Problem Set 1: Matrix Algebra Review

Stat 154, Spring 2018, Prof. Sanchez

Due date: Fri Fep-02 (before midnight)

The purpose of this assignment is to apply in R some of the introductory material, giving you the opportunity to do some work with matrices and vectors.

Use an R markdown (`.Rmd`) file to write your code and answers. You can *knit* the `Rmd` file as html or pdf. Please submit both your `Rmd` and knitted file to bCourses. Make sure to include your name, and your lab section. If you haven't, please take some time to review the policies about the HW assignements:

https://github.com/ucb-stat154/stat154-spring-2018/blob/master/syllabus/policies.md#assignments

Problem 1 (10 pts)

Create the following matrices in R (and display them).

$$\mathbf{X} = \begin{bmatrix} 2 & 3 \\ -1 & 4 \end{bmatrix}; \qquad \mathbf{Y} = \begin{bmatrix} 2 & 0 & 1 \\ 1 & -2 & 3 \end{bmatrix}; \qquad \mathbf{Z} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \\ 0 & 2 \end{bmatrix}; \qquad \mathbf{W} = \begin{bmatrix} 1 & 0 \\ 8 & 3 \end{bmatrix}; \qquad \mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Problem 2 (10 pts)

Use the matrices created in Problem 1 to perform each of the following operations in R. If the indicated operation cannot be performed, explain why.

a. $\mathbf{X} + \mathbf{Y}$

b. $\mathbf{X} + \mathbf{W}$

c. $\mathbf{X} - \mathbf{I}$

d. $\mathbf{XY}$

e. $\mathbf{XI}$

f. $\mathbf{X} + (\mathbf{Y} + \mathbf{Z})$

g. $\mathbf{Y}(\mathbf{I} + \mathbf{W})$

Problem 3 (10 pts)

Determine whether the following statements are True or False.

a. Every orthogonal matrix is nonsingular.

b. Every nonsingular matrix is orthogonal.

c. Every matrix of full rank is square.

d. Every square matrix is of full rank.

e. Every nonsingular matrix is of full rank.

Problem 4 (10 pts)

Let $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ be conformable. Using the properties of transposes, prove that:

$$(\mathbf{XYZ})^{\mathsf{T}} = \mathbf{Z}^{\mathsf{T}}\mathbf{Y}^{\mathsf{T}}\mathbf{X}^{\mathsf{T}}$$

Problem 5 (10 pts)

Consider the eigenvalue decomposition of a symmetric matrix $\mathbf{A}$. Prove that two eigenvectors $\mathbf{v_i}$ and $\mathbf{v_j}$ associated with two distinct eigenvalues λ_i and λ_j of $\mathbf{A}$ are mutually orthogonal; that is, $\mathbf{v_i}^{\mathsf{T}}\mathbf{v_j} = 0$

Problem 6 (20 pts)

Refer to the Gram-Schmidt orthonormalization process described in the following wikipedia entry:

https://en.wikipedia.org/wiki/Gram%E2%80%93Schmidt_process

This procedure is a method for orthonormalizing a set of vectors in an inner product space. In other words, it allows you to find an orthogonal basis for a set of vectors.

The *projection operator* is given by:

$$proj_{\mathbf{u}}(\mathbf{v}) = \frac{< \mathbf{u}, \mathbf{v} >}{< \mathbf{u}, \mathbf{u} >}\mathbf{u}$$

This projector operator projects the vector $\mathbf{v}$ orthogonally onto the line spanned by vector $\mathbf{u}$.

6.1 Function `inner_product` (10 pts)

Write an R function `inner_product()` that calculates the inner product $< \mathbf{u}, \mathbf{v} >$ of two vectors (of the same length) $\mathbf{u}$ and $\mathbf{v}$.

Given two vectors `v` and `u`, you should be able to invoke your function like:

```
inner_product(v, u)
```

Test `inner_product(v, u)` with $\mathbf{v} = (1, 3, 5)$ and $\mathbf{u} = (1, 2, 3)$

6.2 Function `projection()` (10 pts)

Use your `inner_product()` function to write an R function `projection()` for the projection operator.

