Exercises for *Vectors, Matrices, and Least Squares*

Stephen Boyd        Lieven Vandenberghe

September 18, 2016

This is a collection of exercises for the (draft) book *Vectors, Matrices, and Least Squares*, by Stephen Boyd and Lieven Vandenberghe. They are used in EE103 (Stanford) and EE103 (UCLA). Eventually we will move some of them into the book. We will be updating this file frequently, so be sure to download it often, for example, before starting your homework assignment.

The first 19 sections follow the book chapters.

*Stephen Boyd and Lieven Vandenberghe*
## Contents

1. Vectors .................................................. 3
2. Linear functions .......................................... 6
3. Norm and distance ........................................ 8
4. Clustering .................................................. 14
5. Linear independence ...................................... 16
6. Matrices .................................................... 18
7. Matrix examples ........................................... 22
8. Linear equations .......................................... 25
9. Linear dynamical systems ............................... 26
10. Matrix multiplication ..................................... 28
11. Matrix inverses ........................................... 33
12. Least squares ............................................. 38
13. Least squares data fitting ............................... 40
14. Least squares classification ............................ 45
15. Multi-objective least squares ........................... 47
16. Constrained least squares ............................... 50
17. Constrained least squares applications .............. 51
18. Nonlinear least squares ................................ 54
19. Constrained nonlinear least squares ................. 55
20. Julia ....................................................... 56
1 Vectors

1.1 Vector equations. For each of the equations below, determine which of the following three options holds: The equation contains bad notation; it is valid notation and true; or, it is valid notation but false. (You can choose only one of these options.)

(a) \[
\begin{bmatrix}
1 \\
2 \\
1
\end{bmatrix} = [1 \ 2 \ 1].
\]

(b) \[
\begin{bmatrix}
1 \\
2 \\
1
\end{bmatrix} = (1, 2, 1).
\]

(c) \[
\begin{bmatrix}
2 \\
1
\end{bmatrix} = [1, 2, 1].
\]

(d) \((1, 2, 1)) = ((1, 2), 1).\]

1.2 Vector notation. Which of the following expressions uses correct notation? When the expression does make sense, give its length. In the following \(a\) and \(b\) are 10-vectors, and \(c\) is a 20-vector.

(a) \(a + b - c_{3:12}.\)

(b) \((a, b, c_{3:13}).\)

(c) \(2a + c.\)

(d) \((a, 1) + (b, c_1).\)

(e) \(((a, b), a).\)

(f) \([a \ b] + 4c.\)

(g) \[
\begin{bmatrix}
a \\
b
\end{bmatrix} + 4c.
\]

1.3 Julia timing test. Determine how much time it takes for your computer to compute the inner product of two vectors of length \(10^8\) (100 million), and use this to estimate (very crudely) how many Gflops/sec your computer can carry out. The following code generates two (random) vectors of length \(10^8\), and times the evaluation of the inner product. (You might run it a few times; the first time might be a bit slower.)

\[
a = \text{randn}(10^8);
b = \text{randn}(10^8);
tic(); s=\text{dot}(a,b); \text{toc}();
\]

How long would it take a person to carry this out, assuming the person can carry out a floating point operation every 10 seconds for 8 hours each day?

1.4 Interpreting sparsity. Suppose the \(n\)-vector \(x\) is sparse, \(i.e.,\) has only a few nonzero entries. Give a short sentence or two explaining what this means in each of the following contexts.

(a) \(x\) represents the daily cash flow of some business over \(n\) days.
(b) $x$ represents the annual dollar value purchases by a customer of $n$ products or services.
(c) $x$ represents a portfolio, say, the dollar value holdings of $n$ stocks.
(d) $x$ is the daily rainfall in a location over one year.

1.5 **Average age in a population.** Suppose the 100-vector $x$ represents the distribution of ages in a population, with $x_i$ being the number of $i - 1$ year olds, for $i = 1, \ldots, 100$. Find an expression for the average age across the population. (You can assume that $x \neq 0$, and that there is no one in the population over age 99.) Your expression must use vector notation; you cannot use a sum over ages. Your expression can include vectors, inner products, vector addition, scalar-vector multiplication, and the usual operations of (scalar) addition, subtraction, multiplication, and division.

1.6 **Questionnaire scoring.** A questionnaire in a magazine has 30 questions, broken into two sets of 15 questions. Someone taking the questionnaire answers each question with ‘Rarely’, ‘Sometimes’, or ‘Often’. Their score is found by adding up 1 point for every question answered Sometimes, and 2 points for every question answered Often on questions 1–15, and 2 points and 4 points for those responses on questions 16–30. (Nothing is added to the score for Rarely responses.)

The questionnaire answers are recorded as a 30-vector $a$, with $a_i = 1, 2, 3$ if question $i$ is answered Rarely, Sometimes, or Often, respectively.

Express the total score in the form $s = w^T a + v$, where $w$ is a 30-vector, and $v$ is a scalar (number). You must give $v$ and the entries of $w$.

*Remark.* (This is not needed to solve the problem.) Such a questionnaire might be used to answer the question ‘Do I have a Linear Algebra problem?’. An example question would be ‘How often do you dream about vectors or matrices?’. You’d look up your score to get a summary; for example, for scores above 75, the summary might be ‘You may have a serious problem and should seek professional help.’

1.7 **Industry or sector exposure.** Consider a set of $n$ assets or stocks that we invest in. Let $f$ be an $n$-vector that encodes whether each asset is in some specific industry or sector, e.g., pharmaceuticals or consumer electronics. Specifically, we take $f_i = 1$ if asset $i$ is in the sector, and $f_i = 0$ if it is not.

Let the $n$-vector $h$ denote a portfolio, with $h_i$ the dollar value held in asset $i$ (with negative meaning a short position). The inner product $f^T h$ is called the (dollar value) exposure of our portfolio to the sector. It gives the net dollar value of the portfolio that is invested in assets from the sector.

A portfolio $h$ is called *neutral* (to a sector or industry) if $f^T h = 0$.

A portfolio $h$ is called *long only* if each entry is nonnegative, i.e., $h_i \geq 0$ for each $i$. This means the portfolio does not include any short positions.

What does it mean if a long-only portfolio is neutral to a sector, say, pharmaceuticals? Your answer should be in simple English, but you should back up your conclusion with an argument.

1.8 **Cheapest supplier.** You must buy $n$ raw materials in quantities given by the $n$-vector $q$, where $q_i$ is the amount of raw material $i$ that you must buy. A set of $K$ potential suppliers offer the raw materials at prices given by the $n$-vectors $p_1, \ldots, p_K$. (Note that $p_k$ is an $n$-vector; $(p_k)_i$ is the price that supplier $k$ charges per unit of raw material $i$.) We will assume that all quantities and prices are positive.

If you must choose just one supplier, how would you do it? Your answer should use vector notation.
A (highly paid) consultant tells you that you might do better (i.e., get a better total cost) by splitting your order into two, for example ordering \((1/2)q\) from each of two suppliers. He argues that having a diversity of suppliers is better. Is he right?
2 Linear functions

2.1 Linear or not? Determine whether each the following scalar valued functions of \( n \)-vectors is linear. If it is a linear function, give its inner product representation, i.e., an \( n \)-vector \( a \) for which \( f(x) = a^T x \) for all \( x \). If it is not linear, give specific \( x, y, \alpha \), and \( \beta \) for which superposition fails, i.e.,
\[
f(\alpha x + \beta y) \neq \alpha f(x) + \beta f(y).
\]

(a) The spread of values of the vector, defined as \( f(x) = \max_k x_k - \min_k x_k \).
(b) The difference of the last element and the first, \( f(x) = x_n - x_1 \).
(c) The median of an \( n \)-vector, defined as the middle value of the sorted vector, when \( n \) is odd, and the average of the two middle values in the sorted vector, when \( n \) is even.
(d) The average of the entries with odd indices, minus the average of the entries with even indices.
(e) Vector extrapolation, defined as \( x_{n+1} = x_n + (x_n - x_{n-1}) \), for \( n \geq 2 \). (This is a simple prediction of what \( x_{n+1} \) would be, based on a straight line drawn through \( x_n \) and \( x_{n-1} \).

2.2 Regression model. Consider the regression model \( \hat{y} = x^T \beta + v \), where \( \hat{y} \) is the predicted response, \( x \) is an \( 8 \)-vector of features, \( \beta \) is an \( 8 \)-vector of coefficients, and \( v \) is the offset term. Determine whether each of the following statements is true or false. You must justify each answer.

(a) If \( \beta_3 > 0 \) and \( x_3 > 0 \), then \( \hat{y} \geq 0 \).
(b) If \( \beta_2 = 0 \) then \( \hat{y} \) does not depend on \( x_2 \).
(c) If \( \beta_6 = -0.8 \), then increasing \( x_6 \) (keeping all other \( x_i \)'s the same) will decrease \( \hat{y} \).

2.3 Processor powers and temperature. The temperature \( T \) of an electronic device containing three processors is an affine function of the power dissipated by the three processors, \( P = (P_1, P_2, P_3) \). When all three processors are idling, we have \( P = (10, 10, 10) \), which results in a temperature \( T = 30 \). When the first processor operates at full power and the other two are idling, we have \( P = (100, 10, 10) \), and the temperature rises to \( T = 60 \). When the second processor operates at full power and the other two are idling, we have \( P = (10, 100, 10) \) and \( T = 70 \). When the third processor operates at full power and the other two are idling, we have \( P = (10, 10, 100) \) and \( T = 65 \). Now suppose that all three processors are operated at the same power \( P^{\text{same}} \). How large can \( P^{\text{same}} \) be, if we require that \( T \leq 85 \)?

2.4 Interpreting AR model coefficients. Consider an AR model for a time series \( x_1, x_2, \ldots \),
\[
\hat{x}_{t+1} = (x_t, x_{t-1}, \ldots, x_{t-M+1})^T \beta, \quad t = M, M+1, \ldots.
\]
Here \( \hat{x}_{t+1} \) denotes the model’s prediction of \( x_{t+1} \), and \( M \) is the memory length of the AR model, and the \( M \)-vector \( \beta \) is the AR model coefficient vector. For this problem we will assume that the time period is daily, and \( M = 10 \). Thus, the AR model predicts tomorrow’s value, given the value over the last 10 days.

For each of the following cases, give a short interpretation or description of the model in English, that anyone can understand. Your explanation should be brief, and it may not refer to mathematical concepts like vectors, inner product, and so on. You can use words like ‘yesterday’, ‘today’, and so on.
(a) $\beta \approx e_1$.
(b) $\beta \approx 2e_1 - e_2$.
(c) $\beta \approx e_7$.
(d) $\beta \approx 0.5e_1 + 0.5e_2$.

2.5 Deviation of middle element value from average. Suppose $x$ is a $n$-vector, with $n = 2m - 1$ and $m \geq 1$. We define the middle element value of $x$ as $x_m$. Define

$$f(x) = x_m - \frac{1}{n} \sum_{i=1}^{n} x_i,$$

which is the difference between the middle element value and the average of the coefficients in $x$. Express $f$ in the form $f(x) = a^T x$, where $a$ is an $n$-vector.
3 Norm and distance

3.1 RMS value and average of block vectors. Let $x$ be a block vector with two vector elements,

$$x = \begin{bmatrix} a \\ b \end{bmatrix},$$

where $a$ and $b$ are vectors of size $n$ and $m$, respectively.

(a) Express $\text{rms}(x)$ in terms of $\text{rms}(a)$, $\text{rms}(b)$, $m$, and $n$.

(b) Express $\text{avg}(x)$ in terms of $\text{avg}(a)$, $\text{avg}(b)$, $m$, and $n$.

3.2 Properties of RMS value and norm. Let $x$ and $z$ be $n$-vectors, and $y$ be an $m$-vector, with $m \neq n$. Determine whether each of the following equations always holds, does not always hold, or makes no sense. (You must choose one of these options.)

(a) $\text{rms}((x,y)) = \text{rms}((\text{rms}(x), \text{rms}(y)))$

(b) $\| (x,y) \| = \left\| \begin{bmatrix} \|x\| \\ \|y\| \end{bmatrix} \right\|$

(c) $\text{rms}(x + z) \leq \text{rms}(x) + \text{rms}(z)$

3.3 Suppose $x$ is a 100-vector with $\text{rms}(x) = 1$.

(a) What are all possible values of $\text{avg}(x)$?

(b) What is the maximum number of entries of $x$ that satisfy $|x_i| \geq 3$?

(c) Let $q_{90}$ be the 90th percentile of the absolute values of the entries of the vector $x$. (This is the 90th entry of the vector obtained by sorting the absolute values of the entries of $x$ in increasing order.) How big and how small can $q_{90}$ be?

You can give intuitive arguments defending your answers.

3.4 Suppose $x$ is an $n$-vector and $\alpha$ and $\beta$ are scalars.

(a) Show that $\text{avg}(\alpha x + \beta 1) = \alpha \text{avg}(x) + \beta$.

(b) Show that $\text{std}(\alpha x + \beta 1) = |\alpha| \text{std}(x)$.

3.5 Correlation coefficient. Each of the following plots shows the points corresponding to two vectors $x$ and $y$ of the same size, i.e., we plot points at the locations $(x_i, y_i)$. In each case, determine whether the correlation coefficient $\rho$ of the two vectors is positive (and, say, $\geq 0.5$), negative (say, $\leq -0.5$), or near zero (say, less than 0.3 in absolute value). (You must choose one of these options.)
3.6 Distance from Palo Alto to Beijing. The surface of the earth is reasonably approximated as a sphere with radius $R = 6367.5$ km. A location on the earth’s surface is traditionally given by its latitude $\theta$ and its longitude $\lambda$, which correspond to angular distance from the equator and prime meridian, respectively. The 3-D coordinates of the location are given by $R(\sin \lambda \cos \theta, \cos \lambda \cos \theta, \sin \theta)$.

(In this coordinate system $(0,0,0)$ is the center of the earth, $R(0,0,1)$ is the North pole, and $R(0,1,0)$ is the point on the equator on the prime meridian, due south of the Royal Observatory outside London. And no, you don’t need to know any of this.)

The distance through the earth between two locations (3-vectors) $a$ and $b$ is $\|a - b\|$. The distance along the surface of the earth between points $a$ and $b$ is $R\angle(a,b)$. Find these two distances between Palo Alto and Beijing, with latitudes and longitudes given below.

<table>
<thead>
<tr>
<th>City</th>
<th>Latitude $\theta$</th>
<th>Longitude $\lambda$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beijing</td>
<td>39.914°</td>
<td>116.392°</td>
</tr>
<tr>
<td>Palo Alto</td>
<td>37.429°</td>
<td>-122.138°</td>
</tr>
</tbody>
</table>
**Hint.** In Julia, the functions `cos` and `sin` compute cos and sin taking the argument in radians. Julia also has the functions `cosd` and `sind` which take the argument in degrees.

### 3.7 Difference of squared distances

Determine whether the difference of the squared distances to two fixed vectors $c$ and $d$, defined as

$$f(x) = \|x - c\|^2 - \|x - d\|^2,$$

is linear, affine, or neither. If it is linear, give its inner product representation, *i.e.*, an $n$-vector $a$ for which $f(x) = a^T x$ for all $x$. If it is affine, give $a$ and $b$ for which $f(x) = a^T x + b$ holds for all $x$. If it is not, give specific $x$, $y$, $\alpha$, and $\beta$ for which superposition fails, *i.e.*, $f(\alpha x + \beta y) \neq \alpha f(x) + \beta f(y)$.

(Provided $\alpha + \beta = 1$, this shows the function is neither linear nor affine.)

### 3.8 Reverse triangle inequality

Suppose $a$ and $b$ are vectors of the same size. The triangle inequality states that $\|a + b\| \leq \|a\| + \|b\|$. Show that we also have

$$\|a + b\| \geq \|a\| - \|b\|.$$  

Swapping $a$ and $b$ here yields $\|a + b\| \geq \|b\| - \|a\|$, so we have

$$|\|a\| - \|b\|| \leq \|a + b\| \leq \|a\| + \|b\|.$$

**Hints.**

- First draw a picture to get the idea.
- Apply the triangle inequality to $(a + b) + (-b)$.

### 3.9 Regression model sensitivity

Consider the regression model $\hat{y} = x^T \beta + v$, where $\hat{y}$ is the prediction, $x$ is a feature vector, $\beta$ is a coefficient vector, and $v$ is the offset term. If $x$ and $x'$ are feature vectors with corresponding predictions $\hat{y}$ and $\hat{y}'$, show that $|\hat{y} - \hat{y}'| \leq \|\beta\| \|x - x'\|$. This means that when $\|\beta\|$ is small, the prediction is not very sensitive to a change in the feature vector.

