Homework 1

Tuesday, September 22, 2020 10:32 AM

I accidentally deleted this HW from OneNote, so here is a re-imported version from the PDF uploaded to Canvas.

Ari_Porad_QEA_HW 1

HW 1

Friday, September 11, 2020 4:17 PM

HW 1

Vectors

Basics

- Represented by lowercase bold letters **a**
- The point (x, y, z) is interchangeable with the vector extending from the origin to that point: $v = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$
  - This works equally well in non-3 dimensional space
- This is a column vector: $v = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$, this is a row vector: $u = \begin{bmatrix} x & y & z \end{bmatrix}$
  - Converting a column vector v to a row vector (or vis versa) is called transposing and is indicated with a ^{T} superscript. In this case, v^T === u
- Vectors can represent lots of powerful things, and linear algebra gives lots of tools for dealing with them.

· spacial vector = draw it

Multiplication

The product of a row vector and a column vector is:

$uv = \begin{bmatrix} p & q & r \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = px + qy + rz$

Given two column vectors of the same length n, their dot product is:

$v \cdot w = v_1 w_1 + v_2 w_2 + ... + v_n w_n$

The dot product can also be used to measure how aligned the vectors are:

$v \cdot w = ||v|| ||w|| \cos\theta$

Where θ is the angle between v and w and:

$||v|| = \sqrt{v_1^2 + v_2^2 + ... + v_n^2}$

is the absolute length of the vector (for whatever n-dimensional space the vector is in)

NB: The dot product is the same as the product of one times the transpose of the other: $v \cdot w = v^T w$

Exercises

Exercise 3.1

1. Assume **v** and **w** are two vectors of unit length, i.e., $\|\mathbf{v}\| = \|\mathbf{w}\| = 1$. Using the formula above, what angle between **v** and **w** maximizes the dot product? Using the formula above, what angle between **v** and **w** minimizes the dot product?
2. Compute $\mathbf{v} \cdot \mathbf{w}$ where

$$\mathbf{v} = \begin{bmatrix} 1 \\ 3 \\ -4 \\ 6 \end{bmatrix}, \text{ and } \mathbf{w} = \begin{bmatrix} -2 \\ 0 \\ 1 \\ 3 \end{bmatrix}$$

3.1.1: Maximized at 0 and π, minimized at $\frac{\pi}{2}$ and $\frac{3\pi}{2}$

3.1.2: 12

Exercise 3.2

For instance, you may have a three-dimensional vector f whose entries represent the numbers of different fruits you have in your refrigerator. For example, the first entry could be the number of oranges, the second the number of grapefruits and the third could be the number of apples. When organized in this manner, you can use products of row and column vectors to compute the number of different fruits there are. For instance, suppose that

$$\mathbf{f} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad (3.5)$$

i.e. you have 1 orange, 2 grapefruits, and 3 apples in your fridge.

1. Find a row vector t so that the product tf tells you the total number of fruits in your refrigerator.
2. Find a row vector c such that the product cf tells you the total number of *citrus* fruits in your refrigerator.
3. Suppose that in the genetically engineered future, all apples weigh 100 g, all grapefruits weigh 250 g and all oranges weigh 120 g. Find a row vector w, such that the product wf tells you the total weight of fruits in your refrigerator.

3.2.1: $t = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$

3.2.2: $c = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$

3.2.3: $w = \begin{bmatrix} 120 & 250 & 100 \end{bmatrix}$

Linear Algebra

Multiplying vectors together to get information is called *linear combinations*, (in Excercise 3.2 you took linear combinations of the entries of vector f).

Matrices

- Matrices are two-dimensional arrays of numbers that compactly represent linear combinations.
- They can be used lots for lots of things, including to represent data.
- When you multiply a matrix by a vector, you get a new vector.

- This is said by "a matrix operates on a vector", which means matrix * vector.
- Matrices are represented by bold uppercase letters **A**.
- Example: $\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$
- Matrix come in different shapes/sizes, identified by the number of rows/columns. Arbitrary matrix **A** has m rows and n columns, therefore being an $m \times n$ matrix.

- Vectors are matrices: Row vectors are $1 \times n$ matrices, col vectors are $m \times 1$ matrices.
- An $m \times n$ matrix can only operate on an $n \times 1$ col vector & produces an $m \times 1$ col vector
- An $m \times n$ matrix can only act on an $n \times k$ matrix and produces an $m \times k$ matrix. ?