Given two vectors `u` and `v`, you should be able to call your function like:

```
projection(v, u)
```

Test `projection(v, u)` with $\mathbf{v} = (1, 3, 5)$ and $\mathbf{u} = (1, 2, 3)$

Problem 7 (10 pts)

Refer to the same wikipedia entry of the previous question. Once you have your function `projection()`, write R code to apply the Gram-Schmidt orthonormalization procedure to the following sets of vectors:

$\mathbf{x} = (1, 2, 3); \quad \mathbf{y} = (3, 0, 2); \quad \mathbf{z} = (3, 1, 1)$

Start by setting $\mathbf{u_1} = \mathbf{x}$, and report the set of vectors $\mathbf{u_k}$ and the orthonormalized vectors $\mathbf{e_k}$, for $k = 1, 2, 3$.

Problem 8 (10 pts)

The length of a vector $\mathbf{x} = (x_1, x_2, \dots, x_n)$ in the n-dimensional real vector space $\mathbb{R}^n$ is usually given by the Euclidean norm:

$$\|\mathbf{x}\| = (x_1^2 + x_2^2 + \cdots + x_n^2)^{1/2}$$

In many situations, the Euclidean distance is insufficient for capturing the actual distances in a given space. The class of p-norms generalizes the notion of *length* of a vector and it is defined by:

$$\|\mathbf{x}\|_p = (|x_1|^p + |x_2|^p + \cdots + |x_n|^p)^{\frac{1}{p}}$$

where p is a real number ≥ 1.

Write a function `lp_norm()` that computes the L_p-norm of a vector. This function should take two arguments:

- `x` the input vector
- `p` the value for p
- Give `p` a default value of 1
- Allow the user to specify `p = "max"` to compute the L_∞-norm

You should be able to call `lp_norm()` like this:

```
lp_norm(x)              # default p = 1
lp_norm(x, p = 2)
lp_norm(x, p = "max")   # L-max norm
```

Problem 9 (10 pts)

Use your function `lp_norm()` with the following vectors and values for `p`:

a. `zero <- rep(0, 10)` and `p = 1`

b. `ones <- rep(1, 5)` and `p = 2`

c. `u <- rep(0.4472136, 5)` and `p = 2`

d. `u <- 1:500` and `p = 100`

e. `u <- 1:500` and `p = "max"`

Problem 10 (10 pts)

Consider the eigendecomposition of a square matrix $\mathbf{A}$.

a. Prove that the matrix $b\mathbf{A}$, where b is an arbitrary scalar, has $b\lambda$ as an eigenvalue, with $\mathbf{v}$ as the associated eigenvector.

b. Prove that the matrix $\mathbf{A}+c\mathbf{I}$, where c is an arbitrary scalar, has $(\lambda+c)$ as an eigenvalue, with $\mathbf{v}$ as the associated eigenvector.

Problem 11 (20 pts)

For this problem, use the data set `state.x77` that comes in R.

a. Select the first five columns of `state.x77` and convert them as a matrix; this will be the data matrix $\mathbf{X}$. Let n be the number of rows of $\mathbf{X}$, and p the number of columns of $\mathbf{X}$

b. Create a diagonal matrix $\mathbf{D} = \frac{1}{n}\mathbf{I}$ where $\mathbf{I}$ is the $n \times n$ identity matrix. Display the output of `sum(diag(D))`.

c. Compute the vector of column means $\mathbf{g} = \mathbf{X}^\mathsf{T}\mathbf{D}\mathbf{1}$ where $\mathbf{1}$ is a vector of 1's of length n. Display (i.e. print) `g`.

d. Calculate the mean-centered matrix $\mathbf{X_c} = \mathbf{X} - \mathbf{1}\mathbf{g}^\mathsf{T}$. Display the output of `colMeans(Xc)`.

e. Compute the (population) variance-covariance matrix $\mathbf{V} = \mathbf{X}^\mathsf{T}\mathbf{D}\mathbf{X} - \mathbf{g}\mathbf{g}^\mathsf{T}$. Display the output of `V`.

f. Let $\mathbf{D}_{1/S}$ be a $p \times p$ diagonal matrix with elements on the diagonal equal to $1/S_j$, where S_j is the standard deviation for the j-th variable. Display only the elements in the diagonal of $\mathbf{D}_{1/S}$

g. Compute the matrix of standardized data $\mathbf{Z} = \mathbf{X_c}\mathbf{D}_{1/S}$ Display the output of `colMeans(Z)` and `apply(Z, 2, var)`

h. Compute the (population) correlation matrix $\mathbf{R} = \mathbf{D}_{1/S}\mathbf{V}\mathbf{D}_{1/S}$. Display the matrix `R`

i. Confirm that $\mathbf{R}$ can also be obtained as $\mathbf{R} = \mathbf{Z}^\mathsf{T}\mathbf{D}\mathbf{Z}$