### 3.10 Nearest neighbor and smallest angle

Using Julia, find the nearest neighbor of $a = (1, 3, 4)$ among the vectors

$$x_1 = (4, 3, 5), \quad x_2 = (0.4, 10, 50), \quad x_3 = (1, 4, 10), \quad x_4 = (30, 40, 50).$$

Report the minimum distance of $a$ to $x_1, \ldots, x_4$. Also, find which of $x_1, \ldots, x_4$ makes the smallest angle with $a$ and report that angle.

### 3.11 Orthogonality

Suppose the $n$-vectors $a$, $b$, and $c$ satisfy $a \perp c$ and $b \perp c$. Which of the following statements must hold? (That is, are true for any $a$, $b$, and $c$ that satisfy $a \perp c$, $b \perp c$.)

(a) $a \perp b$.
(b) $(a + b) \perp c$.
(c) $(a + c) \perp (b + c)$.
3.12 **Angle between two nonnegative vectors.** Let $x$ and $y$ be two $n$-vectors with nonnegative entries, i.e., each $x_i \geq 0$ and each $y_i \geq 0$. Show that the angle $x$ and $y$ lies between 0 and 90°. Draw a picture for the case when $n = 2$, and give a short geometric explanation. When are $x$ and $y$ orthogonal?

3.13 **Guessing means and standard deviations.** Each of the plots below shows a vector $x$, with $x_i$ plotted on the vertical axis and $i$ on the horizontal axis. For each case, estimate $\text{avg}(x)$ and $\text{std}(x)$. We are looking for a crude guess, say, within a factor of two.

![Graphs](image_url)

3.14 **Risk reduction by diversification.** The $n$-vectors $a$ and $b$ are time series of returns on two assets over the same period, given as percentages. (So, for example, $a_8 = -0.03$ means that the first asset lost 3% in period 8.) We assume that the two assets have the same mean return and risk (i.e., standard deviation) over the full period:

$$\text{avg}(a) = \text{avg}(b) = \mu, \quad \text{std}(a) = \text{std}(b) = \sigma.$$
(These symbols are traditional ones used for the mean and standard deviation.) We let \( \rho \) denote the correlation coefficient of \( a \) and \( b \).

We will **diversify** by investing half in the first asset and half in the second. This diversified portfolio has return time series \( c = (a + b)/2 \).

(a) Show that \( \text{avg}(c) = \mu \). This means that the diversified portfolio has the same mean return as the original ones.

(b) Show that \( \text{std}(c) = \sigma \sqrt{(1 + \rho)/2} \). Comment briefly on why this shows that diversifying is advantageous.

*Note.* Julia’s \( \text{std}(x) \) is not the same as the \( \text{std}(x) \) defined in the book, so if you check this using Julia make sure to use the book’s version.

3.15 Let \( \alpha, \beta, \) and \( \gamma \) be scalars and let \( a, b, \) and \( c \) be pairwise orthogonal \( n \)-vectors. (This means that \( a \perp b, a \perp c, \) and \( b \perp c \).) Express \( \|\alpha a + \beta b + \gamma c\| \) in terms of \( \|a\|, \|b\|, \|c\|, \alpha, \beta, \) and \( \gamma \).

3.16 **True or false.** Determine whether each of the following statements is true or false.

(a) If \( n \)-vectors \( x \) and \( y \) make an acute angle, then \( \|x + y\| \geq \max\{\|x\|, \|y\|\} \).

(b) For any vector \( a \), \( \text{avg}(a) \leq \text{rms}(a) \).

3.17 **Taylor approximation of norm.** Find a formula for the Taylor approximation of the function \( f(x) = \|x\| \) near a given nonzero vector \( \vec{x} \). Check your approximation numerically as follows. First, choose \( \vec{x} \) as a random 5-vector. Then choose several vectors \( x \) randomly, with \( \|x - \vec{x}\| \) on the order of \( 0.1\|\vec{x}\| \), and compare \( f(x) = \|x\| \) and \( f(\vec{x}) = \|\vec{x}\| \).

3.18 **Norm identities.** Verify that the following identities hold for any two vectors \( a \) and \( b \) of the same size.

(a) \((a + b)^T(a - b) = \|a\|^2 - \|b\|^2\).

(b) \(\|a + b\|^2 + \|a - b\|^2 = 2(\|a\|^2 + \|b\|^2)\).

3.19 **Triangle equality.** When does the triangle inequality hold with equality, i.e., what are the conditions on \( a \) and \( b \) to have \( \|a + b\| = \|a\| + \|b\| \)?

3.20 Show that \( \|a + b\| \geq \|\|a\| - \|b\|\| \).

3.21 **Average and norm.** Use the Cauchy-Schwarz inequality to show that

\[
-\frac{1}{\sqrt{n}} \|x\| \leq \frac{1}{n} \sum_{i=1}^{n} x_i \leq \frac{1}{\sqrt{n}} \|x\|
\]

for any \( n \)-vector \( x \). Explain how this shows that, for any vector \( x \),

\[-\text{rms}(x) \leq \text{avg}(x) \leq \text{rms}(x),\]

i.e., the average value lies between plus and minus the RMS value. Is it possible for equality to hold for the lefthand or righthand inequality? If so, when?
3.22 Norm of sum. Derive a formula for $\|a + b\|$ in terms of $\|a\|$, $\|b\|$, and $\theta = \angle(a, b)$. Use this formula to show the following:

(a) $a \perp b$ if and only if $\|a + b\| = \sqrt{\|a\|^2 + \|b\|^2}$.
(b) $a$ and $b$ make an acute angle if and only if $\|a + b\| \geq \sqrt{\|a\|^2 + \|b\|^2}$.
(c) $a$ and $b$ make an obtuse angle if and only if $\|a + b\| \leq \sqrt{\|a\|^2 + \|b\|^2}$.

Draw a picture illustrating each case (in $\mathbb{R}^2$).

3.23 Perfect correlation. Suppose nonzero vectors $x$ and $y$ are perfectly correlated, which means their correlation coefficient is one. Show that this implies there are numbers $a$ and $b$ for which $y = ax + b$.

3.24 Laplacian of a signal. Suppose the $T$-vector $x$ represents a time series or signal. The quantity

$$\mathcal{L} = (x_1 - x_2)^2 + \cdots + (x_{T-1} - x_T)^2,$$

the sum of the differences of adjacent values of the signal, is called the Laplacian of the signal. The Laplacian is a measure of the roughness or wiggliness of the time series. It is sometimes divided by $\|x\|^2$, to give a normalized value.

(a) Express $\mathcal{L}$ is vector notation. (You can use vector slicing, vector addition or subtraction, inner product, norm, and angle.)
(b) How small can the Laplacian be? What signals $x$ have this minimum value of the Laplacian?
(c) Find a signal with entries no more than one in absolute value that has the largest possible value of $\mathcal{L}$. Give the value of the Laplacian achieved.

3.25 Nearest point to a line. Let $a$ be an $n$-vector and $v$ a nonzero $n$-vector. The set of vectors of the form $a + tv$ for all real numbers $t$ is the line passing through $a$ in the direction $v$. This agrees with the usual definition of a line when the vectors represent 2-D or 3-D coordinates.

Let $x$ be any $n$-vector. Find a formula for the point $p$ on the line that is closest to $x$. The point $p$ is called the projection of $x$ onto the line. Show that $(p - x) \perp v$. Hint. You might want to work with the square of the distance between a point on the line and $x$.

3.26 Triangle inequality for angles. Show that $\angle(x, y) \leq \angle(x, z) + \angle(z, y)$ for any nonzero vectors $x$, $y$, $z$. In other words, angles satisfy the triangle inequality. (Recall that angles are normalized to lie between 0 and $\pi$.)

Hints.

- We can just as well assume that $\|x\| = \|y\| = \|z\| = 1$. This will greatly simplify the formulas for the angles.
- When $\|x\| = \|y\| = 1$, $\|x - y\| = \sqrt{2(1 - \cos \angle(x, y))}$.
- You might find the identity $\cos^2(\alpha + \beta) = \cos^2 \alpha + \cos^2 \beta - 1$ useful.
4 Clustering

4.1 Building a recommendation engine using $k$-means. A set of $N$ users of a music-streaming app listens to songs from a library of $n$ songs over some period (say, a month). We describe user $i$’s listening habits by her playlist vector, which is the $n$-vector $p_i$ defined as

$$(p_i)_j = \begin{cases} 1 & \text{user } i \text{ has played song } j \\ 0 & \text{user } i \text{ has not played song } j \end{cases},$$

for $j = 1, \ldots, n$. (Note that $p_i$ is an $n$-vector, while $(p_i)_j$ is a number.) You can assume that if a user listens to a song, she likes it.

Your job (say, during a summer internship) is to design an algorithm that recommends to each user 10 songs that she has not listened to, but might like. (You can assume that for each user, there are at least 10 songs that she has not listened to.)

To do this, you start by running $k$-means on the set of playlist vectors $p_1, \ldots, p_N$. (It’s not relevant here, but a reasonable choice of $k$ might be 100 or so.) This gives the centroids $z_1, \ldots, z_k$, which are $n$-vectors.

Now what do you do? You can explain in words; you do not need to give a formula to explain how you make the recommendations for each user.

4.2 Topic discovery via $k$-means. In this problem you will use $k$-means to cluster 300 Wikipedia articles selected from 5 broad groups of topics. The Julia file wikipedia_corpus.jl contains the histograms as a list of 300 1000-vectors in the variable article_histograms. It also provides the list of article titles in article_titles and a list of the 1000 words used to create the histograms in dictionary.

The file kmeans.jl provides a Julia implementation of the $k$-means algorithm in the function kmeans. The kmeans function accepts a list of vectors to cluster along with the number of clusters, $k$, and returns three things: the centroids as a list of vectors, a list containing the index of each vector’s closest centroid, and a list of the value of $J$ after each iteration of $k$-means. Each time the function kmeans is invoked it initializes the centroids by randomly assigning the data points to $k$ groups and taking the $k$ representatives as the means of the groups. (This means that if you run kmeans twice, with the same data, you might get different results.)

For example, here is an example of running $k$-means with $k = 8$ and finding the 30th article’s centroid.

```julia
include("wikipedia_corpus.jl")
include("kmeans.jl")
using Kmeans

centroids, labels, j_hist = kmeans(article_histograms, 8)
centroids[labels[30]]
```

The list labels contains the index of each vector’s closest centroid, so if the 30th entry in labels is 7, then the the 30th vector’s closest centroid is the 7th entry in centroids.

There are many ways to explore your results. For example, you could print the titles of all articles in a cluster.
julia> article_titles[labels .== 7]
16-element Array{UTF8String,1}:
  "Anemometer"
  "Black ice"
  "Freezing rain"
  ...

Alternatively, you could find a topic’s most common words by ordering `dictionary` by the size of its centroid’s entries. A larger entry for a word implies it was more common in articles from that topic.

julia> dictionary[sortperm(centroids[7],rev=true)]
1000-element Array{ASCIIString,1}:
  "wind"
  "ice"
  "temperature"
  ...

(a) For each of $k = 2$, $k = 5$, and $k = 10$ run $k$-means twice, and plot $J$ (vertically) versus iteration (horizontally) for the two runs on the same plot. Create your plot by passing a vector containing the value of $J$ at each iteration to MMAPlot’s `line_plot` function. Comment briefly on your results.

(b) Choose a value of $k$ from part (a) and investigate your results by looking at the words and article titles associated with each centroid. Feel free to visit Wikipedia if an article’s content is unclear from its title. Give a short description of the topics your clustering discovered along with the 3 most common words from each topic. If the topics do not make sense pick another value of $k$. 
5 Linear independence

5.1 Linear independence of stacked vectors.

(a) Suppose \(a_1, \ldots, a_k\) are linearly independent \(n\)-vectors, and \(b_1, \ldots, b_k\) are any \(m\)-vectors. When are the stacked vectors
\[
c_1 = \begin{bmatrix} a_1 \\ b_1 \end{bmatrix}, \ldots, c_k = \begin{bmatrix} a_k \\ b_k \end{bmatrix}
\]
linearly independent? Your answer must be one of: Always, Never, or Sometimes. ‘Sometimes’ means that \(c_1, \ldots, c_k\) can be linearly independent for some choices of \(a_i\) and \(b_i\), and linearly dependent for other choices. You must justify your answer.

(b) Answer the same question, except that now we assume that \(a_1, \ldots, a_k\) are linearly dependent.

5.2 Linear combinations of cash flows. We consider cash flow vectors over \(T\) time periods, with a positive entry meaning a payment received, and negative meaning a payment made. A (unit) single period loan, at time period \(t\), is the \(T\)-vector \(l_t\) that corresponds to a payment received of \(\$1\) in period \(t\) and a payment made of \(\$(1 + r)\) in period \(t + 1\), with all other payments zero. Here \(r > 0\) is the interest rate (over one period).

(a) Show that \(l_1, \ldots, l_{T-1}\) are linearly independent.

(b) Let \(c\) be a \(\$1 T - 1\) period loan, starting at period 1. This means that \(\$1\) is received in period 1, \(\$(1 + r)^{T-1}\) is paid in period \(T\), and all other payments (i.e., \(c_2, \ldots, c_{T-1}\)) are zero. Express \(c\) as a linear combination of single period loans.

5.3 Orthogonalizing vectors. Suppose that \(a\) and \(b\) are any \(n\)-vectors. Show that we can always find a scalar \(\gamma\) so that \((a - \gamma b) \perp b\). (Give a formula for the scalar \(\gamma\).) Roughly speaking, we can always subtract a multiple of a vector from another one, so that the result is orthogonal to the original vector. This is called orthogonalization, and is a basic idea used in the Gram-Schmidt algorithm.

5.4 Linear independence under combination. Suppose \(S = \{a, b, c\}\) and \(T = \{d, e, f\}\) are two linearly independent sets of \(n\)-vectors. For each of the sets given below, determine which statement is correct. (Only one is correct in each case.)

(a) \(\{a, b, c, d, e, f\}\)
- is always linearly independent.
- is always linearly dependent.
- could be linearly independent or linearly dependent, depending on the values of \(a, \ldots, f\).

(b) \(\{a + d, b + e, c + f\}\)
- is always linearly independent.
- is always linearly dependent.
- could be linearly independent or linearly dependent, depending on the values of \(a, \ldots, f\).

(c) \(\{a, a + b, a + b + c\}\)
- is always linearly independent.
- is always linearly dependent.
- could be linearly independent or linearly dependent, depending on the values of \(a, \ldots, f\).
5.5 A surprising discovery. An intern at a quantitative hedge fund examines the daily returns of around 400 stocks over one year (which has 250 trading days). She tells her supervisor that she has discovered that the returns of one of the stocks, Google (GOOG), can be expressed as a linear combination of the others, which include many stocks that are unrelated to Google (say, in a different type of business or sector).

Her supervisor then says: “It is overwhelmingly unlikely that a linear combination of the returns of unrelated companies can reproduce the daily return of GOOG. So you’ve made a mistake in your calculations.”

Is the supervisor right? Did the intern make a mistake? Give a very brief explanation.

5.6 Linear independence. For each of the following matrices, determine which response is correct.

(a) \[
\begin{bmatrix}
428 & 973 & -163 & 245 & -784 & 557 \\
352 & 869 & 0 & 781 & -128 & 120 \\
1047 & 45 & -471 & 349 & -721 & 781 \\
\end{bmatrix}
\]

- The columns are linearly independent.
- The columns are linearly dependent.
- This is not an appropriate question for an in class 75 minute midterm.

(b) \[
\begin{bmatrix}
768 & 1121 & 3425 & 8023 \\
-2095 & -9284 & 5821 & -6342 \\
4093 & -3490 & -7249 & 8241 \\
834 & 1428 & 4392 & 5835 \\
-7383 & 1435 & 2345 & -293 \\
\end{bmatrix}
\]

- The columns are linearly independent.
- The columns are linearly dependent.
- This is not an appropriate question for an in class 75 minute midterm.

5.7 Gram-Schmidt properties.

(a) What happens if the Gram-Schmidt algorithm is applied to a set of vectors \( a_1, \ldots, a_k \) that are orthonormal?

(b) What happens if you apply the Gram-Schmidt algorithm twice, to any set of vectors? In other words: We run Gram-Schmidt once on the given set of vectors (we assume this is successful), and then we run Gram-Schmidt again on the vectors \( q_1, \ldots, q_k \) that come out.
6 Matrices

6.1 Matrix-vector multiplication. For each of the following matrices, describe in words how \( x \) and \( y = Ax \) are related.

(a) \( A = \begin{bmatrix} 0 & 0 & I_k \\ 0 & I_k & 0 \\ I_k & 0 & 0 \end{bmatrix} \).

