

Assignment 2: Linear Algebra, Numpy, and PCA

1 Instructions

In this assignment, you will make some practice exam questions (Multiple Choice or Select All) based on the lecture materials on Linear Algebra, Numpy, and Principal Component Analysis (PCA). Similar to previous assignment, prepare a presentation in Quarto markdown, following the requirements below. Briefly, you will construct the following practice exam questions and answers.

1. Two math questions on Linear Algebra
2. Two programming questions on Numpy and/or implementing Linear Algebra operations in Numpy.
3. Two conceptual, programming or math questions on Principal Component Analysis (PCA).

Before you begin:

- Make sure you have a solid understanding of the lecture materials.
- Focus on the key concepts rather than getting caught up in less important details.
- If there are parts you don't fully understand, you're not alone—use Piazza to ask questions and reach out to the TAs for help.
- You can also revisit the previous lecture videos in the Kaltura Media Gallery on Brightspace.

2 Content Requirements

2.1 Slide Organization

You must submit the following **12 slides**:

1. Math Linear Algebra Question 1
2. Math Linear Algebra Question 2
3. Numpy Question 1
4. Numpy Question 2
5. PCA Question 1
6. PCA Question 2
7. Answer and Explanation for Q1
8. Answer and Explanation for Q2
9. Answer and Explanation for Q3
10. Answer and Explanation for Q4
11. Answer and Explanation for Q5
12. Answer and Explanation for Q6

2.2 Slide Content

Question slides: Each question slide should have a **multiple choice** with 3-5 choices or **select-all** question with 3 answer options. *Do not include the answer on this slide so that this can be used for practice.* The question should be clear and concise. Additionally, the question should be **valid**: For multiple choice questions, this means that there is only one correct answer and all other answers are wrong. For select-all questions, each option must only have one correct answer (i.e., marked or not marked)—if an option in a select all could be correct in some situations but not correct in others, then the question is invalid.

Answer slides: Each answer slide should provide three main bullets: **Correct Answer**, **Explanation**, and **Why this question matters**. The explanation should justify why each answer choice is true or false with a brief one sentence explanation. The *why this question matters* part should explain why this is a good practice question and relevance to topics in class.

3 Example Questions

3.1 Example Math Question

Linear Algebra Question 1

Assume A, B, C, D are 3×3 real matrices. Which of the following statements are **ALWAYS true**?

- [] A is not singular if and only if $\text{rank}(A) = 3$
- [] $\text{Tr}(ABCD) = \text{Tr}(CDAB)$
- [] $(AB)^T = A^T B^T$
- [] $A(B+C)^T = AB^T + C^T A$

Answer and Explanation for Linear Algebra Question 1

- **Correct Answer:**
 - Statements **(1)** and **(2)** are always true.
- **Explanation:**
 - **(1)** True. A 3×3 matrix is invertible (non-singular) precisely when it has full rank (i.e., $\text{rank} = 3$).
 - **(2)** True. The trace is invariant under cyclic permutations:
 $\text{Tr}(XYZW) = \text{Tr}(ZWXY)$.
 - **(3)** False. The correct transpose of a product is $(AB)^T = B^T A^T$, **not** $A^T B^T$.
 - **(4)** False. Since $(B+C)^T = B^T + C^T$, then $A(B+C)^T = A(B^T + C^T) = AB^T + AC^T$, **not** $AB^T + C^T A$.
- **Why This Question Matters:**
 - This question directly tests knowledge of essential concepts from the Linear Algebra such as **Matrix singularity and rank**, **Trace properties**, and

```
**Transpose and product rules**.
```

3.2 Example Programming Question

```
## Numpy Programming Question 1
```

Assume NumPy arrays `x`` and `y`` are multiplied element-wise via `x * y``, where `S_x`` and `S_y`` are the shapes of `x`` and `y``, respectively. For which pairs `(S_x, S_y)`` will NumPy throw an ****error****?

