Math 2740 – Fall 2021
Assignment 3

~~Due **Friday** 1 October at 10:00~~

General instructions/remarks

- Note the unusual due *day*.
- This assignment relies on material in the Linear Algebra review as well as the slide on projections in Slides 06.
- This assignment consists of **two** parts:
  1. the *written* part consists of long form answers to mathematical problems;
  2. the *computer* part is answered using a `jupyter notebook`.
- Return the solutions to these problems as **two** files on UMLearn. See details in the individual parts. Two folders are available to submit your solutions, one for each assignment type.
- Late assignments will not be accepted and will be given a mark of **zero**. (The system is set to stop accepting submissions after the deadline and email submissions will be ignored.)
- The mark for Assignment 3 will be a percentage mark consisting of the average of the marks obtained for the written and computer parts.
- You must complete the assignment BY YOURSELF. Acts of academic dishonesty will be subject to academic discipline.

Written part

Instructions for the written part

- Return a single pdf file for the written part, indicating clearly on the first page your name, student number and the tutorial section you are registered in.
- Ensure the file you submit is legible, that all pages appear in the correct orientation, etc. Marks will be deducted for illegible submissions.
- Full marks are given to solutions that are not only correct but also are explained with sufficient amount of detail.
- It is possible that not all problems will be marked, but if you have not provided a solution for a problem that is marked, you will receive a **zero** for that question.

Problem 1. Consider the vectors

$$\mathbf{u} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \mathbf{v} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \mathbf{w} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Compute $\mathsf{span}(\mathbf{u}, \mathbf{v}, \mathbf{w})$. Show that $\{\mathbf{u}, \mathbf{v}, \mathbf{w}\}$ is not a linearly independent set nor a basis for $\mathbb{R}^2$.

Problem 2. Let $\mathcal{M}_2(\mathbb{R})$ be the set of square 2×2 matrices with real entries. Show that $\mathcal{M}_2(\mathbb{R})$ is a vector space.

Problem 3. In this problem, we assume known that $\mathcal{M}_2(\mathbb{R})$ is a vector space, as established in Problem 2. Consider the matrices

$$E_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, E_2 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, E_3 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, E_4 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

Show that $\{E_1, E_2, E_3, E_4\}$ form a linearly independent set. Show that $\{E_1, E_2, E_3, E_4\}$ spans $\mathcal{M}_2(\mathbb{R})$. Conclude that $\{E_1, E_2, E_3, E_4\}$ forms a basis for $\mathcal{M}_2(\mathbb{R})$. What is the dimension of $\mathcal{M}_2(\mathbb{R})$?

Problem 4. In this problem, we assume known that $\mathcal{M}_2(\mathbb{R})$ is a vector space, as established in Problem 2. We also admit without proof the following theorem.

Theorem 1. Let V be a vector space over $\mathbb{R}$ and $W \subset V$ be a subset of V. W is a *subspace* of V if the following two conditions hold true.

1. For all $\mathbf{w} \in W$ and all $c \in \mathbb{R}$, $c\mathbf{w} \in W$. [W is *closed under scalar multiplication.*]

2. For all $\mathbf{w}_1, \mathbf{w}_2 \in W$, $\mathbf{w}_1 + \mathbf{w}_2 \in W$. [$W$ is *closed under addition.*]

Consider the subset $\mathcal{U} \subset \mathcal{M}_2(\mathbb{R})$ consisting of upper triangular matrices, i.e.,

$$\mathcal{U} = \left\{ \begin{pmatrix} a & b \\ 0 & c \end{pmatrix}, \quad a, b, c \in \mathbb{R} \right\}.$$

Show that $\mathcal{U}$ is a subspace of $\mathcal{M}_2(\mathbb{R})$. Find a basis for U.

Problem 5. Show that additionally to being a subspace of $\mathcal{M}_2(\mathbb{R})$ as established in Problem 4, the set of upper triangular matrices

$$\mathcal{U} = \left\{ \begin{pmatrix} a & b \\ 0 & c \end{pmatrix}, \quad a, b, c \in \mathbb{R} \right\}$$

has the property that for all $U_1, U_2 \in \mathcal{U}$, $U_1 U_2 \in \mathcal{U}$.

Computer part

Instructions for the computer part

- Return a `jupyter notebook` (.ipynb) file.
- Recall that the notebook **must** use `R`.
- The first cell in the notebook should be a `markdown` cell with your name, student number and tutorial section you are registered in indicated.
- Please make use of the capacity to insert `markdown` text in notebooks. In the same way as long form answers need some explanations to be worth full marks, I will need to be able to be told what you are doing in order for your code to receive full.

Create a `jupyter notebook` with three functions `is.linearly_independent`, `proj_onto`, `perp_along` with the following characteristics.

1. `is.linearly_independent` takes as argument a list of vectors and returns `TRUE` if the list of vectors is a linearly independent set and `FALSE` if not. The function also returns `FALSE` if there is an issue, e.g., if two of the vectors are of incompatible sizes.
2. `proj_onto` takes two arguments: a vector `v` and a list of vectors `W`. It first checks that `W` is a linearly independent set. If it is not or if the set consists of vectors of incompatible sizes, it returns the value `FALSE`. If `W` is linearly independent, it returns the vector $\mathsf{proj}_W(\mathbf{v})$.
3. `perp_along` takes two arguments: a vector `v` and a list of vectors `W`. It first checks that `W` is a linearly independent set. If it is not or if the set consists of vectors of incompatible sizes, it returns the value `FALSE`. If `W` is linearly independent, it returns the vector $\mathsf{perp}_W(\mathbf{v})$.

Apply these functions to several examples.

Some remarks/hints.

- There is a wonderful function in `R`, `lapply`, that applies a given function to all elements in a list. Suppose I have a list `l = list(c(1,2), c(1,2,3), c(1,2,3,4))`. If I want to compute the mean of each vector in this list, I can use `lapply(l, mean)`, giving me a list with the value of `mean` for each vector in `l`. If I further wanted to turn this result list into a vector (which is more convenient for some operations), I could for instance do `unlist(lapply(l, mean))`. I will add that there is a function `length` that returns the length of a list or of a vector and let you connect the dots in terms of checking that the argument to `is.linearly_independent` does not consist of incompatible vectors.

- To check for linear independence, there are several ways. Since computers do not really care about doing more work, pairwise comparison is an easy way. To generate all combinations of two indices, the function `combn` is useful. For instance, `combn(1:5, 2)` generates a matrix that looks something like this:

1	1	1	1	2	2	2	3	3	4
2	3	4	5	3	4	5	4	5	5

  So if I were to compare entries in the provided list of vectors at the positions indicated in each column..

- Build on existing functions: the two projection functions use the function that checks for linear independence; the second projection function uses the first one.