(b) \( A = \begin{bmatrix} E & 0 & 0 \\ 0 & E & 0 \\ 0 & 0 & E \end{bmatrix} \), where \( E \) is the \( k \times k \) matrix with all entries \( 1/k \).

6.2 Checking superposition in Julia. Generate a random 20 \( \times \) 10 matrix \( A \), as well as 10-vectors \( x \) and \( y \), and scalars \( \alpha \) and \( \beta \). Evaluate the two 20-vectors \( A(\alpha x + \beta y) \) and \( \alpha (Ax) + \beta (Ay) \), and verify that they are very close by printing the norm and \text{rms} \ of the difference. (If the numerical calculations were done exactly, they would be equal. Due to very small rounding errors made in the floating-point calculations, they will not be exactly equal.)

Hint. The Julia function \texttt{rand} can be used to generate random scalars, vectors, and matrices. \texttt{rand()} generates a random number, \texttt{rand(n)} generates a random \( n \)-vector, and \texttt{rand(n,m)} generates a random \( n \times m \) matrix.

6.3 Vandermonde matrices. A Vandermonde matrix is an \( m \times n \) matrix of the form

\[
V = \begin{bmatrix}
1 & t_1 & t_1^2 & \cdots & t_1^{n-1} \\
1 & t_2 & t_2^2 & \cdots & t_2^{n-1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & t_m & t_m^2 & \cdots & t_m^{n-1}
\end{bmatrix}
\]

where \( t_1, \ldots, t_m \) are numbers. We will assume that these numbers are distinct, \textit{i.e.}, different from each other. Multiplying the Vandermonde matrix \( V \) by an \( n \)-vector \( c \) is the same as evaluating the polynomial of degree less than \( n \), with coefficients \( c_1, \ldots, c_n \), at the points \( t_1, \ldots, t_m \); see page 86 in the book.

Show that the columns of a Vandermonde matrix are linearly independent provided \( m \geq n \).

Hint. You can use the following fact from algebra: If a polynomial \( p \) with degree less than \( n \) has \( n \) or more roots (points \( t \) for which \( p(t) = 0 \) then all its coefficients are zero.

6.4 Vandermonde matrices in Julia. Write a function that takes a positive integer \( n \) and an \( m \)-vector \( t \) as inputs and generates the corresponding \( m \times n \) Vandermonde matrix, as described in the previous problem.

6.5 Currency exchange matrix. We consider a set of \( n \) currencies, labeled 1, \ldots, \( n \). (These might correspond to USD, RMB, EUR, and so on.) At a particular time the exchange or conversion rates among the \( n \) currencies are given by an \( n \times n \) (exchange rate) matrix \( R \), where \( R_{ij} \) is the amount of currency \( i \) that you can buy for one unit of currency \( j \). (All entries of \( R \) are positive.) The exchange rates include commission charges, so we have \( R_{ji}R_{ij} < 1 \) for all \( i \neq j \). You can assume that \( R_{ii} = 1 \).
Suppose \( y = Rx \), where \( x \) is a vector (with nonnegative entries) that represents the amounts of the currencies that we hold. Choose one of the following and briefly justify your answer.

- \( y \) gives a set of currency holdings that we can obtain by exchanging our holdings.
- \( y_i \) is the amount of currency \( i \) we can obtain by converting all our currencies into currency \( i \).
- \( y_i \) is the amount of currency \( i \) we need to re-create our holdings by exchanges.

6.6 Cash flow to bank account balance. The \( T \)-vector \( c \) represents the cash flow for an interest bearing bank account over \( T \) time periods. Positive values of \( c \) indicate a deposit, and negative values indicate a withdrawal. The \( T \)-vector \( b \) denotes the bank account balance in the \( T \) periods. We have \( b_1 = c_1 \) (the initial deposit or withdrawal) and

\[
b_t = (1 + r)b_{t-1} + c_t, \quad t = 2, \ldots, T,
\]

where \( r > 0 \) is the (per-period) interest rate. (The first term is the previous balance plus the interest, and the second term is the deposit or withdrawal.)

Find the \( T \times T \) matrix \( A \) for which \( b = Ac \). Roughly speaking, this matrix maps a cash flow sequence into a bank account balance sequence. Your description must make clear what all entries of \( A \) are.

6.7 Multiple channel marketing campaign. Potential customers are divided into \( m \) market segments, which are groups of customers with similar demographics, e.g., college educated women aged 25–29. A company markets its products by purchasing advertising in a set of \( n \) channels, i.e., specific TV or radio shows, magazines, web sites, blogs, direct mail, and so on. The ability of each channel to deliver impressions or views by potential customers is characterized by the reach matrix, the \( m \times n \) matrix \( R \), where \( R_{ij} \) is the number of views of customers in segment \( i \) for each dollar spent on channel \( j \). (We assume that the total number of views in each market segment is the sum of the views from each channel, and that the views from each channel scale linearly with spending.) The \( n \)-vector \( c \) will denote the company’s purchases of advertising, in dollars, in the \( n \) channels. The \( m \)-vector \( v \) gives the total number of impressions in the \( m \) market segments due to the advertising in all channels. Finally, we introduce the \( m \)-vector \( a \), where \( a_i \) gives the profit in dollars per impression in market segment \( i \). The entries of \( R \), \( c \), \( v \), and \( a \) are all nonnegative.

(a) Express the total amount of money the company spends on advertising using vector/matrix notation.

(b) Express \( v \) using vector/matrix notation, in terms of the other vectors and matrices.

(c) Express the total profit from all market segments using vector/matrix notation.

(d) How would you find the single channel most effective at reaching market segment 3, in terms of impressions per dollar spent?

(e) What does it mean if \( R_{35} \) is very small (compared to other entries of \( R \))?  

6.8 Integral of product of polynomials. Let \( p \) and \( q \) be two quadratic polynomials, given by

\[
p(x) = c_1 + c_2x + c_3x^2, \quad q(x) = d_1 + d_2x + d_3x^2.
\]

Express the integral \( J = \int_0^1 p(x)q(x) \, dx \) in the form \( J = c^T G d \), where \( G \) is a \( 3 \times 3 \) matrix. Give the entries of \( G \) (as numbers).
6.9 Let $A$ and $B$ be two $m \times n$ matrices. For each of the statements below, determine whether $A = B$ must always hold, or whether $A = B$ holds only sometimes.

(a) Suppose $Ax = Bx$ holds for all $n$-vectors $x$.
(b) Suppose $Ax = Bx$ for some nonzero $n$-vector $x$.

6.10 Linear functions of images. In this problem we consider several linear functions of a monochrome image with $N \times N$ pixels. To keep the matrices small enough to work out by hand, we will consider the case with $N = 3$ (which would hardly qualify as an image). We represent a $3 \times 3$ image as a 9-vector using the ordering of pixels shown below.

$$
\begin{array}{ccc}
1 & 4 & 7 \\
2 & 5 & 8 \\
3 & 6 & 9 \\
\end{array}
$$

(This ordering is called column-major.) Each of the operations or transformations below defines a function $y = f(x)$, where the 9-vector $x$ represents the original image, and the 9-vector $y$ represents the resulting or transformed image. For each of these operations, give the $9 \times 9$ matrix $A$ for which $y = Ax$.

(a) Turn the original image $x$ upside-down.
(b) Rotate the original image $x$ clockwise $90^\circ$.
(c) Translate the image up by 1 pixel and to the right by 1 pixel. In the translated image, assign the value $y_i = 0$ to the pixels in the first column and the last row.
(d) Set each pixel value $y_i$ to be the average of the neighbors of pixel $i$ in the original image. By neighbors, we mean the pixels immediately above and below, and immediately to the left and right. The center pixel has 4 neighbors; corner pixels have 2 neighbors, and the remaining pixels have 3 neighbors.

6.11 Linear functions. For each description of $y$ below, express it as $y = Ax$ for some $A$. (You should specify $A$.)

(a) $y_i$ is the difference between $x_i$ and the average value of $x_1, \ldots, x_{i-1}$. (We take $y_1 = x_1$.)
(b) $y_i$ is the difference between $x_i$ and the average value of all other $x_j$'s, i.e., the average value of $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$.

6.12 Trimming a vector. Find a matrix $A$ for which $Ax = (x_2, \ldots, x_{n-1})$, where $x$ is an $n$-vector. (Be sure to specify the size of $A$, and describe all its entries.)

6.13 Block matrix notation. Consider the block matrix

$$A = \begin{bmatrix}
I & B & 0 \\
B^T & 0 & 0 \\
0 & 0 & BB^T
\end{bmatrix}$$

where $B$ is $10 \times 5$. What are the dimensions of the zero matrices and the identity matrix in the definition of $A$, and of $A$ itself?
6.14 **Downsampling and up-conversion.** We consider \( n \)-vectors \( x \) that represent signals, with \( x_k \) the value of the signal at time \( k \) for \( k = 1, \ldots, n \). Below we describe two linear functions of \( x \) that produce new signals \( f(x) \). For each function, give a matrix \( A \) such that \( f(x) = Ax \) for all \( x \).

(a) 2× downsampling. We assume \( n \) is even, and define \( f(x) \) as the \( n/2 \)-vector \( y \) with elements \( y_k = x_{2k} \). To simplify your notation you can assume that \( n = 8 \), i.e.,

\[
    f(x) = (x_2, x_4, x_6, x_8).
\]

(b) 2× up-conversion with linear interpolation. We define \( f(x) \) as the \((2n - 1)\)-vector \( y \) with elements \( y_k = x_{(k+1)/2} \) if \( k \) is odd and \( y_k = (x_{k/2} + x_{k/2+1})/2 \) if \( k \) is even. To simplify your notation you can assume that \( n = 5 \), i.e.,

\[
    f(x) = (x_1, \frac{x_1 + x_2}{2}, x_2, \frac{x_2 + x_3}{2}, x_3, \frac{x_3 + x_4}{2}, x_4, \frac{x_4 + x_5}{2}, x_5).
\]
7 Matrix examples

7.1 Projection on a line. Let $P(x)$ denote the projection of the 2-D point (2-vector) $x$ onto the line that passes through $(0,0)$ and $(1,3)$. (This means that $P(x)$ is the point on the line that is closest to $x$; see exercise ??.) Show that $P$ is a linear function, and give a matrix $A$ for which $P(x) = Ax$ for any $x$.

7.2 Some basic properties of convolution. Suppose that $a$ is an $n$-vector, $b$ is an $m$-vector, and $c$ is a $p$-vector.

(a) Show that convolution is commutative, i.e., $a * b = b * a$.

Hint. Use the formula $(a * b)_k = \sum_{i+j=k+1} a_i b_j$, where the sum means you should sum over all $i,j$ that satisfy $i+j = k+1$, and we interpret $a_i$ and $b_j$ as zero when the index is outside its normal range.

(b) Show that convolution is associative, i.e., $(a * b) * c = a * (b * c)$. (This justifies writing either of these as $a * b * c$.)

Hint. Show that both the left and right hand sides can be expressed as

$$(a * b * c)_k = \sum_{i+j+\ell = k+2} a_i b_j c_\ell,$$

with $a_i$, $b_j$, and $c_\ell$ interpreted as 0 outside their ranges.

(c) Convolution with 1. What is $1 * a$? (Here we interpret 1 as a 1-vector.)

(d) Convolution with a unit vector. Let $e_{k,q}$ denote the $k$th unit vector of dimension $q$. What is $e_{k,q} * a$? Describe this vector mathematically (i.e., give its entries), and via a brief English description. You might find vector slice notation useful.

7.3 Equalization in communication. A communication system (such as a wireless, wired, or fiber optic link) is used to transfer a message, given as a Boolean $N$-vector $s$, to a receiver, over a channel. Boolean means that the entries of $s$ are all either 0 or 1. (Another common choice is $-1$ or $+1$.)

The entries of $s$ are called symbols; the set of possible values, in this case, 0 and 1, is called the symbol constellation.

Convolution is a very common model of the effect of the channel on the transmitted message $s$. This means that the receiver gets the received signal $y = c * s$, where $c$ is an $n$-vector called the channel impulse response, and $n$ is called the channel length. In typical channels $n$ is small, say, no more than 10, and $c_1 \approx 1$ is the largest entry. When the coefficients $c_2, \ldots, c_n$ are nonzero, $y_i$ depends not just on $s_i$, but also $s_{i-1}, \ldots, s_{i-n+1}$. This effect is called inter-symbol interference (ISI).

A common method to estimate or guess the transmitted signal is $\hat{s} = \text{round}(y_{1:N})$, where the round function rounds each entry to 0 or 1 (with threshold $1/2$). This method works well when the channel impulse response satisfies $c \approx e_1$, since then we have $y \approx (s, 0_{n-1})$, and so $\hat{s}_i = s_i$, after rounding. When $\hat{s}_i \neq s_i$, we say that a bit error has occurred in the transmission of bit $i$. The bit error rate (BER) is the number of bit errors, divided by $N$, the message length.

Equalization is a widely used method to combat the effects of ISI. An $n$-vector $h$ (called the equalizer impulse response) is chosen or designed, so that $h * c \approx e_{1,2n-1}$. (The vector $h * c$ is sometimes
called the *equalized channel impulse response.* The receiver then forms the vector \( \tilde{y} = h \ast y \). This means that

\[
\tilde{y} = h \ast y = h \ast (c \ast s) = (h \ast c) \ast s \approx e_{1,2n-1} \ast s = (s_{1:N}, 0_{2n-2}).
\]

It follows that the signal can be decoded at the receiver using \( \hat{s}^{eq} = \text{round}(\tilde{y}_{1:N}) \). The interpretation is that the equalizer ‘undoes’ the affect of the channel.

Run the file `channel_equalization_data.jl`, which will define a message \( s \), a channel \( c \), and an equalizer \( h \). (Your are welcome to look inside the file to see how we designed the equalizer.)

Plot \( c \), \( h \), and \( h \ast c \). Make a brief comment about the channel and equalized channel impulse responses.

Plot \( s \), \( y \), and \( \tilde{y} \) over the index range \( i = 1, \ldots, 100 \). Is it clear from this plot that \( \hat{s} = \text{round}(y_{1:N}) \) will be worse estimate of \( s \) than \( \hat{s}^{eq} = \text{round}(\tilde{y}_{1:N}) \)?

Report the BER for \( \hat{s} \) (estimating the message without equalization), and for \( \hat{s}^{eq} \) (estimating the message with equalization).

*Hint:* To round a real vector \( x \) to \( \{0, 1\} \) in Julia you can use \( (x \ast 0.5) \), which yields a Boolean vector. You can convert it to an integer vector (say, for plotting) using \( \text{int}(x \ast 0.5) \).

7.4 *Convolution in Julia.* Use Julia’s \texttt{conv()} function to find the coefficients of the polynomial \((1 - x + 2x^2)^4\). *Hint.* Convolution gives the coefficients of the product of two polynomials.

7.5 *Audio filtering.* When the vector \( x \) represents an audio signal, and \( h \) is another (usually much shorter) vector, the convolution \( y = h \ast x \) is called the *filtered* version of \( x \), and \( h \) is called the *filter impulse response.* Filters can be used to smooth out audio signals (which reduces high frequency sounds and enhances low frequency sounds), or to sharpen them (which enhances high frequency sounds and reduces low frequency sounds), as in audio bass and treble tone controls. In this problem you will experiment with, and listen to, the effects of several audio filters.

The file `audio_filtering_original.wav` contains a 10-second recording with sample rate of \( f = 44100/\text{sec} \). We let \( x \) denote the 441000-vector representing this recording. You can read in \( x \) and the sample rate \( f \) using the following code:

Pkg.add("WAV")
using WAV
x, f = wavread("audio_filtering_original.wav");
x = vec(x);

To play the signal, run:

\texttt{wavplay(x, f)};

If this not supported on your system, you can write the signal into a file, download the file from JuliaBox if you are using that, and then listen to it on your machine:

\texttt{wavwrite(x, f, "filename.wav")};

(a) *1ms smoothing filter.* Let \( h^{\text{smooth}} \) be the 44-vector \( h^{\text{smooth}} = \frac{1}{44} \mathbf{1}_{44} \). (The subscript 44 gives the length of the vector.) The signal \( h^{\text{smooth}} \ast x \) is the 1ms moving average of the input \( x \). We can construct the vector \( h^{\text{smooth}} \) and compute the output signal as follows:
h_smooth = 1 / 44 * ones(44);
output = conv(h_smooth, x);
wavplay(output, f);

Listen to the output signal and briefly describe the effect of convolving $h^{\text{smooth}}$ with $x$ in one sentence.

