HOMEWORK 1

PSTAT 100 - DATA SCIENCE: CONCEPTS AND ANALYSIS

INSTRUCTOR: Ethan P. Marzban

Submission Instructions

This homework assignment consists of a mix of written and coding questions.

Written Portion

- Please show all of your work
- Answers may be handwritten or typeset (using LaTeX, Word, etc.)
- Please write legibly; if the grader cannot read your work, you will not receive full marks.

Coding Portion

- Please make sure to interpret **all** code outputs.
  - As a general rule-of-thumb: if there is a code chunk whose output is not being interpreted, you should move the code chunk to an Appendix.

Final Submission

- You should combine your written and coding answers into a **single** PDF, which you upload to Gradescope.
  - Here is a free online resource to help you merge PDFs.
  - Please note: Gradescope will only allow you to upload a single PDF.
- Ensure you match pages in your Gradescope submission; failure to do so may incur point penalties.

Due Date

You must upload your homework to Gradescope by no later than **11:59 pm on Sunday, June 29, 2025**.

Information on Grading

- A handful of parts will be selected from this homework to be graded on correctness; these parts will be graded collectively out of 12 points.
  - We will not reveal which parts are to be graded upon correctness until after the homework is graded, so please attempt all problems!
- You will be assigned 2 additional points for submitting the *entirety* of your homework, and 1 additional point for matching pages on your gradescope submission.
  - As such, if you fail to submit an attempt for all parts and fail to match pages, you will not receive anything above an 80%.

Written Portion

Problem 1: Some Linear Algebra Results

Motivation

We will frequently leverage commonly-derived results from Linear Algebra, as well as the *techniques* used to derive these results. This problem is designed to not only introduce three useful results, but also give you practice with using eigenvalue decompositions and singular value decompositions to derive results - a technique we will utilize in lecture next week.

The following parts do not depend on one another.

(a) Suppose $\mathbf{A}$ is a diagonalizable matrix. Show that the trace of $\mathbf{A}$ is equal to the sum of its eigenvalues. **Hint:** Consider the eigenvalue decomposition (EVD) of $\mathbf{A}$.

(b) Recall that a square matrix $\mathbf{A}$ is said to be **idempotent** if $\mathbf{A}^2 = \mathbf{A}$. Show that the eigenvalues of an idempotent matrix are either 0 or 1. **Hint:** idempotent matrices are always diagonalizable.

(c) Let $\mathbf{A}$ be a matrix (not necessarily square). Show that the trace of $\mathbf{A}^\mathsf{T}\mathbf{A}$ is equal to the sum of the squares of the singular values of $\mathbf{A}$.

Problem 2: Random Vectors

Motivation

In data science, we are often concerned with *multiple* values that may be random. As such, we need to extend the notion of a random variable defined in PSTAT 120A: this problem introduces you to such an extension.

Recall from PSTAT 120A that a **random variable** *X* is essentially a mapping from an outcome space Ω to the real line. In this way, we can view a *collection* $\vec{\boldsymbol{X}} := (X_1, \cdots, X_n)^\mathsf{T}$ of *n* random variables as a mapping from an *n*-dimensional outcome space to $\mathbb{R}^n$. Such a mapping is called a **random vector**. We define the **mean (vector)** of $\vec{\boldsymbol{X}}$ to be

$$\vec{\boldsymbol{\mu}} := \mathbb{E}[\vec{\boldsymbol{X}}] := \begin{pmatrix} \mathbb{E}[X_1] \\ \vdots \\ \mathbb{E}[X_n] \end{pmatrix}$$

and the **covariance matrix** (sometimes called the **variance-covariance matrix**) of $\vec{\boldsymbol{X}}$ to be

$$\boldsymbol{\Sigma} = \mathbb{E}\left[(\vec{\boldsymbol{X}} - \vec{\boldsymbol{\mu}})(\vec{\boldsymbol{X}} - \vec{\boldsymbol{\mu}})^\mathsf{T}\right] = \begin{pmatrix} \mathsf{Var}(X_1) & \mathsf{Cov}(X_1, X_2) & \cdots & \mathsf{Cov}(X_1, X_n) \\ \mathsf{Cov}(X_1, X_2) & \mathsf{Var}(X_2) & \cdots & \mathsf{Cov}(X_2, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathsf{Cov}(X_1, X_n) & \mathsf{Cov}(X_2, X_n) & \cdots & \mathsf{Var}(X_n) \end{pmatrix}$$

to be the **mean vector** and **covariance matrix** of $\vec{\boldsymbol{X}}$, respectively.

(a) Consider an n-dimensional random vector $\vec{\boldsymbol{X}}$ where the elements of $\vec{\boldsymbol{X}}$ are i.i.d. (independent and identically distributed) with common mean μ and common variance σ^2. Write down the mean vector $\vec{\boldsymbol{\mu}}$ and covariance matrix $\boldsymbol{\Sigma}$ of $\vec{\boldsymbol{X}}$.

