hat{\Sigma}_{ij} = \frac{1}{n}\sum_{t=1}^{n}(X_{ti} - \hat{\mu}_i)(X_{tj} - \hat{\mu}_j)$ is appropriate because:

$$
\begin{align}
E[\hat{\Sigma}_{ij}] &= E\left[\frac{1}{n}\sum_{t=1}^{n}(X_{ti} - \hat{\mu}_i)(X_{tj} - \hat{\mu}_j)\right] \tag{1}\\
&= \frac{1}{n}\sum_{t=1}^{n} E[(X_{ti} - \hat{\mu}_i)(X_{tj} - \hat{\mu}_j)] \tag{2}\\
&\approx \frac{1}{n}\sum_{t=1}^{n} E[(X_{ti} - \mu_i)(X_{tj} - \mu_j)] \tag{3}\\
&= \frac{1}{n}\sum_{t=1}^{n} \Sigma_{ij} \tag{4}\\
&= \Sigma_{ij} \tag{5}
\end{align}
$$

Additionally, by the **Law of Large Numbers**, as $n \to \infty$:

$$\hat{\Sigma}_{ij} \to \Sigma_{ij} \text{ (convergence in probability)}$$

This makes it a **consistent estimator**.

(b) Assume that $\hat{\mu} = 0$, show that $\hat{\Sigma} = \hat{X}^T\hat{X}$

$$\hat{\Sigma}_{ij} = \frac{1}{n}\sum_{l=1}^{n}(\hat{X}_{li} - \hat{\mu}_i)(\hat{X}_{lj} - \hat{\mu}_j)$$

$$\hat{\Sigma}_{ij} = \frac{1}{n}\sum_{l=1}^{n}(\hat{X}_{li})(\hat{X}_{lj})$$

$$\hat{\Sigma}_{ij} = \frac{1}{n}\sum_{l=1}^{n}(\hat{X}^T)_{il}(\hat{X}_{lj})$$

$$\hat{\Sigma} = \frac{1}{n}\hat{X}^T\hat{X}$$

(c) Don't assume that $\hat{\mu} = 0$, show that $\hat{\Sigma} = \frac{1}{n}(\hat{X} - 1\hat{\mu}^T)^T(\hat{X} - 1\hat{\mu}^T)$

$$\hat{\Sigma}_{ij} = \frac{1}{n}\sum_{l=1}^{n}(\hat{X}_{li} - \hat{\mu}_i)(\hat{X}_{lj} - \hat{\mu}_j)$$

$$\hat{\Sigma}_{ij} = \frac{1}{n}\sum_{l=1}^{n}(\hat{X}_{li}\hat{X}_{lj}) - (\hat{\mu}_i\hat{X}_{lj}) - (\hat{\mu}_j\hat{X}_{li}) + (\hat{\mu}_i\hat{\mu}_j)$$

$$\hat{\Sigma}_{ij} = \frac{1}{n}\sum_{l=1}^{n}(\hat{X}_{li}\hat{X}_{lj}) - \frac{1}{n}\sum_{l=1}^{n}(\hat{\mu}_i\hat{X}_{lj}) - \frac{1}{n}\sum_{l=1}^{n}(\hat{\mu}_j\hat{X}_{li}) + \frac{1}{n}\sum_{l=1}^{n}(\hat{\mu}_i\hat{\mu}_j)$$

$$\hat{\Sigma}_{ij} = \frac{1}{n}\sum_{l=1}^{n}(\hat{X}_{li}\hat{X}_{lj}) - (\hat{\mu}_i\hat{\mu}_j) - (\hat{\mu}_j\hat{\mu}_i) + (\hat{\mu}_i\hat{\mu}_j)$$

$$\hat{\Sigma}_{ij} = \frac{1}{n}\sum_{l=1}^{n}(\hat{X}_{li}\hat{X}_{lj}) - (\hat{\mu}_j\hat{\mu}_i)$$

$$\hat{\Sigma} = \frac{1}{n}\hat{X}^T\hat{X} - (\hat{\mu}\hat{\mu}^T)$$

$$\hat{\Sigma} = \frac{1}{n}[(\hat{X} - \mathbf{1}\hat{\mu}^T)^T(\hat{X} - \mathbf{1}\hat{\mu}^T)]$$

$$\hat{\Sigma} = \frac{1}{n}[(\hat{X}^T - \hat{\mu}\mathbf{1}^T)(\hat{X} - \mathbf{1}\hat{\mu}^T)]$$

$$\hat{\Sigma} = \frac{1}{n}[\hat{X}^T\hat{X} - \hat{X}^T(\mathbf{1}\hat{\mu}^T) - (\hat{\mu}\mathbf{1}^T)\hat{X} + (\hat{\mu}\mathbf{1}^T)(\mathbf{1}\hat{\mu}^T)]$$

$$\hat{\Sigma} = \frac{1}{n}[\hat{X}^T\hat{X} - n\hat{\mu}\hat{\mu}^T - n\hat{\mu}\hat{\mu}^T + n\hat{\mu}\hat{\mu}^T]$$

$$\hat{\Sigma} = \frac{1}{n}[\hat{X}^T\hat{X} - n\hat{\mu}\hat{\mu}^T]$$

$$\hat{\Sigma} = \frac{1}{n}\hat{X}^T\hat{X} - \hat{\mu}\hat{\mu}^T$$

(d) Why the projection length is $|\tilde{X}^{(i)} \cdot w|$ and why $(\tilde{X}^{(i)} \cdot w)w$ is the projection vector:

Why the projection length is $|\tilde{X}^{(i)} \cdot w|$

$$\begin{aligned} \text{Let } l &= \text{length of the projection} \\ \cos(\theta) &= \frac{l}{\|\tilde{X}^{(i)}\|} \\ l &= \|\tilde{X}^{(i)}\| \cos(\theta) \\ \tilde{X}^{(i)} \cdot w &= \|\tilde{X}^{(i)}\| \|w\| \cos(\theta) \\ \tilde{X}^{(i)} \cdot w &= \|\tilde{X}^{(i)}\| \|w\| \left(\frac{l}{\|\tilde{X}^{(i)}\|} \right) \\ \tilde{X}^{(i)} \cdot w &= \|w\| \cdot l \\ \tilde{X}^{(i)} \cdot w &= l \quad (\text{since } \|w\| = 1) \end{aligned}$$

Therefore, $|\tilde{X}^{(i)} \cdot w| = |l| = l$, which is the length of the projection.

Why $(\tilde{X}^{(i)} \cdot w)w$ is the projection vector:

$$\begin{aligned} \text{We know: } \tilde{X}^{(i)} \cdot w &= l \text{ (length of projection)} && (6) \\ \text{The projection vector} &= l \cdot w && (7) \\ &= (\tilde{X}^{(i)} \cdot w)w && (8) \end{aligned}$$

The scalar $\tilde{X}^{(i)} \cdot w$ tells us how far to go, and the unit vector w tells us which direction. Multiplying them gives the projection vector.

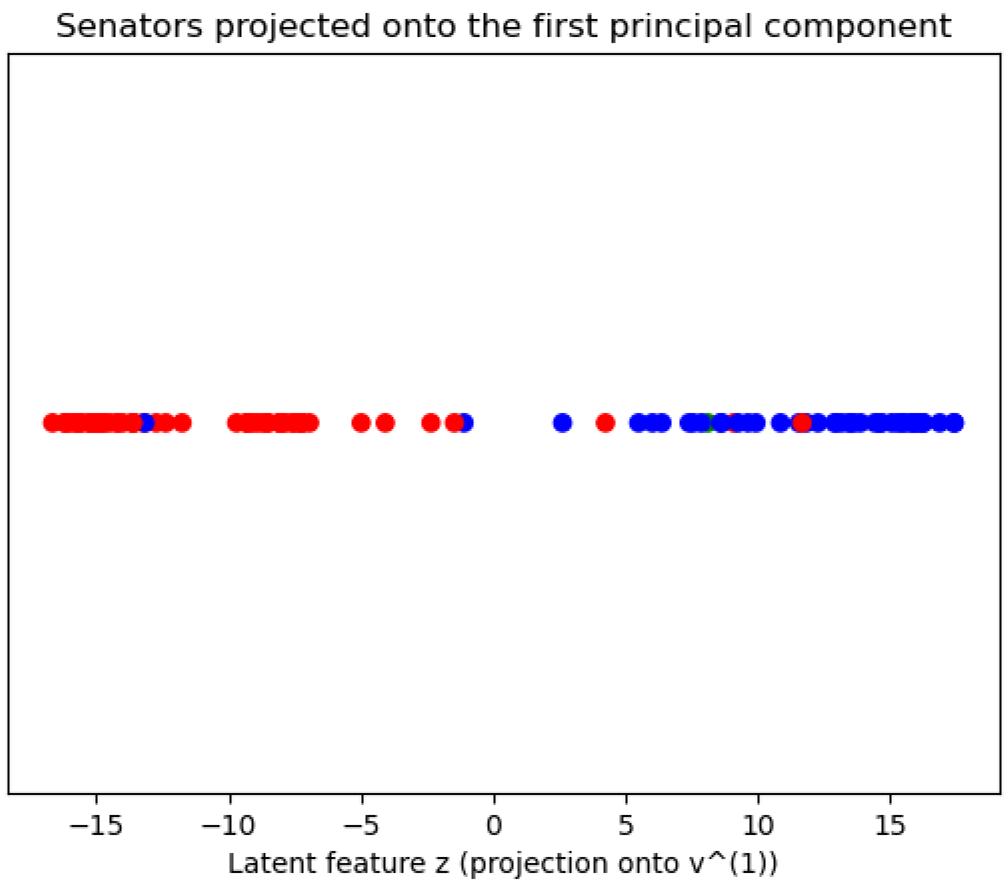
Homework 1. Senators dataset

This dataset contains information on how each senator votes on 542 different bills etc, there are 100 senators in totall; +1 means voted for, -1 means voted against and 0 means abstained. We will use **1-D PCA** to see if we can find the vector of maximum variation.