(b) **Echo filter.** What filter (i.e., vector) $h^{\text{echo}}$ has the property that $h^{\text{echo}} \ast x$ consists of the original recording, plus an echo of the original recording 0.25 seconds delayed, with half the original amplitude? Since sound travels at about 340m/s, this is equivalent to the effect of hearing an echo from a wall about 42.5m away. Construct $h^{\text{echo}}$ using Julia and listen to the output signal $h^{\text{echo}} \ast x$ to confirm the effect. Form and listen to the signal $h^{\text{echo}} \ast h^{\text{echo}} \ast x$ and very briefly describe what you hear.

*Hint.* The entries of the output signal $y = h^{\text{echo}} \ast x$ satisfy $y_i = x_i + 0.5x_{i-k}$, where we take $x_j = 0$ for $j$ outside the range $1, \ldots, 441000$, and $k$ is the number of samples in 0.25 seconds.

7.6 **Another property of convolution.** Suppose that $a$ is an $n$-vector and $b$ is an $m$-vector that satisfy $a \ast b = 0$. Is it true that either $a = 0$ or $b = 0$? If yes, explain why. If no, give a specific example of $a$ and $b$, not both zero, with $a \ast b = 0$.

7.7 **Social network graph.** Consider a group of $n$ people or users, and some symmetric social relation among them. This means that some pairs of users are connected, or friends (say). We can create a directed graph by associated a node with each user, and an edge between each pair of friends, arbitrarily choosing the direction of the edge. Now consider an $n$-vector $v$, where $v_i$ is some quantity for user $i$, for example, age or education level (say, given in years). Let $L(v)$ denote the Laplacian associated with the graph and $v$, thought of as a potential on the nodes.

(a) Explain why the number $L(v)$ does not depend on the choice of directions for the edges of the graph.

(b) Would you guess that $L(v)$ is small or large? This is an open-ended, vague question; there is no right answer. Just make a guess as to what you might expect, and give a short English justification of your guess.
8 Linear equations

8.1
9 Linear dynamical systems

9.1 Dynamics of a compartmental system. A compartmental system is a model used to describe the movement of some material over time among a set of $n$ compartments of a system, and the outside world. It is widely used in pharmaco-kinetics, the study of how the concentration of a drug varies over time in the body. In this application, the material is a drug, and the compartments are the bloodstream, lungs, heart, liver, kidneys, and so on.

We have $n$ compartments, and we let the $n$-vector $x_t$, for $t = 1, 2, \ldots$, denote the vector of concentrations or amounts of the material in the compartments in time period $t$, so $(x_t)_i$ is the amount of the material in compartment $i$ in time period $t$. These amounts are typically, but not always, nonnegative.

We have a set of directed edges with positive weights that connect (different) compartments, or the compartments and the outside world. We let $w_{ij}$ denote the weight of the edge going from compartment $j$ to compartment $i$. If there is no edge from compartment $j$ to compartment $i$, we let $w_{ij} = 0$. We can also have edges that go from compartments to the outside world. We express these (nonnegative) edge weights as $w_{0j}$. (In the pharmaco-kinetics application, these correspond to the drug being eliminated or absorbed or otherwise rendered inactive.)

Material flows from one compartment to another, or to the outside world, over each time period, along the edges. Between time period $t$ and $t + 1$, an amount of material $w_{ij}(x_t)_j$ moves along the edge from compartment $j$ to compartment $i$. This reduces the amount of material in compartment $j$ and increases it by the same amount at compartment $i$. An amount of $w_{0j}(x_t)_j$ of the material is simply removed from node $j$ between time $t$ and $t + 1$. A standard assumption is that for each compartment, the sum of the weights on outgoing edges is no more than one. (What would it mean if this were not true?)

Compartmental systems are special cases of linear dynamical systems; that is, we have $x_{t+1} = Ax_t$ for some $n \times n$ matrix $A$.

A compartmental system with $n = 3$ compartments is shown below. It is a simple pharmaco-kinetic model, with compartment 1 being the bloodstream, and the periods representing 15 minute intervals (say).

![Diagram of a compartmental system with nodes 1, 2, 3 and edges with weights 0.10, 0.05, and 0.05.]

(a) For the specific compartmental system shown above, find the matrix $A$ for which $x_{t+1} = Ax_t$. Be sure to account for all movement of the drug, out of and into each compartment, and elimination of the drug from the body.

(b) Assume that $x_1 = e_1$, which means we give a unit dose of the drug intravenously at period $t = 1$. Use Julia to plot the amounts of the drug in each of the components, on the same graph, for...
You can use the function `plot_compartments` in `compartmental_system.jl` to create your plot. The function accepts a $3 \times 100$ matrix, where the $i$th row of the matrix is the amount of drug in compartment $i$ for $t = 1, \ldots, 100$.

(c) Use Julia to plot the total amount of the drug in the body (the 3 compartments) versus $t$, for $t = 1, \ldots, 100$. How many periods does it take for the total amount of the drug in the body to drop below $1/3$ of its initial value? You can use the function `plot_totals` in `compartmental_system.jl` to create your plot. The function accepts a 100-vector that contains the total amount of drug in the body for $t = 1, \ldots, 100$.

9.2 Dynamics of an economy. An economy (of a country or region) is described by an $n$-vector $a_t$, where $(a_t)_i$ is the economic output in sector $i$ in year $t$ (measured in billions of dollars, say). The total output of the economy in year $t$ is $a_T^T a_t$. A very simple model of how the economic output changes over time is $a_{t+1} = B a_t$, where $B$ is an $n \times n$ matrix. (This is closely related to the Leontief input-output model described in §8.3 of the book. But the Leontief model is static, i.e., doesn’t consider how an economy changes over time.) The entries of $a_t$ and $B$ are positive in general.

In this problem we will consider the specific model with $n = 4$ sectors and

\[
B = \begin{bmatrix}
0.8 & 0.1 & 0.6 & 0.2 \\
0.3 & 0.5 & 0.1 & 0.1 \\
0.2 & 0.1 & 0.5 & 0.2 \\
0.6 & 0.1 & 0.1 & 0.4
\end{bmatrix}.
\]

(a) Briefly interpret $B_{23}$, in English.

(b) Simulation. Suppose $a_1 = (1.3, 0.2, 0.8, 0.2)$. Simulate the sector outputs for $t = 1, \ldots, 20$. Plot the sector outputs, and the total economic output over this time range.
10 Matrix multiplication

10.1 Matrix sizes. Suppose $A$, $B$, and $C$ are matrices that satisfy $A + BB^T = C$. Determine which of the following statements are necessarily true. (There may be more than one true statement.)

(a) $A$ is square.
(b) $A$ and $B$ have the same dimensions.
(c) $A$, $B$, and $C$ have the same number of rows.
(d) $B$ is a tall matrix.

10.2 Matrix power identity. A student says that for any square matrix $A$,

$$(A + I)^3 = A^3 + 3A^2 + 3A + I.$$ 

Is she right? If she is, explain why; if she is wrong, give a specific counterexample, i.e., a square matrix $A$ for which it does not hold.

10.3 Matrix equations. Consider two $m \times n$ matrices $A$ and $B$. Suppose that for $j = 1, \ldots, n$, the $j$th column of $A$ is a linear combination of the first $j$ columns of $B$. How do we express this as a matrix equation? Choose one of the matrix equations below.

- $A = GB$ for some upper triangular matrix $G$.
- $A = BH$ for some upper triangular matrix $H$.
- $A = FB$ for some lower triangular matrix $F$.
- $A = BJ$ for some lower triangular matrix $J$.

10.4 Matrix cancellation. Suppose $AX = AY$ for matrices $A$, $X$, and $Y$. If these are scalars (i.e., $1 \times 1$ matrices), we can conclude that $X = Y$ provided $A \neq 0$. What condition on $A$ is needed in the general (matrix) case to conclude that $X = Y$? Choose the correct response.

(a) $A \neq 0$.
(b) All entries of $A$ are nonzero.
(c) The columns of $A$ are independent.
(d) The rows of $A$ are independent.
(e) $A$ is square.

10.5 Multiplication by a diagonal matrix. Suppose that $A$ is an $n \times n$ matrix, $D$ is a diagonal matrix, and $B = DA$. Choose one of the following.

- The $i$th column of $B$ is the $i$th column of $A$, scaled by $D_{ii}$.
- The $i$th row of $B$ is the $i$th row of $A$, scaled by $D_{ii}$.
- $B$ is the same as $A$, except for the diagonal entries which satisfy $B_{ii} = D_{ii}A_{ii}$.
- None of the above choices are correct.

10.6 Suppose $A$ is an $n \times n$ matrix that satisfies $A^2 = 0$. Does this imply that $A = 0$? (This is the case when $n = 1$.) If this is (always) true, explain why. If it is not, give a specific counterexample, i.e., a matrix $A$ that is nonzero but satisfies $A^2 = 0$. 

28
10.7 Choose one of the responses *Always*, *Never*, or *Sometimes* for each of the statements below. ‘Always’ means the statement is always true, ‘Never’ means it is never true, and ‘Sometimes’ means it can be true or false, depending on the particular values of the matrix or matrices.

(a) An upper triangular matrix has linearly independent columns.
(b) The rows of a tall non-square matrix are linearly dependent.
(c) The product of two lower triangular matrices is lower triangular.
(d) The product of two orthogonal matrices is orthogonal.

10.8 *Patients and symptoms*. Each of a set of $N$ patients can exhibit any number of a set of $n$ symptoms. We express this as an $N \times n$ matrix $S$, with

$$S_{ij} = \begin{cases} 1 & \text{patient } i \text{ exhibits symptom } j \\ 0 & \text{patient } i \text{ does not exhibit symptom } j. \end{cases}$$

Give simple English descriptions of the following expressions. Include the dimensions, and describe the entries.

(a) $S1$.
(b) $S^T1$.
(c) $S^T S$.
(d) $SS^T$.
(e) $\|s_i - s_j\|^2$, where $s_k^T$ is the $k$th row of $S$.

10.9 *Orthogonal matrices*. Let $U$ and $V$ be two orthogonal $n \times n$ matrices.

(a) Show that the matrix $U^T$ is orthogonal.
(b) Show that the matrix $UV$ is orthogonal.
(c) Show that the $2n \times 2n$ matrix

$$\frac{1}{\sqrt{2}} \begin{bmatrix} U & U \\ V & -V \end{bmatrix}$$

is orthogonal.

10.10 *Orthogonal $2 \times 2$ matrices*. In this problem, you will show that every $2 \times 2$ orthogonal matrix is either a rotation or a reflection (see §7.1 of the book).

(a) Let

$$Q = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

be an orthogonal $2 \times 2$ matrix. Show that the following equations hold:

$$a^2 + b^2 = 1, \quad a^2 + c^2 = 1, \quad b^2 + d^2 = 1, \quad c^2 + d^2 = 1,$$

$$ac + bd = 0, \quad ab + cd = 0.$$

*Hint*. Consider $Q^T Q$ and $QQ^T$. 

29
(b) Whenever we have an equation of the form $x^2 + y^2 = 1$, we can find an angle $\theta$ such that $x = \cos \theta$ and $y = \pm \sin \theta$ (i.e., either $y = \sin \theta$ or $y = -\sin \theta$). Using this fact and part (a), show that we can find $\theta$ such that

$$a = \cos \theta, \quad b = \pm \sin \theta, \quad c = \pm \sin \theta, \quad d = \pm \cos \theta.$$

(c) Suppose we pick $d = -\cos \theta$. What are the two possible matrices we get for $Q$?

(d) Given a 2-vector $x$, let’s try to visualize the effects of transforming $x$ by $Q$. That is, let’s visualize what changes between $x$ and $Qx$. Compute the numerical value for one of the matrices $Q$ you obtained in part (c) for $\theta = \pi$. Pass your $Q$ to the plot_orthogonal_transform function provided in orthogonal_transform.jl. Give a one sentence interpretation of what is happening in the plot, and include a printout or drawing of the plot. The following code gives an example of how to use the plot_orthogonal_transform function.

```julia
include("orthogonal_transform.jl")
Q = [cos(pi) 0; 0 cos(pi)]
plot_orthogonal_transform(Q)
```

(e) Suppose we pick $d = \cos \theta$. What are the two possible matrices we get for $Q$?

(f) Repeat part (d) using one of the two matrices you obtained in (e) and $\theta = \pi/2$. Give a one sentence interpretation of what is happening in the plot, and include a printout or drawing of the plot.

10.11 Matrix multiplication Julia timing test. Determine how long it takes your computer to compute the product of two $n \times n$ matrices for $n = 500, 1000, 2000, 4000$, and use your result for $n = 4000$ to estimate (very crudely) how many Gflops/sec your computer can carry out. (Hopefully your results for the different values of $n$ will give roughly consistent estimates of computer speed.)

The follow code generates two random $500 \times 500$ matrices and times the evaluation of their product. (You might run it a few times; the first time might be a bit slower, since the matrix multiplication code has to be loaded and compiled.)

```julia
A = randn(500,500); B = randn(500,500);
tic(); C=A*B; toc();
```

How long would it take a person to carry this out, assuming the person can carry out a floating point operation every 10 seconds for 8 hours each day?

10.12 Dynamics of an economy. Let $x_1, x_2, \ldots$ be $n$-vectors that give the level of economic activity of a country in years 1, 2, \ldots, in $n$ different sectors (like energy, defense, manufacturing). Specifically, $(x_t)_i$ is the level of economic activity in economic sector $i$ (say, in billions of dollars) in year $t$. A common model that connects these economic activity vectors is $x_{t+1} = Ax_t$, where the $n \times n$ matrix $A$ is called the input-output matrix for the economy.

Give a matrix expression for the total economic activity across all sectors in year $t = 11$.

10.13 Students, classes, and majors. We consider $m$ students, $n$ classes, and $p$ majors. Each student can be in any number of the classes (although we’d expect the number to range from 3 to 6), and can
have any number of the majors (although the common values would be 0, 1, or 2). The data about
the students’ classes and majors are given by an \( m \times n \) matrix \( C \) and an \( m \times p \) matrix \( M \), where

\[
C_{ij} = \begin{cases} 
1 & \text{student } i \text{ is in class } j \\
0 & \text{student } i \text{ is not in class } j,
\end{cases}
\]

and

\[
M_{ij} = \begin{cases} 
1 & \text{student } i \text{ is in major } j \\
0 & \text{student } i \text{ is not in major } j.
\end{cases}
\]

(a) Let \( E \) be the \( n \)-vector with \( E_i \) being the enrollment in class \( i \). Express \( E \) using matrix notation, in terms of the matrices \( C \) and \( M \).

(b) Define the \( n \times p \) matrix \( S \) where \( S_{ij} \) is the total number of students in class \( i \) with major \( j \). Express \( S \) using matrix notation, in terms of the matrices \( C \) and \( M \).

10.14 Customer purchase history matrix. A store keeps track of its sales of products from \( K \) different
product categories to \( N \) customers over some time period, like one month. (While it doesn’t matter
for this problem, \( K \) might be on the order of 1000 and \( N \) might be 100000.) The data is stored in
an \( N \times K \) matrix \( C \), with \( C_{ij} \) being the total dollar purchases of product \( j \) by customer \( i \). All the
entries of \( C \) are nonnegative. The matrix \( C \) is typically sparse, i.e., many of its entries are zero.

(a) What is \( C1 \)?

(b) What is \( C^T1 \)?

(c) Give a short matrix-vector expression for the total dollar amount of all purchases, by all
customers.

(d) What does it mean if \( (CC^T)_{kl} = 0 \)? Your answer should be simple English.

(e) Suppose you run \( k \)-means on the rows of \( C \), with \( k = 100 \). How would you interpret the
centroids \( z_1, \ldots, z_{100} \)?

10.15 Student group membership. Let \( G \in \mathbb{R}^{m \times n} \) represent a contingency matrix of \( m \) students who are
members of \( n \) groups:

\[
G_{ij} = \begin{cases} 
1 & \text{student } i \text{ is in group } j \\
0 & \text{student } i \text{ is not in group } j.
\end{cases}
\]

(A student can be in any number of the groups.)

(a) What is the meaning of the 3rd column of \( G \)?

(b) What is the meaning of the 15th row of \( G \)?

(c) Give a simple formula (using matrices, vectors, etc.) for the \( n \)-vector \( M \), where \( M_i \) is the total
membership (i.e., number of students) in group \( i \).

(d) Interpret \( (GG^T)_{ij} \) in simple English.

(e) Interpret \( (G^TG)_{ij} \) in simple English.