(b) Show that the covariance matrix of a random vector is always positive semidefinite. Recall that a matrix $\mathbf{A}$ is said to be positive semidefinite if, for every (conformable) vector $\vec{\boldsymbol{x}}$, we have $\vec{\boldsymbol{x}}^{\mathsf{T}}\mathbf{A}\vec{\boldsymbol{x}} \geq 0$. A fact you may use without proof: for a random vector $\vec{\boldsymbol{Z}}$ and a vector $\vec{\boldsymbol{a}}$ of constants,

$$\vec{\boldsymbol{a}}^{\mathsf{T}}\mathbb{E}[\vec{\boldsymbol{Z}}\vec{\boldsymbol{Z}}^{\mathsf{T}}]\vec{\boldsymbol{a}} = \mathbb{E}[\vec{\boldsymbol{a}}^{\mathsf{T}}\vec{\boldsymbol{Z}}\vec{\boldsymbol{Z}}^{\mathsf{T}}\vec{\boldsymbol{a}}]$$

Coding Portion

Problem 1: Data Science Prospects

Motivation

In this problem, we'll gain some practice with data manipulation and plotting in `R` using the `tidyverse`. I encourage you to use this as practice with:

- Filtering, mutating, pivoting, and melting dataframes
- Generating plots using `ggplot2`
- Interpreting plots to draw conclusions about data

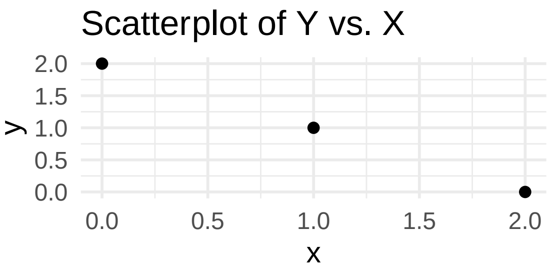
Many of you are on track to get swanky new Data Science jobs after you graduate… So let's take a look at some of your job prospects! Specifically, we'll consider a dataset containing information on the annual salaries of different data-science-focused jobs. The source for this dataset, along with a (pretty well-documented) data dictionary, can be found at **this** link.

Two Important Instructions:

- For each question, please initially provide only your code outputs (remember that you can always change the code chunk options to ensure that only the output of each code chunk is displayed), and interpret your outputs thoroughly.
- Then, include an "Appendix" including the code you used. **Please note:** the grader reserves the right to *not* explore your appendix thoroughly, so you should not include any answers to questions in the Appendix.

Example Question: What is the relationship between `x <- c(0, 1, 2)` and `y <- c(2, 1, 0)`?

Example Answer:



From this plot, we see that as `x` increases `y` decreases at a constant rate, meaning there is a negative linear association between `x` and `y`.

Appendix:

```
data.frame(x = c(0, 1, 2), y = c(2, 1, 0)) %>%
  ggplot(aes(x = x, y = y)) +
  geom_point() +
  ggtitle("Scatterplot of Y vs. X") +
  theme_minimal()
```

Note

Part of the Data Science learning process is Googling! As such, it is the intention that, for some of these questions, you may need to look up the help file for some functions not discussed in lecture, or consult Google for help on how to accomplish a certain goal. Again, not only is there no shame in this - this is *expected*! Just please be sure to cite your sources (even just including a link is sufficient).

Part I: Exploring the Dataset

We'll start out with some basic explorations of the dataset.

a) Navigate to the source for the dataset (linked above), and take a look through the data dictionary. Pick three variables, and write them down in a **bulleted list** along with a short description of what they represent.

b) Load the data (stored in a file called `salaries.csv`, located in the `data/` subfolder) R. Isolate the job titles present in the dataset, and determine the number of *unique* job titles present in the dataset.

Part II: Data Scientists

Now, let's take a look at only the job whose titles include the phrase "Data Scientist". If you have a vector `unique_job_titles` that stores the names of the unique job titles present in the dataset, then the following code chunk will extract out only the job titles containing the phrase "Data Scientist" and assign the resulting vector of job titles to a vector called `ds_titles`.

```
ds_titles <- unique_job_titles[str_detect(
  unique_job_titles, "Data Scientist")]
```

(Don't worry too much about the details behind how this code works; we'll talk about string manipulation a bit later in this course.)

c) Generate an appropriate plot (it's up to you to figure out the best type of plot here!) that displays the different job titles containing the phrase "Data Scientist" on the horizontal axis and the corresponding salaries on the vertical axis. Include descriptive axis labels, as well as a title for your plot. **Hint:** start by filtering the `salaries` dataframe to include only job titles in the `ds_title` vector, and then pipe the result into an appropriate call to `ggplot()`. **Note:** your final plot may look messy; **that's by design!** We'll work on formatting this plot in the next part.

d) Re-do your plot from part (c), but now apply a log transformation to the vertical axis (i.e. instead of plotting raw salaries, plot log-transformed salaries). Additionally, add a call to `theme()` with the appropriate arguments specified to rotate the text on the horizontal axis 90 degrees. **Hint:** Look up the help file for `theme()`, and the help file for `element_text()`.

Part III: Comparisons Over Time

Let's take a look at some comparisons over time! Since there are so many job titles represented in the dataset, we'll restrict ourselves to considering only the job titles: "Data Analyst", "Data Scientist", and "Machine Learning Engineer."

e) Plot the median salary over time, and color by job title. Again, **it is up to you to identify the most appropriate type of plot.** Then, provide a brief interpretation of your plot. Specifically, have any jobs seen an increase in median salary over time? How have the median salaries across these three job titles compared, and how has that comparison changed over time? **Tip:** To relabel your legend (assuming you have correctly colored by job title), add the following call to your code: `labs(colour = "Job Title")`.