Example:

3x2 Matrix $\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 3 & -1 \\ 0 & 4 \end{bmatrix}$

col vec $\mathbf{V} = \begin{bmatrix} -2 \\ 1 \end{bmatrix}$

$$\mathbf{W} = \mathbf{Av} = \begin{bmatrix} 2 & 1 \\ 3 & -1 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} -2 \\ 1 \end{bmatrix} = \begin{bmatrix} -3 \\ -7 \\ 4 \end{bmatrix}$$

col 1 each multiplied by row 1

$$\begin{bmatrix} 2 & 1 \\ 3 & -1 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} -2 \\ 1 \end{bmatrix}$$

Add results for each Row:
$(2)(-2)+(1)(1)=-3$

col 2 each multiplied by Row 2

OR

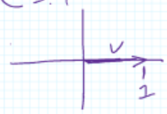
Each item in **W** is dot prod of one transposed row of **A** & **v**

Exercise 3.3: What is **Gf**?

One vector is # of all fruits, one is citrus fruits

Exercise 3.4:

1. $v = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$



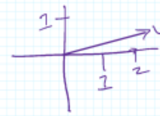
2. $w = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$

3. The angle changed to 45° & mag changed too

4. $u = Bv$

$u = \begin{bmatrix} \cos 30° & -\sin 30° \\ \sin 30° & \cos 30° \end{bmatrix} v$

$= \begin{bmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} v = \begin{bmatrix} \sqrt{3}/2 \\ 1/2 \end{bmatrix}$



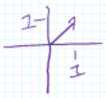
5. Angle changed to 30°

6. $t = Rv$

$= \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$

7. Equal to unit vector of angle θ

8. $w = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$



9. $s = Rw$

$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta - \sin\theta \\ \sin\theta + \cos\theta \end{bmatrix}$

10. changes the angle

$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$ is a rotation matrix, because it rotates vectors when multiplied. See book for proof.

When multiplying generic matrecies A (m×n) and v (n×1) to get w (m×1)

Homework Page 3

When multiplying generic matrecies

A $(m \times n)$ and $\underline{v}$ $(n \times 1)$ to get w $(m \times 1)$,

each element $w_i = a_{i1} + \underline{v}_1 + a_{i2}\underline{v}_2 + \dots$

$$= \sum_{j=1}^{n} a_{ij} \underline{v}_j$$

Adding, subtracting & transposing

Addition & subtraction are done element-wise & must be same size

$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ $\quad B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$ ↳ Produces same size

$A - B = \begin{bmatrix} 1-5 & 2-6 \\ 3-7 & 4-8 \end{bmatrix} = \begin{bmatrix} -4 & -4 \\ -4 & -4 \end{bmatrix}$

Multipling by a scalar just multiplies each element:

$3A = \begin{bmatrix} 3 & 6 \\ 9 & 12 \end{bmatrix}$

Transposing:

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$

Important identity:

$(Av)^T = A^T v^T$

Exercises

3.5)

$A = \begin{bmatrix} 3 & 4 & 1 \\ 3 & 1 & 1 \end{bmatrix}$ $\quad B = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 2 & 1 \end{bmatrix}$

$4A - 5B = \begin{bmatrix} 12 & 16 & 4 \\ 12 & 4 & 4 \end{bmatrix} - \begin{bmatrix} 5 & 10 & 15 \\ 10 & 10 & 5 \end{bmatrix}$

$$\begin{bmatrix} \cdots \\ 12 & 4 & 4 \end{bmatrix} - \begin{bmatrix} 5 & 10 & 15 \\ 10 & 10 & 5 \end{bmatrix}$$

$$= \begin{bmatrix} 7 & 6 & -11 \\ 2 & -6 & -1 \end{bmatrix}$$

3.6) If A is 4×5, A^T is 5×4

3.7) 5×1, 1×5

Products of Matricies

$$A = \begin{bmatrix} 2 & 1 \\ 3 & -1 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 5 \\ -2 & 3 \end{bmatrix}$$

$$C = AB = \begin{bmatrix} 2 & 1 \\ 3 & -1 \end{bmatrix} \begin{bmatrix} 1 & 5 \\ -2 & 3 \end{bmatrix}$$

$$= \begin{bmatrix} (2)(1)+(1)(-2) & (2)(5)+(1)(3) \\ (3)(1)+(-1)(-2) & (3)(5)+(-1)(3) \end{bmatrix}$$

$$= \begin{bmatrix} A_{x1} \cdot B_{1x} & A_{x1} \cdot B_{2x} \\ A_{x2} \cdot B_{1x} & A_{x2} \cdot B_{2x} \end{bmatrix}$$