10.16 Choose one of the responses always, never, or sometimes for each of the statements below. ‘Always’
means the statement is always true, ‘never’ means it is never true, and ‘Sometimes’ means it can be
true or false, depending on the particular values of the matrix or matrices. Give a brief justification
of each answer.
10.17 State feedback control. Consider a time-invariant linear dynamical system with \( n \)-vector state \( x_t \) and \( m \)-vector input \( u_t \), with dynamics

\[
x_{t+1} = Ax_t + Bu_t, \quad t = 1, 2, \ldots
\]

The entries of the state often represent deviations of \( n \) quantities from their desired values, so \( x_t \approx 0 \) is a goal in operation of the system. The entries of the input \( u_t \) are deviations from the standard or nominal values. For example, in an aircraft model, the states might be the deviation from the desired altitude, climb rate, speed, and angle of attack; the input \( u_t \) represents changes in the control surface angles or engine thrust from their normal values.

In state feedback control, the states are measured and the input is a linear function of the state, \( u_t = Kx_t \). The \( m \times n \) matrix \( K \) is called the state feedback gain matrix. The state feedback gain matrix is very carefully designed, using several methods. State feedback control is very widely used in many application areas (including, for example, control of airplanes).

(a) Open and closed-loop dynamical system. With \( u_t = 0 \), the system satisfies \( x_{t+1} = Ax_t \), which is called the open-loop dynamics. When \( u_t = Kx_t \), the system dynamics can be expressed as \( x_{t+1} = \tilde{A}x_t \), where the matrix \( \tilde{A} \) is the closed-loop dynamics matrix. Find an expression for \( \tilde{A} \) in terms of \( A \), \( B \), and \( K \).

(b) Aircraft control. The longitudinal dynamics of a 747 flying at 40000 ft at Mach 0.81 is given by

\[
A = \begin{bmatrix}
.99 & .03 & -0.02 & -0.32 \\
.01 & .47 & 4.7 & .00 \\
.02 & -0.06 & .40 & -0.00 \\
.01 & -0.04 & .72 & .99
\end{bmatrix}, \quad B = \begin{bmatrix}
0.01 & 0.99 \\
-3.44 & 1.66 \\
-0.83 & 0.44 \\
-0.47 & 0.25
\end{bmatrix},
\]

where the sampling time is one second. (The state and control variables are described in more detail in the lecture on control.) We will use the state feedback matrix

\[
K = \begin{bmatrix}
-0.038 & 0.021 & 0.319 & -0.270 \\
-0.061 & -0.004 & -0.120 & 0.007
\end{bmatrix}.
\]

(The matrices \( A \), \( B \), and \( K \) can be found in 747_cruise_dyn_data.jl, so you don’t have to type them in.) Plot the open-loop and closed-loop state trajectories from several nonzero initial states, such as \( x_1 = (1, 0, 0, 0) \), or ones that are randomly generated, from \( t = 1 \) to \( t = 100 \) (say). Would you rather be a passenger in the plane with the state feedback control turned off \( (i.e., \text{open-loop}) \) or on \( (i.e., \text{closed-loop}) \)?
11 Matrix inverses

11.1 Properties of pseudo-inverses. For an \( m \times n \) matrix \( A \) and its pseudo-inverse \( A^\dagger \), show that \( A = AA^\dagger A \) and \( A^\dagger = A^\dagger AA^\dagger \) in each of the following cases.

(a) \( A \) is tall with linearly independent columns.
(b) \( A \) is wide with linearly independent rows.
(c) \( A \) is square and invertible.

11.2 Inverse of a block matrix. Consider the \( (n+1) \times (n+1) \) matrix

\[ A = \begin{bmatrix} I & a \\ a^T & 0 \end{bmatrix}, \]

where \( a \) is an \( n \)-vector.

(a) When is \( A \) invertible? Give your answer in terms of \( a \). Points will be deducted from answers that are correct, but more complicated than needed. Justify your answer.

(b) Assuming the condition you found in part (a) holds, give an expression for the inverse matrix \( A^{-1} \). Answers that are correct but too complicated will be penalized.

11.3 Inverse of a block matrix. Let \( B \) and \( D \) be invertible matrices of sizes \( m \times m \) and \( n \times n \), respectively, and let \( C \) be any \( m \times n \) matrix. Find the inverse of the block matrix

\[ A = \begin{bmatrix} B & C \\ 0 & D \end{bmatrix} \]

in terms of \( B^{-1} \), \( C \), and \( D^{-1} \).

Hints.

- Your goal is to find matrices \( W \), \( X \), \( Y \), and \( Z \) (in terms of \( B^{-1} \), \( C \), and \( D^{-1} \)) that satisfy

\[ A \begin{bmatrix} W & X \\ Y & Z \end{bmatrix} = I. \]

Use block matrix multiplication to express this as a set of four matrix equations that you can then solve.

- You can get an idea of what the solution should look like by considering the case when \( B \), \( C \), and \( D \) are scalars.

11.4 Affine combinations of left inverses. Let \( Z \) be a tall \( m \times n \) matrix with linearly independent columns, and let \( X \) and \( Y \) be left inverses of \( Z \). Show that for any scalars \( \alpha \) and \( \beta \) satisfying \( \alpha + \beta = 1 \), \( \alpha X + \beta Y \) is also a left inverse of \( Z \).

11.5 Rows and columns of a matrix and its inverse. Suppose the \( n \times n \) matrix \( A \) is invertible, with inverse \( B = A^{-1} \). We let the \( n \)-vectors \( a_1, \ldots, a_n \) denote the columns of \( A \), and \( b_1^T, \ldots, b_n^T \) the rows of \( B \). Determine whether each of the following statements is true or false, and justify your answer. True means the statement always holds, with no further assumptions. False means the statement does not always hold, without further assumptions.
(a) For any $n$-vector $x$, we have $x = \sum_{i=1}^{n} (b_i^T x) a_i$.

(b) For any $n$-vector $x$, we have $x = \sum_{i=1}^{n} (a_i^T x) b_i$.

(c) For $i \neq j$, $a_i \perp b_j$.

(d) For any $i$, $\|b_i\| \geq 1/\|a_i\|$.

(e) For any $i$ and $j$, $b_i + b_j \neq 0$.

(f) For any $i$, $a_i + b_i \neq 0$.

11.6 Reverse-time linear dynamical system. A linear dynamical system has the form

$$x_{t+1} = Ax_t,$$

where $x_t$ in the $(n$-vector$)$ state in period $t$, and $A$ is the $n \times n$ dynamics matrix. This formula gives the state in the next period as a function of the current state.

We want to derive a recursion of the form

$$x_{t-1} = A^{rev} x_t,$$

which gives the previous state as a function of the current state. We call this the reverse time linear dynamical system.

(a) When is this possible? When it is possible, what is $A^{rev}$?

(b) For the specific linear dynamical system with dynamics matrix

$$A = \begin{bmatrix} 3 & 2 \\ -1 & 4 \end{bmatrix},$$

find $A^{rev}$, or explain why the reverse time linear dynamical system doesn’t exist.

11.7 Solving linear equations in Julia. Generate a random $20 \times 20$ matrix $A$ and a random 20-vector $b$ using the following code:

```julia
A = rand(20, 20)
b = rand(20)
```

(It’s very likely that the matrix you generate will be invertible.)

We solve the linear equation $Ax = b$, i.e., compute the solution $x = A^{-1} b$ in Julia using several methods. In each case, you should check that the $x$ you compute satisfies the equations by evaluating and reporting the norm of the residual, $\|Ax - b\|$. (This should be very small.)

(a) Using the backslash operator:

```julia
x = A \ b
```

(b) Computing the inverse of $A$ explicitly:

```julia
x = inv(A) * b
```
(c) Using QR factorization, from the formula \( x = R^{-1}Q^Tb \):

\[
Q, R = \text{qr}(A) \\
x = R \backslash (Q' * b)
\]

(You should check that the matrix \( Q \) obtained is very nearly orthogonal, \( R \) is an upper triangular matrix, and that \( A \) is very near \( QR \).

11.8 Julia timing test for linear equations.

(a) Determine how long it takes for your computer to solve a system of \( n = 2000 \) linear equations in \( n = 2000 \) variables (with invertible coefficient matrix) using Julia’s \( \backslash \) operator. You may use the following code.

\[
A = 1 + \text{rand}(2000, 2000) \\
b = \text{ones}(2000) \\
@time A\backslash b;
\]

(b) Julia is rather clever about how it solves systems of equations with \( \backslash \). Determine how long it takes for your computer to solve the following system of \( n = 2000 \) linear equations in \( n = 2000 \) variables.

\[
L = 1 + \text{rand}(2000, 2000) \\
\text{for } i = 1:2000 \\
    \text{for } j = i+1:2000 \\
        L[i, j] = 0 \\
    \end{eqnarray}
\]

\[
b = \text{ones}(2000) \\
@time L\backslash b;
\]

(c) Can you explain why the times differ by so much between the two systems, \( i.e. \), what is special about the matrix \( L \) as opposed to \( A \)? Make a hypothesis about what you think Julia is doing behind the scenes.

11.9 Sensitivity of solution of linear equations. Let \( A \) be an invertible \( n \times n \) matrix, and \( b \) and \( x \) be \( n \)-vectors satisfying \( Ax = b \). Suppose we now perturb the \( j \)th entry \( b \) by \( \epsilon \neq 0 \) (which is a traditional symbol for a small quantity), which means that \( b \) becomes \( \tilde{b} = b + \epsilon e_j \). Let \( \tilde{x} \) be the \( n \)-vector that satisfies \( A\tilde{x} = \tilde{b} \), \( i.e. \), the solution of the linear equations using the perturbed right-hand side. We are interested in \( \|x - \tilde{x}\| \), which is how much the solution changes due to the change in the right-hand side.

(a) Show that \( \|x - \tilde{x}\| \) does not depend on \( b \); it only depends on the matrix \( A \), \( \epsilon \), and \( j \).

(b) How would you find the index \( j \) that maximizes the value of \( \|x - \tilde{x}\| ? \) By part (a), your answer should be in terms of \( A \) (or quantities derived from \( A \)) and \( \epsilon \) only.

(c) Try this out in Julia with the following values:

\[
A = \begin{bmatrix}
1/2 & 1/3 & 1/4 \\
1/3 & 1/4 & 1/5 \\
1/4 & 1/5 & 1/6
\end{bmatrix}, \quad b = \begin{bmatrix}
1 \\
1 \\
1
\end{bmatrix}, \quad \epsilon = 0.1.
\]
To prevent numerical imprecision errors, use the following code to build the data.

\[
A = \begin{bmatrix}
1/2 & 1/3 & 1/4; \\
1/3 & 1/4 & 1/5; \\
1/4 & 1/5 & 1/6
\end{bmatrix}
\]

\[
b = [1.0, 1.0, 1.0]
\]

\[
epsilonilon = 0.1
\]

Which \( j \) do you pick to maximize \( \|x - \tilde{x}\| \), and what value do you get for \( \|x - \tilde{x}\| \)? Check your answer from part (b) by direct calculation (i.e., simply finding \( \tilde{x} \) after perturbing entry \( j = 1, 2, 3 \) of \( b \)).

11.10 Interpolation of rational functions. In this problem, you will find a rational function (i.e., ratio of polynomials)

\[
f(t) = \frac{c_0 + c_1 t + c_2 t^2}{1 + d_1 t + d_2 t^2}
\]

that satisfies the following (interpolation) conditions:

\[
f(1) = 2, \quad f(2) = 5, \quad f(3) = 9, \quad f(4) = -1, \quad f(5) = -4.
\]

Your job is to find numbers \( c_0, c_1, c_2, d_1 \) and \( d_2 \) for which these conditions hold.

(a) Let \( x = (c_0, c_1, c_2, d_1, d_2) \). Explain how to formulate this problem as finding a vector \( x \) that satisfies a system of linear equations \( Ax = b \). Be sure to specify what \( A \) and \( b \) are.

(b) Solve the system of linear equations in Julia to find the coefficients. What do you get for your vector \( x \)? Please include your code as part of your answer.

(c) Using MMAPlot, plot your rational function in Julia as a line plot and the 5 given points as a scatter plot. You can use the following code to produce the data needed for plotting, given that you have a vector \( x \) from part (b) ready to go.

\[
t = -2:0.01:8
\]

\[
f = (x[1] + x[2]*t + x[3]*t.^2) ./ (1 + x[4]*t + x[5]*t.^2)
\]

\[
points_t = [1, 2, 3, 4, 5]
\]

\[
points_f = [2, 5, 9, -1, -4]
\]

Please include your code and plots as part of your answer.

11.11 Combinations of invertible matrices. Suppose the \( n \times n \) matrices \( A \) and \( B \) are both invertible. Determine whether each of the statements below is true or false. True means that the statement must hold, for any invertible matrices \( A \) and \( B \). False means that there are invertible matrices \( A \) and \( B \) for which the statement is false.

(a) \( [A B] \) has linearly independent columns.

(b) \( [A B] \) has linearly independent rows.

(c) \( A + B \) is invertible.

(d) \( \begin{bmatrix} A & A+B \\ 0 & B \end{bmatrix} \) is invertible.
11.12 Another left inverse. Suppose the \( m \times n \) matrix \( A \) is tall, and has independent columns. One left inverse of \( A \) is \( A^\dagger \), its pseudo-inverse. In this problem we explore another one. Write \( A \) as the block matrix \( A = \begin{bmatrix} A_1 & A_2 \end{bmatrix} \), where \( A_1 \) is \( n \times n \). We will assume that \( A_1 \) is invertible (which need not happen in general).

(a) Show that \( \tilde{A} = \begin{bmatrix} A_1^{-1} & 0_{n \times m-n} \end{bmatrix} \) is a left inverse of \( A \).

(b) Let \( b \) be an \( m \)-vector, and let \( \tilde{x} = \tilde{A}b \). What can you say about the associated residual vector \( A\tilde{x} - b \)?

11.13 Middle inverse. Suppose \( A \) is an \( n \times p \) matrix and \( B \) is a \( q \times n \) matrix. If a \( p \times q \) matrix \( X \) exists that satisfies \( AXB = I \), we call it a middle inverse of the pair \( A, B \). (This is not a standard concept.) Note that when \( A \) or \( B \) is an identity matrix, the middle inverse reduces to the right or left inverse, respectively.

(a) Describe the conditions on \( A \) and \( B \) under which a middle inverse \( X \) exists. Give your answer using only the following four concepts: Independence of the rows or columns of \( A \), and independence of the rows or columns of \( B \). You must justify your answer.

(b) Give an expression for a middle inverse, assuming the conditions in part (a) hold.

11.14 Invertibility of population dynamics matrix. Consider the population dynamics matrix

\[
A = \begin{bmatrix}
  b_1 & b_2 & \cdots & b_{99} & b_{100} \\
  1 - d_1 & 0 & \cdots & 0 & 0 \\
  0 & 1 - d_2 & \cdots & 0 & 0 \\
  \vdots & \vdots & \ddots & \vdots & \vdots \\
  0 & 0 & \cdots & 1 - d_{99} & 0 
\end{bmatrix},
\]

where \( b_i \geq 0 \) are the birth rates and \( 0 \leq d_i \leq 1 \) are death rates. What are the conditions on \( b_i \) and \( d_i \) under which \( A \) is invertible? (If the matrix is never invertible or always invertible, say so.) Justify your answer.
12 Least squares

12.1 Weighted least-squares. In least squares, the objective (to be minimized) is
\[ \|Ax - b\|^2 = \sum_{i=1}^{m} (\tilde{a}_i^T x - b_i)^2, \]
where \( \tilde{a}_i \) are the rows of \( A \), and the \( n \)-vector \( x \) is to chosen. In the weighted least-squares problem, we minimize the objective
\[ \sum_{i=1}^{m} w_i (\tilde{a}_i^T x - b_i)^2, \]
where \( w_i \) are given positive weights. The weights allow us to assign different weights to the different components of the residual vector.

(a) Show that the weighted least-squares objective can be expressed as \( \|D(Ax - b)\|^2 \) for an appropriate diagonal matrix \( D \). This allow us to solve the weighted least-squares problem as a standard least-squares problem, by minimizing \( \|Ax - \tilde{b}\|^2 \), where \( \tilde{A} = DA \) and \( \tilde{b} = Db \).

(b) Show that when \( A \) has independent columns, so does the matrix \( \tilde{A} \).

(c) The least-squares approximate solution is given by \( \hat{x} = (A^T A)^{-1} A^T b \). Give a similar formula for the solution of the weighted least-squares problem. You might want to use the matrix \( W = \text{diag}(w) \) in your formula.

12.2 Solving least-squares problems in Julia. Generate a random 20 × 10 matrix \( A \) and a random 20-vector \( b \).

(a) Compute the solution \( \hat{x} \) of the associated least-squares problem using the methods listed below, and verify that the solutions found are the same, or more accurately, very close to each other; they will be very slightly different due to small roundoff errors in the computations.