In [ ]:

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

# read in data
df = pd.read_csv('../data/senators.csv').set_index('senator')
# get the party affiliation
parties = df.index.str.split("-").str[-1]


# prep for 1-D PCA
X = df.to_numpy()

# calculate the covariance matrix
cov_matrix = np.cov(X, rowvar=False, bias=False)

# eigenvectors and eigenvalues, sorted in ascending order
# for smallest to largest eivenvalue, get the largest one
eigvals, eigvecs = np.linalg.eigh(cov_matrix)
dominant_eigenvector = eigvecs[:, -1]
print(f'The direction of maximum variance (ie the dominant eigenvector) is {dominant_eig


# calculating encoding per senator
encoding = X @ dominant_eigenvector
decoding = np.outer(encoding, dominant_eigenvector)

# Encoding: 1D → single-column DataFrame
encoding_df = pd.DataFrame(
    encoding,
    index=df.index,
    columns=["PC1_encoding"]
)

# Decoding: same shape as X → keep original feature names
decoding_df = pd.DataFrame(
    decoding,
    index=df.index,
    columns=df.columns
)

# plotting along party lines
fig, ax = plt.subplots()
colors = parties.map({'D': 'blue', 'R': 'red', 'I': 'green'})
ax.scatter(encoding, [0]*len(encoding), c=colors)
ax.set_yticks([])  # hide y-axis since it’s 1D
ax.set_xlabel("Latent feature z (projection onto v^(1))")
ax.set_title('Senators projected onto the first principal component')
ax.set_ylim(-0.05, 0.05)   # squish y-axis close to 0
```

```
plt.show()

## assessing best and worst at decoding

errors = ((df - decoding_df) ** 2).mean(axis=1)
errors.name = "MSE"

best_decoded = errors.nsmallest(5)    # 5 senators with lowest MSE
worst_decoded = errors.nlargest(5)    # 5 senators with highest MSE
```



Interpretation: what is v^i exactly?

v^i represents the direction of maximum variance, ie the direction where there is clearest separation in voting behavior. In the chart above, we see that this divide is clearest in the party affiliation. While this is not actually the party affiliation, it is quite similar as thats the direction which separates voting the clearest.

In [ ]:
```
selected_senators = ['Barack H. Obama-D', 'John S. McCain-R', 'Ted Stevens-R']

print("Selected senators and their PC1 projections:")
for senator in selected_senators:
    encoding_val = encoding_df.loc[senator, 'PC1_encoding']
    mse_error = errors.loc[senator]
    print(f"{senator}: PC1 = {encoding_val:.3f}, MSE = {mse_error:.3f}")
```

```
Selected senators and their PC1 projections:
Barack H. Obama-D: PC1 = 7.555, MSE = 0.195
John S. McCain-R: PC1 = -9.702, MSE = 0.751
Ted Stevens-R: PC1 = -13.648, MSE = 0.645
```

PC1 Projections (Ideological Positioning):

Obama (7.555): Strong positive value = liberal end of the spectrum McCain (-9.702): Moderate negative value = conservative, but not extremely so Stevens (-13.648): Most negative value = very conservative end

The PC1 clearly captures the liberal-conservative ideological axis, with positive values = Democrats/liberals and negative values = Republicans/conservatives. MSE (Reconstruction Errors):

Obama (0.195): Very low error = his voting pattern is well-explained by the main ideological dimension McCain (0.751): Highest error = his voting pattern is poorly captured by just the liberal-conservative axis Stevens (0.645): Moderate error = reasonably well-explained by ideology alone

Which senators vote along the primary axis the most often (likely interpreted as party lines)?

In [ ]:
```
best_decoded
```

Out[ ]:
```
senator
Robert Menendez-D              0.085103
Jim W. DeMint-R                0.178241
John H. 'Johnny' Isakson-R     0.182145
Mel Martinez-R                 0.193517
Barack H. Obama-D              0.195432
Name: MSE, dtype: float64
```

In [ ]:
```
worst_decoded
```

Out[ ]:
```
senator
Olympia J. Snowe-R                 0.994183
Arlen Spector-R                    0.936055
James Merrill 'Jim' Jeffords-I     0.852945
Michael 'Mike' DeWine-R            0.838235
Susan M. Collins-R                 0.814496
Name: MSE, dtype: float64
```