- [] (1) `S_x = (1, 10), S_y = (10, 10)``
- [] (2) `S_x = (10, 1), S_y = (10, 1)``
- [] (3) `S_x = (10, 100), S_y = (100, 10)``
- [] (4) `S_x = (1, 10, 100), S_y = (1, 1, 1)``

```
## Answer and Explanation for Numpy Programming Question 1
```

- ****Correct Answer:****
 - ****3)**** - Numpy throws an error for `S_x = (10, 100)`` and `S_y = (100, 10)``.
- ****Explanation:****
 - ****1)** Works:**** `(1, 10)`` broadcasts across the first dimension to match `(10, 10)``.
 - ****2)** Works:**** `(10, 1)`` against `(10, 1)``-identical shapes-multiply fine.
 - ****3)** Error:**** `(10, 100)`` and `(100, 10)`` are incompatible; neither trailing nor leading dimensions match nor equal 1.
 - ****4)** Works:**** `(1, 1, 1)`` easily broadcasts to `(1, 10, 100)``.
- ****Why This Question Matters:****
 - This question directly tests the ability to understand Numpy broadcasting rules.

3.3 Example PCA Question

```
## PCA Question 1
```

Which of the following are equivalent PCA problems (select all)?

- Minimizing reconstruction error
- Minimizing variance of latent projection
- Maximizing variance of latent projection
- Maximizing reconstruction error

```
## Answer and Explanation for PCA Question 1
```

- **Correct Answer:**

- Statements **(Minimizing reconstruction error)** and **(Maximizing variance of latent projection)** are equivalent PCA formulations.

- **Explanation:**

- **(1) Minimizing reconstruction error**

PCA finds the low-dimensional subspace that best reconstructs the data. Minimizing this error is the standard formulation.

- **(2) Minimizing variance of latent projection**

PCA seeks directions with *maximum* variance, not minimum. Minimizing variance would collapse information.

- **(3) Maximizing variance of latent projection**

Equivalent to minimizing reconstruction error. PCA projects data into directions with maximal variance.

- **(4) Maximizing reconstruction error**

The opposite of PCA's goal.

- **Why This Question Matters:**

- This tests the students ability to understand that

```
PCA admits two equivalent, yet conceptually distinct formulations:  
**Minimizing reconstruction error** and  
**Maximizing variance of the latent projection**.
```

3.4 Format

Plain text in [Quarto RevealJS markdown format](#) **without the YAML header** containing the slides outlined above. You can use more advanced RevealJS markdown like two column layouts if helpful. Any images used must have absolute URLs so that I can copy and paste into my lecture slides.

You can use the following YAML header to test the visualization of your slides but please don't include the header in the submission:

```
---  
format:  
  revealjs:  
    width: 1920  
    height: 1080  
    scrollable: true  
---
```

3.5 Character Limit

9000 (Any content after the character limit will be truncated during grading)

4 Rubric

Criterion	Excellent (5)	Good (4)	Satisfactory (3)	Needs Improvement (2)	Unsatisfactor (1)	Weight
Completion & Correctness	Fully meets assignment requirements (6 questions with 3-5 answer options, 6 answer explanations). All questions/answers are clear, unambiguous, and correct.	Minor issues in correctness (e.g., 1 missing element or ambiguity).	Most requirements met but with 2+ omissions or vague/partially correct answers.	Significant omissions (e.g., several missing slides or explanations).	Does not follow instructions or mostly incorrect.	25%
Relevance to Lectures	Tests key concepts from Linear Algebra, NumPy, and PCA lecture content; rationale shows strong connection.	Mostly relevant, with generally good rationales.	Loosely relevant; rationales are weak or generic.	Minimal connection to lecture content.	Unrelated to assigned lectures.	25%
Conciseness & Clarity	Slides are concise, clear, and insightful well-calibrated in difficulty: challenging but not trivial or overly advanced.	Mostly concise and at the right level, with minor issues.	Understandable but somewhat wordy, or questions occasionally too easy/hard.	Often verbose or cluttered, or difficulty poorly calibrated.	Very hard to follow and consistently miscalibrated in difficulty.	25%
Quality of Explanations	Explanations show strong understanding, highlight misconceptions, and add learning value.	Correct and useful, but lack depth.	Technically correct but minimal.	Incomplete or partially incorrect.	Missing or mostly incorrect.	25%