- Using the Julia backslash operator.
- Using \( \hat{x} = (A^T A)^{-1} A^T b \).
- Using \( \hat{x} = A\backslash b \).

Hints. In Julia, \( \text{inv()} \) computes the inverse matrix, \( \text{pinv()} \) computes the pseudo-inverse matrix, and \( A\backslash b \) directly solves the least-squares problem.

(b) Let \( \hat{x} \) be one of the solutions found in part (a). Generate a random nonzero \( n \)-vector \( \delta \) and verify that \( \|A(\hat{x} + \delta) - b\|^2 > \|A\hat{x} - b\|^2 \). Repeat several times with different values of \( \delta \); you might try choosing a small \( \delta \) (say, by scaling the original random vector).

Be sure to submit your code, including the code that checks if the solutions in part (a) are close to each other, and whether the expected inequality in part (b) holds.

12.3 Julia timing test for least-squares. Determine how long it takes for your computer to solve a least-squares problem with \( m = 100000 \) equations and \( n = 100 \) variables. (You can use the backslash operator.)

Remark. Julia compiles just in time, so you should run the code a few times to get the correct time.
12.4 Approximate right inverse. Suppose the tall \( m \times n \) matrix \( A \) has independent columns. Unless it is square, it does not have a right inverse, i.e., there is no \( n \times m \) matrix \( X \) for which \( AX = I \). So instead we seek the matrix \( X \) for which the residual matrix \( R = AX - I \) is as small as possible, where we measure the size of \( R \) by the sum of the squares of its entries, \( \sum_{i,j} R_{ij}^2 \). We call this matrix the least-squares approximate right inverse of \( A \).

Show that the least-squares right inverse of \( A \) is given by \( X = A^\dagger \).

Hint. Let \( x_i \) denote the \( i \)th column of \( X \). Show that the objective can be expressed as

\[
\sum_{i,j} R_{ij}^2 = \|Ax_1 - e_1\|^2 + \cdots + \|Ax_m - e_m\|^2.
\]

12.5 Least squares equalizer design. You are given a channel impulse response, the \( n \)-vector \( c \). Your job is to find an equalizer impulse response, the \( n \)-vector \( h \), that minimizes \( \|h * c - e_1\|^2 \). You can assume that \( c_1 \neq 0 \). Remark. \( h \) is called an equalizer since it approximately inverts, or undoes, convolution by \( c \).

(a) Explain how to find \( h \).

(b) Find the least squares equalizer \( h \) for the channel \( c = (1.0, 0.7, -0.3, -0.1, 0.05) \). Plot \( c \), \( h \), and \( h * c \).

12.6 Network tomography. A network consists of \( n \) links, labeled 1, \ldots, \( n \). A path through the network is a subset of the links. (The order of the links on a path does not matter here.) Each link has a (positive) delay, which is the time it takes to traverse it. We let \( d \) denote the \( n \)-vector that gives the link delays. The total travel time of a path is the sum of the delays of the links on the path.

Our goal is to estimate the link delays (i.e., the vector \( d \)), from a large number of (noisy) measurements of the travel times along different paths. This data is given to you as an \( N \times n \) matrix \( P \), where

\[
P_{ij} = \begin{cases} 
1 & \text{link } j \text{ is on path } i \\
0 & \text{otherwise},
\end{cases}
\]

and an \( N \)-vector \( t \) whose entries are the (noisy) travel times along the \( N \) paths. You can assume that \( N > n \).

You will choose your estimate \( \hat{d} \) by minimizing the RMS deviation between the measured travel times (\( t \)) and the travel times predicted by the sum of the link delays.

Explain how to do this, and give a matrix expression for \( \hat{d} \). If your expression requires assumptions about the data \( P \) or \( t \), state them explicitly.

Remark. This problem arises in several practical contexts. The network could be a computer network, and a path gives the sequence of communication links data packets traverse. The network could be a transportation system, with the links representing road segments.

12.7 Least squares timing. A computer takes around one second to fit a regression model (using least-squares) with 20 parameters using \( 10^6 \) data points.

(a) About how long do you guess it will take the same computer to fit the same 20-parameter model using \( 10^7 \) data points (i.e., \( 10 \times \) more data points)?

(b) About how long do you guess it will take the same computer to fit a 200-parameter model using \( 10^6 \) data points (i.e., \( 10 \times \) more model parameters)?
13 Least squares data fitting

13.1 Saving TA time using midterm score prediction. The TAs very carefully graded all problems on all
students’ midterms. But suppose they had just graded the first half of the exam, i.e., problems 1–5,
and used a regression model to predict the total score on each exam.

The 95 × 10 matrix of midterm scores is available online, but for your convenience, we have created
the data file midterm_scores.jl for you. This file can be loaded with include("midterm_scores.jl");.

(a) Find the average and standard deviation of the total midterm scores.

(b) Here is a very simple way to predict the total score based on the 5 scores for first half of the
exam: Sum the 5 first half scores, and double the result. What RMS prediction error does
this simple method achieve?

(c) Find a regression model that predicts the total score based on the 5 scores for problems 1–5.
Give the coefficients and briefly interpret them. What is the RMS prediction error of your
model?

(Just for fun, evaluate the predictor on your own exam score. Would you rather have your
actual score or your predicted score?)

Remark. Your dedicated EE103 teaching staff would never do anything like this. Really.

13.2 Moore’s law. The figure and table below show the number of transistors N in 13 microprocessors,
and the year of their introduction.

<table>
<thead>
<tr>
<th>year</th>
<th>N (transistors)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1971</td>
<td>2,250</td>
</tr>
<tr>
<td>1972</td>
<td>2,500</td>
</tr>
<tr>
<td>1974</td>
<td>5,000</td>
</tr>
<tr>
<td>1978</td>
<td>29,000</td>
</tr>
<tr>
<td>1982</td>
<td>120,000</td>
</tr>
<tr>
<td>1985</td>
<td>275,000</td>
</tr>
<tr>
<td>1989</td>
<td>1,180,000</td>
</tr>
<tr>
<td>1993</td>
<td>3,100,000</td>
</tr>
<tr>
<td>1997</td>
<td>7,500,000</td>
</tr>
<tr>
<td>1999</td>
<td>24,000,000</td>
</tr>
<tr>
<td>2000</td>
<td>42,000,000</td>
</tr>
<tr>
<td>2002</td>
<td>220,000,000</td>
</tr>
<tr>
<td>2003</td>
<td>410,000,000</td>
</tr>
</tbody>
</table>

These numbers are available in the Julia file moore_data.jl In this data file, t is the first column
of the table (introduction year) and N is the second column (number of transistors).

The plot gives the number of transistors on a logarithmic scale. Find the least-squares straight-line
fit of

\[(\log_{10} N, t - 1970),\]
where $t$ is the year and $N$ is the number of transistors, from the given data. This gives an approximation of the form
\[
\log_{10} N \approx \theta_1 + \theta_2(t - 1970).
\]

(a) Find the coefficients $\theta_1$ and $\theta_2$ that minimize the RMS error on the data, and give the RMS error on the data. Plot the model you find along with the data points.

(b) Use your model to predict the number of transistors in a microprocessor introduced in 2015. Compare the prediction to the IBM Z13 microprocessor, released in 2015, which has around $4 \times 10^9$ transistors.

(c) Compare your result with Moore’s law, which states that the number of transistors per integrated circuit roughly doubles every one and a half to two years.

Hints. In Julia, the function $\log_{10}$ is `log10`.

13.3 Orthogonality principle.

(a) Suppose that $A$ has independent columns, and $\hat{x}$ is the least-squares approximate solution of $Ax = b$. Let $\hat{r} = A\hat{x} - b$ be the associated residual. Show that for any $x$, $(Ax) \perp \hat{r}$. This is sometimes called the orthogonality principle: every linear combination of the columns of $A$ is orthogonal to the least-squares residual.

(b) Now consider a data fitting problem, with first basis function $\phi_1(x) = 1$, and data set $(x_1, y_1), \ldots, (x_N, y_N)$. Assume the matrix $A$ in the associated least-squares problem has independent columns, and let $\hat{\theta}$ denote the parameter values that minimize the mean square prediction error over the data set. Let the $N$-vector $\hat{r}$ denote the prediction errors using the optimal model parameter $\hat{\theta}$. Show that $\text{avg}(\hat{r}) = 0$. In other words: With the least-squares fit, the mean of the prediction errors over the data set is zero.

Hint. Consider part (a), with $x = e_1$.

13.4 Auto-regressive time series prediction. Suppose that $x$ is an $N$-vector representing time series data. The (one step ahead) prediction problem is to guess $x_{t+1}$, based on $x_1, \ldots, x_t$. We will base our prediction $\hat{x}_{t+1}$ of $x_{t+1}$ on the previous $M$ values, $x_t, x_{t-1}, \ldots, x_{t-M+1}$. (The number $M$ is called the memory length of our predictor.) When the prediction is a linear function,
\[
\hat{x}_{t+1} = \beta_1 x_t + \beta_2 x_{t-1} + \cdots + \beta_M x_{t-M+1},
\]
it is called an auto-regressive predictor. (It is possible to add an offset to $\hat{x}_{t+1}$, but we will leave it out for simplicity.) Of course we can only use our auto-regressive predictor for $M \leq t \leq N - 1$.

Some very simple and natural predictors have this form. One example is the predictor $\hat{x}_{t+1} = x_t$, which guesses that the next value is the same as the current one. Another one is $\hat{x}_{t+1} = x_t + (x_t - x_{t-1})$, which guesses what $x_{t+1}$ is by extrapolating a line that passes through $x_t$ and $x_{t-1}$.

We judge a predictor (i.e., the choice of coefficients $\beta_i$) by the mean-square prediction error
\[
J = \frac{1}{N-M} \sum_{t=M}^{N-1} (\hat{x}_{t+1} - x_{t+1})^2.
\]

A sophisticated choice of the coefficients $\beta_i$ is the one that minimizes $J$. We will call this the least-squares auto-regressive predictor.
(a) Find the matrix $A$ and the vector $b$ for which $J = \|A\beta - b\|^2/(N - M)$. This allows you to find the coefficients that minimize $J$, i.e., the auto-regressive predictor that minimizes the mean-square prediction error on the given time series. Be sure to give the dimensions of $A$ and $b$.

(b) For $M = 2, \ldots, 12$, find the coefficients that minimize the mean-square prediction error on the time series $x_{\text{train}}$ given in `time_series_data.jl`. The same file has a second time series $x_{\text{test}}$ that you can use to test or validate your predictor on. Give the values of the mean-square error on the train and test series for each value of $M$. What is a good choice of $M$? Also find $J$ for the two simple predictors described above.

*Hint.* Be sure to use the `toeplitz` function contained in `time_series_data.jl`. It’ll make your life a lot easier. Documentation for the function is also contained in `time_series_data.jl`.

13.5 **Conclusions from 5-fold cross validation.** You have developed a regression model for predicting a scalar outcome $y$ from a feature vector $x$ of dimension 20, using a collection of $N = 600$ data points. The mean of the outcome variable $y$ across the given data is 1.85, and its standard deviation is 0.32. After running 5-fold cross validation we get the following RMS test errors (based on forming a model based on the data excluding fold $i$, and testing it on fold $i$).

<table>
<thead>
<tr>
<th>Fold excluded</th>
<th>RMS test error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.13</td>
</tr>
<tr>
<td>2</td>
<td>0.11</td>
</tr>
<tr>
<td>3</td>
<td>0.09</td>
</tr>
<tr>
<td>4</td>
<td>0.11</td>
</tr>
<tr>
<td>5</td>
<td>0.14</td>
</tr>
</tbody>
</table>

(a) How would you expect your model to do on new, unseen (but similar) data? Respond briefly. Feel free to say that you are not comfortable making any prediction at all, and why, if this is the appropriate response. You should also feel free to give a guarantee, if you think it is warranted based on the results above. In any case, briefly justify your response.

(b) A co-worker observes that the regression model parameters found in the 5 different folds are quite close, but not the same. He says that for the production system, you should use the regression model parameters found when you excluded fold 3 from the original data, since it achieved the best RMS test error. Comment briefly.

(c) Another co-worker says that you might get a much better model (in terms of predicting new, unseen data) by adding a new 21st feature, which is $x_{21} = \max\{x_1, \ldots, x_{20}\}$. Give an appropriate, and brief, response to her.

13.6 **Augmenting features.** You are fitting a regression model $\hat{y} = x^T\beta + v$ to data, using least squares without regularization, computing the model coefficients $\beta$ and $v$ via the QR factorization. A friend suggests adding a new feature, which is the average of the original features. (That is, he suggests using the new feature vector $\tilde{x} = (x, \text{avg}(x)).$) He explains that by adding this new feature, you might end up with a better model. (Of course, you would test the new model using validation.)

Choose one of the following, and then explain why below.

- This is a good idea, and it’s worth a try.
This is a bad idea, and will lead to trouble.

13.7 *Interpreting model fitting results.* Five different models are fit using the same training data set, and tested on the same (separate) test set (which has the same size as the training set). The RMS prediction error on the training and test set are reported below.

Write no more than two sentences in English interpreting or commenting on the results of each model. You might mention whether the model is good or bad, whether it is likely to generalize to unseen data, or whether it is over-fit. You are also welcome to say that you don’t believe the results, or think the reported numbers are fishy.

<table>
<thead>
<tr>
<th>Model</th>
<th>Train RMS</th>
<th>Test RMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1.355</td>
<td>1.423</td>
</tr>
<tr>
<td>B</td>
<td>9.760</td>
<td>9.165</td>
</tr>
<tr>
<td>C</td>
<td>5.033</td>
<td>0.889</td>
</tr>
<tr>
<td>D</td>
<td>0.211</td>
<td>5.072</td>
</tr>
<tr>
<td>E</td>
<td>0.633</td>
<td>0.633</td>
</tr>
</tbody>
</table>
A university registrar keeps track of class attendance, measured as a percentage (so 100 means full attendance, and 30 means 30% of the students attend). For each class lecture, she records the attendance $y$, and several features:

- $x_1$ is the day of week, with Monday coded as 1 and Friday coded as 5.
- $x_2$ is the week of the quarter, coded as 1 for the first week, and 10 for the last week.
- $x_3$ is the hour of the lecture, with 8AM coded as 8, and 4PM coded as 16.
- $x_4 = \max\{T - 80, 0\}$, where $T$ is the outside temperature (so $x_4$ is the number of degrees above 80°F).
- $x_5 = \max\{50 - T, 0\}$, where $T$ is the outside temperature (so $x_5$ is the number of degrees below 50°F).

(These features were suggested by a professor who is an expert in the theory of class attendance.)

An EE103 alumna carefully fits the data with the following regression model,

$$\hat{y} = -1.4x_1 - 0.3x_2 + 1.1x_3 - 0.6x_4 - 0.5x_5 + 68.2,$$

and validates it properly.

Give a short story/explanation, in English, of this model.
14 Least squares classification

14.1 Least-squares classification. The file `lsq_classifier.jl` contains feature \( n \)-vectors \( x_1, \ldots, x_N \), and the associated binary labels, \( y_1, \ldots, y_N \), each of which is either +1 or −1. The feature vectors are stored as an \( n \times N \) matrix \( X \) with columns \( x_1, \ldots, x_N \), and the labels are stored as an \( N \)-vector \( y \).

We want to choose the weight \( n \)-vector \( w \) and the offset (scalar) \( v \) so that \( \hat{f}(x) = \text{sign}(w^T x + v) \) is a good predictor of the label \( y \), as judged by the classification error rate, which is the fraction of examples for which \( y_i \neq \hat{y}_i = \hat{f}(x_i) \). We will evaluate the error rate on the (training) data \( X, y \) and also (to check if the model generalizes) a test set \( X_{\text{test}}, y_{\text{test}} \), also given in `lsq_classifier.jl`.

You may use `LinearLeastSquares` for all parts of the problem. Also, include your Julia code in your solution.

(a) Least-squares classifier. Find \( w, v \) that minimize \( \sum_{i=1}^{N} (w^T x_i + v - y_i)^2 \). Copy your code for finding \( w, v \) into the appropriate section of `lsq_classifier.jl`. Report the classification error on the training and test sets.

(b) Regularized least-squares classifier. Now we add regularization to improve the generalization ability of the classifier. Find \( w, v \) that minimize

\[
\sum_{i=1}^{N} (w^T x_i + v - y_i)^2 + \lambda \|w\|^2,
\]

where \( \lambda > 0 \) is the regularization parameter, for a range of values of \( \lambda \). We suggest the range \( 10^{-1} \) to \( 10^4 \), say, 100 values logarithmically spaced. Again, copy your code for finding \( w, v \) into the appropriate section of `lsq_classifier.jl`.

