Name: ______________________________ Student ID: ____________________

Math 2740 – Fall 2025
Sample final examination (Variant 2)
2 hours

Instructions

- This examination has **9 exercises**.
- Show all your work. Correct answers without justification will receive little or no credit.
- You may use the back of pages if needed.
- No electronic devices (including calculators) are permitted.
- The exam is out of 130 points.

Exercise 1. [**Definitions and Core Results – 15 points**]

State the definition or theorem for each of the following. Be precise and complete.

1. [**5 pts**] State the Gram-Schmidt procedure.
2. [**4 pts**] Define a discrete-time Markov chain.
3. [**3 pts**] Define the dot product of $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$.
4. [**4 pts**] Define the *principal components* of a centered data matrix.

Exercise 2. **[Gram–Schmidt Orthonormalization – 20 points]**
Consider the vectors in $\mathbb{R}^3$:

$$\mathbf{v}_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{v}_2 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \quad \mathbf{v}_3 = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}.$$

1. [**4 pts**]Can you apply the Gram–Schmidt procedure to these vectors? Justify your answer.
2. [**4 pts**] Apply the Gram–Schmidt procedure to $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ to obtain an *orthogonal* set $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$.
3. [**4 pts**] Normalize your vectors to obtain an *orthonormal* set $\{q_1, q_2, q_3\}$.
4. [**4 pts**] Verify orthonormality by computing the inner products $\langle q_i, q_j \rangle$ for all i, j and by checking $\|q_i\| = 1$.
5. [**4 pts**] Form the matrix $Q = [q_1\ q_2\ q_3]$ and state whether Q is orthogonal (justify your answer).

Exercise 3. **[Least Squares via QR – 15 points]**

Let $A \in \mathbb{R}^{m\times n}$ have full column rank and let $A = QR$ be its *reduced* QR decomposition, where $Q \in \mathbb{R}^{m\times n}$ has orthonormal columns and $R \in \mathbb{R}^{n\times n}$ is upper triangular.

1. [**8 pts**] Using an *important theorem*, prove that the least-squares solution to $A\mathbf{x} = \mathbf{b}$ is $\tilde{\mathbf{x}} = R^{-1}Q^T\mathbf{b}$.

   Important Theorem 1 (Least Squares via QR)**.** Let $A = QR$ be a reduced QR decomposition with $Q^TQ = I$ and R upper triangular. Then the least-squares solution to $A\mathbf{x} = \mathbf{b}$ satisfies $\tilde{\mathbf{x}} = R^{-1}Q^T\mathbf{b}$ and the residual is orthogonal to $\mathrm{col}(A)$.

2. [**7 pts**] For

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & -1 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 2 \\ 1 \\ 1 \\ 0 \end{pmatrix},$$

   compute the reduced QR decomposition $A = QR$ (you may use Gram–Schmidt on the columns) and find $\tilde{\mathbf{x}}$.

Exercise 4. **[Singular Value Decomposition – 15 points]**

Consider

$$A = \begin{pmatrix} 3 & 0 \\ 0 & 2 \\ 0 & 0 \end{pmatrix}.$$

1. [**8 pts**] Compute the full singular value decomposition $A = U\Sigma V^T$ of A. Show your work by computing $A^T A$, its eigenvalues and eigenvectors, then construct V, Σ, and U.
2. [**3 pts**] What is the rank of A?
3. [**4 pts**] Compute the Moore-Penrose pseudoinverse A^+ using the SVD.

Exercise 5. **[PCA on Centered Data – 10 points]**

Let the centered data matrix be

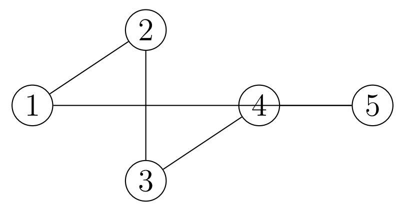
$$\tilde{X} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \\ 1 & -1 \\ 0 & 0 \end{pmatrix}.$$

1. [**6 pts**] Compute the covariance matrix $S = \frac{1}{n-1}\tilde{X}^T\tilde{X}$ and its eigenvalues/eigenvectors.
2. [**4 pts**] Identify the first principal component and the variance explained by it.

Exercise 6. **[Graph Measures I – 12 points]**

Consider the simple undirected graph G on vertices $V = \{1, 2, 3, 4, 5\}$ with edge set

$$E = \{\{1,2\}, \{1,5\}, \{2,3\}, \{3,4\}, \{4,5\}\}.$$



1. **[4 pts]** Compute the degree $\deg(i)$ of each vertex and give the degree sequence in nonincreasing order.
2. **[4 pts]** Compute the density of G, defined as $\delta(G) = \dfrac{2|E|}{|V|(|V|-1)}$.
3. **[4 pts]** Compute the local clustering coefficient C_i for each vertex with $\deg(i) \geq 2$ and state the average clustering coefficient.

Exercise 7. **[Graph Measures II – 13 points]**

For the same graph G as in Exercise 6:

1. **[5 pts]** Compute the graph diameter and the average shortest-path length $\ell(G)$.
2. **[4 pts]** Compute the (normalized) degree centrality of each vertex, $C_D(i) = \deg(i)/(n-1)$ where $n = |V|$.
3. **[4 pts]** Compute the closeness centrality of each vertex, $C_C(i) = \dfrac{n-1}{\sum_{j \neq i} d(i,j)}$.

Exercise 8. **[Absorbing Markov Chains – 20 points]**

Consider a Markov chain with four states $\{1, 2, 3, 4\}$ and column-stochastic transition matrix:

$$P = \begin{pmatrix} 1 & 0.3 & 0.2 & 0 \\ 0 & 0.4 & 0.1 & 0 \\ 0 & 0.2 & 0.5 & 0.5 \\ 0 & 0.1 & 0.2 & 0.5 \end{pmatrix},$$

where P_{ij} is the probability of moving from state j to state i.

1. **[4 pts]** Draw the directed graph representation of this Markov chain, showing all states and transition probabilities on the edges.
2. **[3 pts]** Identify which states are absorbing and which are transient. Justify your answer.
3. **[3 pts]** Explain why this Markov chain is classified as an absorbing Markov chain.
4. **[5 pts]** Reorder the states (if necessary) to write P in canonical form

$$P = \begin{pmatrix} I & 0 \\ R & Q \end{pmatrix}$$

and identify the matrices I, R, and Q.

5. **[5 pts]** Compute the fundamental matrix $N = (I - Q)^{-1}$. Interpret what the entries N_{ij} represent.

Exercise 9. [Reading R Code – 10 points]

What does the following function do? Explain your answer by describing the algorithm and its purpose. You do not need to carry out a numerical run, but you should identify what mathematical operation is being performed.

```
mystery_function <- function(A, tol = 1e-10) {
  if (!is.matrix(A)) stop("A␣must␣be␣a␣matrix")

  m <- nrow(A)
  n <- ncol(A)

  M1 <- matrix(0, nrow = m, ncol = n)
  M2 <- matrix(0, nrow = n, ncol = n)

  for (j in 1:n) {
    v <- A[, j]

    if (j > 1) {
      for (i in 1:(j-1)) {
        M2[i, j] <- sum(M1[, i] * A[, j])
        v <- v - M2[i, j] * M1[, i]
      }
    }

    M2[j, j] <- sqrt(sum(v^2))

    if (M2[j, j] < tol) {
      stop(sprintf("Column␣%d␣is␣linearly␣dependent␣on␣previous␣columns", j))
    }

    M1[, j] <- v / M2[j, j]
  }

  return(list(M1 = M1, M2 = M2))
}
```

END OF EXAMINATION