Show the plot of classification error versus \( \lambda \), for both the training and test sets, as plotted by `lsq_classifier.jl`. Suggest a reasonable choice of \( \lambda \).

14.2 Trading off false positive and false negative rates. In this problem you use the Boolean least squares classifier with skewed decision point, described on pages 183-184 of the textbook, to alter the false positive and false negative rates. The classifier is \( \hat{f}(x) = \text{sign}(\tilde{f}(x) - \alpha) \), where \( \tilde{f} \) is the (real-valued) least squares predictor of the labels encoded as \( \pm 1 \), and \( \alpha \) is a constant that skews the decision point. We will use basic regression, i.e., \( \tilde{f}(x) = x^T \beta + v \).

(a) Basic classifier. The file `classifier_with_tradeoff_data.jl` contains train and test sets \( X_{\text{train}}, X_{\text{test}} \), and the corresponding labels \( y_{\text{train}} \) and \( y_{\text{test}} \). Find the model parameters \( \beta \) and \( v \) using the training data set, and give the confusion matrix for the associated classifier (with \( \alpha = 0 \)) for both the training and test sets. Give the error rate for the train and test sets.

(b) ROC curve from skewed decision point classifier. The ROC curve plots the performance of different classifiers with the vertical axis showing the true positive rate, and the horizontal axis showing the false positive rate. (See textbook, page 183.) Give ROC plots for the training and test data sets, for the skewed decision point classifier, for 20 values of \( \alpha \) ranging from −0.3 to +0.3. (You will use the values of \( \beta \) and \( v \) found in part (a)).
14.3 Equalizer design from training message. We consider a communication system, with message to be sent given by an \( N \)-vector \( s \), whose entries are 0 or 1, and received signal \( y \), where \( y = c * s \), where \( c \) is an \( n \)-vector, the channel impulse response. The receiver applies equalization to the received signal, which means that it computes \( \tilde{y} = h * y = h * c * s \), where \( h \) is an \( n \)-vector, the equalizer impulse response. The receiver then estimates the original message using \( \hat{s} = \text{round}(\tilde{y}_{1:N}) \). This works well if \( h * c \approx e_1 \).

In some cases, the channel impulse response \( c \) can be measured. Once we know \( c \), we can design or choose \( h \), for example, by least squares. In many other cases (especially when the channel is wireless), it’s not practical to measure \( c \), and in addition, \( c \) changes over time. In those cases a different method for choosing \( h \) is used, which we explore in this problem.

The method directly designs \( h \) without estimating or measuring \( c \). The sender first sends a message that is known to the receiver, called the training message, \( s^{tr} \). (From the point of view of communications, this is wasted transmission, and is called overhead.) The receiver receives the signal \( y^{tr} = c * s^{tr} \) from the training message, and then chooses \( h \) to minimize \( \| (h * y^{tr})_{1:N} - s^{tr} \|_2 \).

In practice, this equalizer is used until the bit error rate increases (which means the channel has changed), at which point another training message is sent.

Finally we get to the problem. The file \texttt{eq\_design\_data.jl} contains the training message \( s^{tr} \), the value of \( n \), and the signal received from the training message, \( y^{tr} \). Your first task is to design an equalizer \( h \) using this data, and plot it.

The file also includes the received signal \( y \) from a message \( s \) that is sent. First, round \( y \) to get an estimate of the message, and print it as a string. Then estimate the message \( s \) using your equalizer, and print it as text. You’ll know when your equalizer is working.

Hints.

- In Julia you can round to 0 or 1 using
  \[
  s\_round = \text{int}(y .>= 0.5);
  \]

- You can turn a Boolean vector, a vector with entries only 0 or 1, into a string using the function \texttt{binary2string}. (While not needed for this problem, the function \texttt{string2binary} converts a string to a Boolean vector.)
15 Multi-objective least squares

15.1 Some properties of bi-objective least squares. Consider the bi-objective least-squares problem with objectives \( J_i(x) = \| A_i x + b_i \|^2, i = 1, 2 \). Let \( \hat{x}(\lambda) \) denote the minimizer of \( J_1(x) + \lambda J_2(x) \), for \( \lambda > 0 \). (We assume the columns of the stacked matrix are linearly independent.) We define \( J_i^*(\lambda) = J_i(\hat{x}(\lambda)) \), for \( i = 1, 2 \), the value of the two objectives as a function of the weight parameter. The optimal trade-off curve is the set of points \((J_1^*(\lambda), J_2^*(\lambda))\), as \( \lambda \) varies over all positive numbers.

(a) Effect of weight on objectives in a bi-objective problem. Show the following: for \( \lambda < \mu \), we have \( J_1^*(\lambda) \leq J_1^*(\mu) \) and \( J_2^*(\lambda) \geq J_2^*(\mu) \), which means if you increase the weight (on the second objective), the second objective goes down (or stays the same), and the first objective goes up (or stays the same). This means that the trade-off curve always slopes downward.

\( \text{Hint.} \) Resist the urge to write out any equations or formulas. Use the fact that \( \hat{x}(\lambda) \) minimizes \( J_1(x) + \lambda J_2(x) \), and similarly for \( \hat{x}(\mu) \), to deduce the inequalities

\[
J_1^*(\mu) + \lambda J_2^*(\mu) \geq J_1^*(\lambda) + \lambda J_2^*(\lambda), \quad J_1^*(\lambda) + \mu J_2^*(\lambda) \geq J_1^*(\mu) + \mu J_2^*(\mu).
\]

Use these to show \( J_1^*(\lambda) \leq J_1^*(\mu) \) and \( J_2^*(\lambda) \geq J_2^*(\mu) \).

(b) Slope of the trade-off curve. The slope of the trade-off curve at the point \((J_1^*(\lambda), J_2^*(\lambda))\) is given by

\[
S = \lim_{\mu \to \lambda} \frac{J_2^*(\mu) - J_2^*(\lambda)}{J_1^*(\mu) - J_1^*(\lambda)}.
\]

(This limit is the same if \( \mu \) approaches \( \lambda \) from below or from above.) Show that \( S = -1/\lambda \).

This gives another interpretation of the parameter \( \lambda \): \( (J_1^*(\lambda), J_2^*(\lambda)) \) is the point on the trade-off curve that has slope \(-1/\lambda\).

\( \text{Hint.} \) First assume that \( \mu \) approaches \( \lambda \) from above (meaning, \( \mu > \lambda \)), and use the inequalities in the hint for problem 1 to show that \( S \geq -1/\lambda \). Then assume that \( \mu \) approaches \( \lambda \) from below, and show that \( S \leq -1/\lambda \).

15.2 Trading off tracking error and input size in control. A system that we want to control has input (time series) \( u \) and output (time series) \( y \), related by convolution: \( y = h * u \), where

\[
h = (0.3, 0.5, 0.6, 0.4, 0.3, 0.2, 0.1).
\]

(See §6.5.4 in the textbook.) We are given \( y^{\text{des}} \), the (time series of) desired or target output values, and we will choose the input \( u \) to minimize \( \| y - y^{\text{des}} \|^2 + \lambda \| u \|^2 \), where \( \lambda > 0 \) is a parameter we use to trade off tracking error (\( i.e., \| y - y^{\text{des}} \|^2 \)) and input size (\( i.e., \| u \|^2 \)). We will take the desired output to be the 100-vector

\[
y^{\text{des}}_t = \begin{cases} 10 & 10 \leq t < 40 \\ -5 & 40 \leq t < 80 \\ 0 & \text{otherwise.} \end{cases}
\]

Plot a trade off curve of the tracking error (\( \text{rms}(y - y^{\text{des}}) \)) versus the regularization error (\( \text{rms}(u) \)). Pick 3 values of \( \lambda \) that correspond to too little regularization, too much regularization, and a reasonable amount of regularization. (Reasonable might correspond to RMS tracking error around 0.3) Plot the input \( u \) found for each choice of \( \lambda \) on the same figure. Also plot the output \( y \) found for each \( \lambda \) on the same figure, along with \( y^{\text{des}} \).

\( \text{Hint.} \) Make sure to use the \texttt{toeplitz} function described in the hint for problem 2.
15.3 Least-squares with smoothness regularization. Consider the weighted sum least-squares objective
\[ \|Ax - b\|^2 + \lambda \|Dx\|^2, \]
where the \(n\)-vector \(x\) is the variable, \(A\) is an \(m \times n\) matrix, \(D\) is the \((n-1) \times n\) difference matrix, with \(i\)th row \((e_{i+1} - e_i)^T\), and \(\lambda > 0\). Although it does not matter in this problem, this objective is what we would minimize if we want an \(x\) that satisfies \(Ax \approx b\), and has entries that are smoothly varying. We can express this objective as a standard least-squares objective with a stacked matrix of size \((m+n-1) \times n\).

What are the conditions under which the stacked matrix has independent columns? Express your answer in the simplest way possible, in terms of the rows or columns of \(A\), and the dimensions \(m\) and \(n\). (You may not need to mention all of these.) We will deduct points from solutions that are correct, but more complicated than they need to be. You must justify your answer.

15.4 Least squares classification with regularization. The file \(\text{lsq\_classifier\_data.jl}\) contains feature \(n\)-vectors \(x_1, \ldots, x_N\), and the associated binary labels, \(y_1, \ldots, y_N\), each of which is either +1 or -1. The feature vectors are stored as an \(n \times N\) matrix \(X\) with columns \(x_1, \ldots, x_N\), and the labels are stored as an \(N\)-vector \(y\). We will evaluate the error rate on the (training) data \(X, y\) and (to check if the model generalizes) a test set \(X_{\text{test}}, y_{\text{test}}\), also given in \(\text{lsq\_classifier\_data.jl}\). You may use \(\text{LinearLeastSquares}\) for all parts of the problem. Include your Julia code in your solution.

(a) Least squares classifier. Find \(\beta, v\) that minimize \(\sum_{i=1}^{N} (x_i^T \beta + v - y_i)^2\) on the training set. Our predictions are then \(\hat{f}(x) = \text{sign}(x^T \beta + v)\). Report the classification error on the training and test sets, the fraction of examples where \(\hat{f}(x_i) \neq y_i\).

(b) Regularized least squares classifier. Now we add regularization to improve the generalization ability of the classifier. Find \(\beta, v\) that minimize
\[ \sum_{i=1}^{N} (x_i^T \beta + v - y_i)^2 + \lambda \|\beta\|^2, \]
where \(\lambda > 0\) is the regularization parameter, for a range of values of \(\lambda\). We suggest the range \(10^{-1}\) to \(10^4\), say, 100 values logarithmically spaced. The function \(\text{logspace}\) may be useful. Use it to plot the training and test set errors against \(\log_{10}(\lambda)\). Suggest a reasonable choice of \(\lambda\).

15.5 Estimating the elasticity matrix. In this problem you create a standard model of how demand varies with the prices of a set of products, based on some observed data. There are \(n\) different products, with (positive) prices given by the \(n\)-vector \(p\). The prices are held constant over some period, say, a day. The (positive) demands for the products over the day is given by the \(n\)-vector \(d\). The demand in any particular day varies, but it is thought to be (approximately) a function of the prices. The units of the prices and demands don’t really matter in this problem. Demand could be measured in 10000 units, and prices in $100.

The nominal prices are given by the \(n\)-vector \(p_{\text{nom}}\). You can think of these as the prices that have been charged in the past for the products. The nominal demand is the \(n\)-vector \(d_{\text{nom}}\). This is the average value of the demand, when the prices are set to \(p_{\text{nom}}\). (The actual daily demand fluctuates around the value \(d_{\text{nom}}\).) You know both \(p_{\text{nom}}\) and \(d_{\text{nom}}\). We will describe the prices by
their (fractional) variations from the nominal values, and the same for demands. We define $\delta^p$ and $\delta^d$ as the (vectors of) relative price change and demand change:

$$
\delta^p_i = \frac{p_i - p_i^{\text{nom}}}{p_i^{\text{nom}}}, \quad \delta^d_i = \frac{d_i - d_i^{\text{nom}}}{d_i^{\text{nom}}}, \quad i = 1, \ldots, n.
$$

So $\delta^p_3 = +0.05$ means that the price for product 3 has been increased by 5% over its nominal value, and $\delta^d_5 = -0.04$ means that the demand for product 5 in some day is 4% below its nominal value.

Your task is to build a model of the demand as a function of the price, of the form

$$
\delta^d \approx E \delta^p,
$$

where $E$ is the $n \times n$ elasticity matrix.

You don’t know $E$, but you do have the results of some experiments in which the prices were changed a bit from their nominal values for one day, and the day’s demands were recorded. This data has the form

$$(p_1, d_1), \ldots, (p_N, d_N),$$

where $p_i$ is the price for day $i$, and $d_i$ is the observed demand.

Explain how you would estimate $E$, given this price-demand data. Be sure to explain how you will test for, and (if needed) avoid over-fit.

*Hint.* You might find it easier to separately fit the models $\delta^d_i \approx \tilde{e}_i^T \delta^p$, where $\tilde{e}_i$ is the $i$th row of $E$. (We use the tilde above $e_i$ to avoid conflict with the notation for unit vectors.)

Carry out your method using the price and demand data in the matrices Prices and Demands, found in price_elasticity.jl. Give your estimate $\hat{E}$, and guess (roughly) how accurate your model $\delta^d = \hat{E} \delta^p$ might be (in terms of RMS prediction error) on unseen data.

Here are some facts about elasticity matrices that might help you check that your estimates make sense (but you don’t need to incorporate this information into your estimation method). The diagonal entries of $E$ are always negative, and typically on the order of one. (This means that when you raise the price of one product only, demand for it goes down by a similar fractional amount as the price increase.) The off-diagonal entries can have either sign, and are typically (but not always) smaller than one in magnitude.

15.6 *Greedy regulation policy.* Consider a linear dynamical system given by $x_{t+1} = Ax_t + Bu_t$, where the $n$-vector $x_t$ is the state at time $t$, and the $m$-vector $u_t$ is the input at time $t$. The goal in regulation is to choose the input so as to make the state small. (In applications, the state $x_t = 0$ corresponds to the desired operating point, so small $x_t$ means the state is close to the desired operating point.)

One way to achieve this goal is to choose $u_t$ so as to minimize

$$
\|x_{t+1}\|^2 + \rho \|u_t\|^2,
$$

where $\rho$ is a (given) positive parameter that trades off using a small input versus making the (next) state small.

Show that choosing $u_t$ this way leads to a state feedback policy $u_t = Kx_t$, where $K$ is an $m \times n$ matrix. Give a formula for $K$ (in terms of $A$, $B$, and $\rho$). If an inverse appears in your formula, state the conditions under which the inverse exists.

*Remark.* This policy is called greedy since it does not take into account the effect of the input $u_t$ on future states, beyond $x_{t+1}$. It can work very poorly in practice.
16 Constrained least squares

16.1 Smallest right inverse. Suppose the \( m \times n \) matrix \( A \) is wide, with independent rows. Its pseudo-inverse \( A^\dagger \) is then a right inverse of \( A \). In fact, there are many right inverses of \( A \), and it turns out that \( A^\dagger \) is the smallest one among them, as measured by the sum of the squares of the entries. In other words, if \( X \) satisfies \( AX = I \), then

\[
\sum_{i,j} X_{ij}^2 \geq \sum_{i,j} (A^\dagger)_{ij}^2.
\]

You will show this in this problem.

(a) Suppose \( AX = I \), and let \( x_1, \ldots, x_m \) denote the columns of \( X \). Let \( a_j \) denote the \( j \)th column of \( A^\dagger \). Explain why \( \|x_j\|_2 \geq \|a_j\|_2 \).

\textbf{Hint.} Show that \( z = a_j \) is the vector of smallest norm that satisfies \( Az = e_j \), for \( j = 1, \ldots, m \).

(b) Use the inequalities from part (a) to establish the inequality above.

16.2 Closest solution to a given point. Suppose the wide matrix \( A \) has independent rows. Find an expression for the point \( x \) that is closest to a given vector \( \tilde{x} \) (i.e., minimizes \( \|x - \tilde{x}\|_2 \)) among all vectors that satisfy \( Ax = b \).

\textbf{Remark.} This problem comes up when \( x \) is some set of inputs to be found, \( Ax = b \) represents some set of requirements, and \( \tilde{x} \) is some nominal value of the inputs. For example, when the inputs represent actions that are re-calculated each day (say, because \( b \) changes every day), \( \tilde{x} \) might be yesterday’s action, and the today’s action \( x \) found as above gives the least change from yesterday’s action, subject to meeting today’s requirements.

16.3 Computing least-norm solutions. Generate a random \( 10 \times 100 \) matrix \( A \) and a 10-vector \( b \). Use Julia to compute the least norm solution for \( Ax = b \) using the methods listed below, and verify that the solutions found are the same (or more accurately, very close to each other). Be sure to submit your code, including the code that checks if the solutions are close to each other.

\begin{itemize}
  \item Using the formula \( \hat{x} = A^T (A A^T)^{-1} b \).
  \item Using the pseudo inverse: \( \hat{x} = A^\dagger b \).
  \item Using the Julia backslash operator.
  \item Using the Julia package \texttt{LinearLeastSquares}.
\end{itemize}

16.4 Julia timing test for least-norm. Determine how long it takes for your computer to solve a least-norm problem with \( m = 600 \) equations and \( n = 4000 \) variables. (You can use the backslash operator.) What approximate flop rate does your result suggest?

\textbf{Remark.} Julia compiles just in time, so you should run the code a few times to get the correct time.

16.5 Nearest vector with a given average. Let \( a \) be an \( n \)-vector, and \( \beta \) a scalar. How would you find the \( n \)-vector \( x \) that is closest to \( a \) among all \( n \)-vectors that have average value \( \beta \)?

Your answer can involve any matrix or vector operations, and should be as simple as possible. Express the solution in English as well. You must justify your answer.
17 Constrained least squares applications

17.1 Portfolio optimization. In this problem you will optimize a set of holdings to minimize risk for various average returns. The file portfolio_optimization.jl contains \texttt{train_returns} and \texttt{test_returns} which are return matrices for 20 assets over 2000 days and 500 days, respectively. Using \texttt{train_returns} find asset allocation weights for 4 portfolios that minimize risk for annualized returns of 5\%, 10\%, 20\%, and 40\%. Report the annualized return and risk on the training and test sets for all 4 portfolios, and comment briefly. Plot the cumulative value for each portfolio over time, starting from the conventional initial investment of $10000, for both the train and test sets of returns. You can use the \texttt{cumprod} function to compute the product $V_1(1+r_{T1}^1 w)(1+r_{T2}^2 w)\cdots(1+r_{TT}^T w)$. For each of the 4 portfolios, report the leverage, defined as $\sum|w_i|$. (Several other definitions of leverage are used.) This number is always at least one, and it is exactly one only if the portfolio has no short positions.

17.2 Rendezvous. The dynamics of two vehicles, at sampling times $t = 1, 2, \ldots$, are given by

$$x_{t+1} = Ax_t + Bu_t, \quad z_{t+1} = Az_t + Bv_t$$

where the $n$-vectors $x_t$ and $z_t$ are the states of vehicles 1 and 2, and the $m$-vectors $u_t$ and $v_t$ are the inputs of vehicles 1 and 2. The $n \times n$ matrix $A$ and the $n \times m$ matrix $B$ are known.

The position of vehicle 1 at time $t$ is given by $Cx_t$, where $C$ is a known $2 \times n$ matrix. Similarly, the position of vehicle 2 at time $t$ is given by $Cz_t$.

The initial states of the two vehicles are fixed and given:

$$x_1 = x_{start}, \quad z_1 = z_{start}.$$ 

We are interested in finding a sequence of inputs for the two vehicles over the time interval $t = 1, \ldots, T - 1$ so that they rendezvous at time $t = T$, i.e., $x_T = z_T$. You can select the inputs to the two vehicles,

$$u_1, u_2, \ldots, u_{T-1}, \quad v_1, v_2, \ldots, v_{T-1}.$$ 

Among choices of the sequences $u_1, \ldots, u_{T-1}$ and $v_1, \ldots, v_{T-1}$ that satisfy the rendezvous condition, we want the one that minimizes the weighted sum of squares of the two vehicle inputs,

$$J = \sum_{t=1}^{T-1} \|u_t\|^2 + \lambda \sum_{t=1}^{T-1} \|v_t\|^2,$$

where $\lambda > 0$ is a parameter that trades off the two objectives.

(a) Explain how to find the sequences $u_1, \ldots, u_{T-1}$ and $v_1, \ldots, v_{T-1}$ that minimize $J$ while satisfying the rendezvous condition by solving a constrained least-squares problem.

(b) The problem data $A$, $B$, $C$, $x_{\text{start}}$, and $z_{\text{start}}$ are defined in rendezvous.jl. Use \texttt{LinearLeastSquares} to find $u_1, \ldots, u_{T-1}$ and $v_1, \ldots, v_{T-1}$ for $\lambda = 0.1$, $\lambda = 1$, and $\lambda = 10$. Plot the vehicle trajectories (i.e., their positions) for each $\lambda$ using the plotting code in rendezvous.jl.

(c) Give a simple expression for $x_T$ in the limit where $\lambda \to \infty$ and for $z_T$ in the limit where $\lambda \to 0$. Assume that for any $w \in \mathbb{R}^n$ there exist sequences $u_1, \ldots, u_{T-1}$ and $v_1, \ldots, v_{T-1}$ such that the rendezvous constraints are satisfied with $w = z_T = x_T$. 

51
17.3 A linear regulator for a linear dynamical system. We consider a linear dynamical system with dynamics \( x_{t+1} = Ax_t + Bu_t \), where the \( n \)-vector \( x_t \) is the state at time \( t \) and the \( m \)-vector \( u_t \) is the input at time \( t \). We assume that \( x = 0 \) represents the desired operating point; the goal is to find an input sequence \( u_1, \ldots, u_{T-1} \) that results in \( x_T = 0 \), given the initial state \( x_1 \). Choosing an input sequence that takes the state to the desired operating point at time \( T \) is called regulation.

(a) Find an explicit formula for the sequence of inputs that yields regulation, and minimizes \( \|u_1\|^2 + \cdots + \|u_{T-1}\|^2 \), in terms of \( A, B, T, \) and \( x_1 \). This control is called the minimum energy regulator.

Hint. Your formula may involve the controllability matrix
\[
C = \begin{bmatrix} B & AB & \cdots & A^{T-2}B \end{bmatrix},
\]
and the vector \( u = (u_{T-1}, u_{T-2}, \ldots, u_1) \) (which is the input sequence in reverse order). You do not need to expand expressions involving \( C \), such as \( CC^T \) or \( C^\dagger \), in terms of \( A \) and \( B \); you are welcome to simply give your answers using \( C \). You may assume that \( C \) is wide and has independent rows.

(b) Show that \( u_t \) (the \( t \)th input in the sequence found in part (a)) can be expressed as \( u_t = K_t x_1 \), where \( K_t \) is an \( m \times n \) matrix. Show how to find \( K_t \) from \( A \) and \( B \). (But feel free to use the matrix \( C \) in your answer.)

Hint. Your expression for \( K_t \) can include submatrices of \( C \) or \( C^\dagger \).

(c) A constant linear regulator. A very common regulator strategy is to simply use \( u_t = K_1 x_t \) for all \( t, \ t = 1, 2, 3, \ldots \). This is called a (constant) linear regulator, and \( K_1 \) is called the state feedback gain (since it maps the current state into the control input). Using this control strategy can be interpreted as always carrying out the first step of minimum energy control, as if we were going to steer the state to zero \( T \) steps in the future. This choice of input does not yield regulation in \( T \) steps, but it typically achieves asymptotic regulation, which means that \( x_t \to 0 \) as \( t \to \infty \).

Find the state feedback gain \( K_1 \) for the specific system with
\[
A = \begin{bmatrix} 1.003 & 0 & -0.008 \\ 0.005 & 0.997 & 0 \\ 0 & 0.005 & 1.002 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ 4 & 5 \\ 6 & 2 \end{bmatrix},
\]
using \( T = 10 \). You may find the code in \texttt{regulation.jl} useful.

(d) Simulate the system given in part (c) from several choices of initial state \( x_1 \), under two conditions: open-loop, which means \( u_t = 0 \), and closed-loop, which means \( u_t = K_1 x_t \), where \( K_1 \) is the state feedback gain found in part (c). Use \texttt{regulation.jl} to plot \( x \).

17.4 Computing least-norm solutions. Generate a random \( 10 \times 100 \) matrix \( A \) and a \( 10 \)-vector \( b \). Use Julia to compute the least norm solution for \( Ax = b \) using the methods listed below, and verify that the solutions found are the same (or more accurately, very close to each other). Be sure to submit your code, including the code that checks if the solutions are close to each other.

- Using the formula \( \hat{x} = A^T (AA^T)^{-1} b \).
- Using the pseudo inverse: \( \hat{x} = A^\dagger b \).
• Using the Julia backslash operator.
• Using the Julia package `LinearLeastSquares`.

17.5 *Julia timing test for least-norm.* Determine how long it takes for your computer to solve a least-norm problem with $m = 600$ equations and $n = 4000$ variables. (You can use the backslash operator.) What approximate flop rate does your result suggest?

*Remark.* Julia compiles just in time, so you should run the code a few times to get the correct time.

17.6 *Nearest vector with a given average.* Let $a$ be an $n$-vector, and $\beta$ a scalar. How would you find the $n$-vector $x$ that is closest to $a$ among all $n$-vectors that have average value $\beta$?

Your answer can involve any matrix or vector operations, and should be as simple as possible. Express the solution in English as well. You must justify your answer.
18 Nonlinear least squares

18.1
19 Constrained nonlinear least squares

19.1
20 Julia

20.1 Creating vectors in Julia. In each of the parts below, use Julia to create the described vector $a$. In each case, check that $a^T x$ gives the correct result, for a random vector $x$.

(a) $a^T x$ extracts (is equal to) the 5th entry of the 10-vector $x$.
(b) $a^T x$ is the weighted average of a 3-vector $x$, assigning weights 0.3 to the first component, 0.4 to the second, and 0.3 to the third.
(c) $a^T x$ (with $x$ a 22-vector) is the sum of $x_i$ for $i$ a multiple of 4, minus the sum of $x_i$ for $i$ is a multiple of 7.
(d) $a^T x$ (with $x$ an 11-vector) is the average of the middle five entries of $x$, i.e., entries 4 to 8.

20.2 Checking the MMA functions. Check if the following MMA functions work by checking them on values. rms, avg, std.

20.3 Nearest neighbor. Using Julia, find the nearest neighbor of $a = (1, 3, 4)$ among the vectors

$$x_1 = (4, 3, 5), \quad x_2 = (0.4, 10, 50), \quad x_3 = (1, 4, 10), \quad x_4 = (3, 4, 5), \quad x_5 = (5, 10, 50).$$

Report the minimum distance of $a$ to $x_1, \ldots, x_5$.

20.4 Angle and distance. Generate two random 10-vectors $a$ and $b$ and calculate the angle and distance between $a$ and $b$.

20.5 Triangle inequality. Check the triangle inequality by generating three random 10-vectors $a$, $b$, and $c$, and comparing $\text{dist}(a, c)$ with $\text{dist}(a, b) + \text{dist}(b, c)$.

20.6 Chebyshev inequality. Generate a random 200-vector $x$. Verify that no more than 8 of the entries of $x$ satisfy $|x_i| \geq 5 \text{rms}(x)$.

20.7 Verifying Cauchy-Schwarz in Julia. Generate two 20-vectors $x$ and $y$. Compute $|x^T y|$ and $\|x\||\|y\|$ and verify that the Cauchy-Schwarz inequality holds.

20.8 Linear independence. Are the following sets of vectors linearly dependent or linearly independent? Give a simple argument showing dependence or independence, and also check your answers using the Julia function `gram_schmidt()` in `gramschmidt.jl`.

The `gram_schmidt()` function accepts a list of vectors ($a_i$ in our notation in the slides and book), and returns a list of vectors ($q_i$ in our notation). So, $\text{length}(\text{gram_schmidt}(\text{vectors})) = \text{length}(\text{vectors})$ is TRUE when the vectors are linearly independent.

For example, the following code checks if the vectors $(1, 0), (1, 0), (0, 1)$ are linearly independent:

```julia
include("gramschmidt.jl")
using GramSchmidt

vectors = [[[1,0], [1,0], [0,1]]
gram_schmidt(vectors)
```
Out[19]:
1-element Array{Any,1}:
 [1.0,0.0]

The output was only the vector (1, 0), so the vectors are not linearly independent.

(a) (1, -1.1), (-2.8, -0.3), (-0.4, 1.5).
(b) (1, 0, 1, 0, 1), (0, 1, 0, 0, 1), (0, 1, 0, 1, 0).
(c) (1, 2, 0), (-2, 0, 3), (1, 0, 2).

20.9 Verifying superposition in Julia. Generate a 10 by 20 matrix $A$, 10-vectors $x$ and $y$, and scalars $\alpha$ and $\beta$. Evaluate $A(\alpha x + \beta y)$ and $\alpha(Ax) + \beta(Ay)$ and verify that they are close.

20.10 Matrix multiplication. Write a Julia script that multiplies matrices $A$ and $B$ to create the product $C = AB$ only using scalar operations, and accessing the individual entries of $A$ and $B$. Compare the results to Julia’s matrix multiplication on some random matrices.

20.11 Matrices. In each part below, use Julia to create the matrix $A$, described by its effect on an arbitrary vector $x$ as $y = Ax$. Check that the matrix you create has the correct behavior when it multiplies a randomly generated vector $x$.

(a) $x$ is a 5-vector; $y$ is a 3-vector containing the first 3 components of $x$.
(b) $x$ and $y$ are 7-vectors; $y$ has the entries of $x$ in reverse order.
(c) $x$ is a 10-vector, and $y$ is the 9-vector of differences of consecutive entries of $x$:

$$y = (x_2 - x_1, x_3 - x_2, \ldots, x_{10} - x_9).$$

(d) $x$ and $y$ are 8-vectors, and $y_i$ is the difference between $x_i$ and the average of $x_1, \ldots, x_i$.


(a) Verify that
\[
\begin{bmatrix}
A & B \\
C & D
\end{bmatrix}^T
\]
is equal to
\[
\begin{bmatrix}
A^T & B^T \\
C^T & D^T
\end{bmatrix}.
\]

(b) Verify that
\[
\begin{bmatrix}
A & B \\
C & D
\end{bmatrix}
\begin{bmatrix}
E & F \\
G & H
\end{bmatrix}
\]
is equal to
\[
\begin{bmatrix}
AE + BG & AF + BH \\
CE + DG & CF + DH
\end{bmatrix}.
\]
20.13 **Column-major order.**

*reshape* changes the dimension of a matrix by moving its entries. It can be used, for example, to vectorize matrices. There are multiple ways this could occur, but here we examine Julia’s implementation.

(a) Reshape the matrix

\[
\begin{bmatrix}
1 & 3 & 5 \\
2 & 4 & 6 \\
\end{bmatrix}
\]

into a (column) vector. Note the order of the entries, and briefly explain what *reshape* is doing.

(b) Now reshape the same matrix into a $3 \times 2$ matrix. Are the entries where you expect?

(c) In a single line of code, express the matrix

\[
\begin{bmatrix}
1 & 2 & 3 & 4 & 5 \\
6 & 7 & 8 & 9 & 10 \\
\end{bmatrix}
\]

using only range construction (i.e., 1:10), reshape, and transpose.

20.14 **Taylor series approximations.** For each of the following functions below, plot the function, $f$ over the interval $[-10, 10]$. Then, on the same graph, plot the function’s Taylor approximation near 0.

(a) $f(x) = x^2 + x.$

(b) $f(x) = 0.5x^4 + 0.3x^2 + x.$

(c) $f(x) = \sin(x).$

20.15 **Checking least squares in Julia.** Create a random $30 \times 10$ matrix $A$ and a random 30-vector $b$, and the compute the least-squares approximate solution $\hat{x}$, and the associated residual error $\|Ax - b\|$. Generate several other random values of $x$, and verify that $\|A\hat{x} - b\| \leq \|Ax - b\|$

20.16 **KKT and least squares.** Solve the following least-norm problem

\[
\begin{align*}
\text{minimize} & \quad \|x\|^2 \\
\text{subject to} & \quad Ax = b 
\end{align*}
\]

using LLS, and then again using KKT and backslash. Confirm that your solutions are nearly identical, and that both satisfy the constraint (up to numerical error). You may generate $A$ and $b$ using the following code.

```
srand(0); A = randn(3,5); b = randn(3,1);
```

20.17 **Checking a classifier.** Write a Julia script that computes the false positive and false negative rates for a given data set, with a given linear classifier.

20.18 **Plotting in Julia.** Create the 100-vectors $x$ and $y$ in Julia where

\[
x_i = \sin(i/8), \quad y_i = \begin{cases} 
0 & i \leq 25 \\
(i - 25)/50 & 25 < i < 75 \\
1 & i \geq 75
\end{cases}
\]

Using MMAPlot, plot $x$. On a second figure, plot $x$, $y$ and $x + y$. Use different colors to distinguish the three lines, and add labels to the lines.