OR:

$C = AB$

$C_{ij} = A_{xj} \cdot B_{ix}$

x means all

Example 2:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 2 \\ 4 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 4 \\ 2 & 3 \end{bmatrix}$$

$C = AB$

$$= \begin{bmatrix} (1)(1)+(2)(2) & (1)(4)+(2)(3) \\ (3)(1)+(2)(2) & (3)(4)+(2)(3) \\ (4)(1)+(1)(2) & (4)(4)+(1)(3) \end{bmatrix}$$

OR: Each col of B is a vector, multiply each by A, each output is a col of C

NB: Matrix multiplication isn't commutative:

$AB \neq BA$

However, it is Distributive:

$A(B+C) = AB + AC$

$(B+C)A = BA + CA$

For square matrices, exponents work!

Exercises

$A = \begin{bmatrix} -2 & 4 \\ 0 & 3 \end{bmatrix}$ $B = \begin{bmatrix} 5 & -3 \\ -1 & -1 \end{bmatrix}$ $C = \begin{bmatrix} -5 & -1 \\ -3 & 2 \end{bmatrix}$

3.8: $AB = \begin{bmatrix} (-2)(5)+(4)(-1) & (-2)(-3)+(4)(-1) \\ (0)(5)+(3)(-1) & (0)(-3)+(3)(-1) \end{bmatrix}$

$= \begin{bmatrix} -14 & 2 \\ -3 & -3 \end{bmatrix}$

3.9: $BA = [(5)(-2)+(-3)(0) \quad (5)(4)+(-3)(3)$

$$3.9: BA = \begin{bmatrix} (5)(-2)+(-3)(0) & (5)(4)+(-3)(3) \\ (-1)(-2)+(-1)(0) & (-1)(4)+(-1)(3) \end{bmatrix}$$

$$= \begin{bmatrix} -10 & 11 \\ 2 & -7 \end{bmatrix}$$

$$3.10: A(B+C) = A \begin{bmatrix} 0 & -4 \\ -4 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} (-2)(0)+(4)(-4) & (-2)(-4)+(4)(1) \\ (0)(0)+(3)(-4) & (0)(-4)+(3)(1) \end{bmatrix}$$

$$= \begin{bmatrix} -16 & 12 \\ -12 & 3 \end{bmatrix}$$

$$3.11: AB + AC =$$

$$= \begin{bmatrix} -2 & 4 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 5 & -3 \\ -1 & -1 \end{bmatrix} + \begin{bmatrix} -2 & 4 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} -5 & -1 \\ -3 & 2 \end{bmatrix}$$

$$= \begin{bmatrix} -14 & 2 \\ -3 & -3 \end{bmatrix} + \begin{bmatrix} -2 & 10 \\ -9 & 6 \end{bmatrix}$$

$$= \begin{bmatrix} -16 & 12 \\ -12 & 3 \end{bmatrix} \text{ Same } \checkmark$$

3.12:

$$1)\ A^2 = \begin{bmatrix} -2 & 4 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} -2 & 4 \\ 0 & 3 \end{bmatrix}$$

$$= \begin{bmatrix} 4 & 4 \\ 0 & 9 \end{bmatrix}$$

2) $B^2 = \begin{bmatrix} 5 & -3 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} 5 & -3 \\ -1 & -1 \end{bmatrix}$

$$= \begin{bmatrix} 28 & -12 \\ -4 & 4 \end{bmatrix}$$

13.3)

1. Square Matrix [of order n]: an $n \times n$ matrix (same # of rows & colums)
2. Rectangular Matrix: any non-square Matrix
3. Diagonal Matrix: a (usually square) matrix where all values not on the main diagonal (top left → bottom right) are zero. The diagonal values may, but need not, be equal to each other.
   Ex: $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & 2 \end{bmatrix}$
4. Identity Matrix: A diagonal matrix where all nonzero values are 1. May be of any size, often inferred from context. Denoted as $\mathbf{I}$, $\mathbb{1}$, or $\mathbf{I}_n$
   For A $m \times n$: $\mathbf{I}_m A = A \mathbf{I}_n = A$
   (Multiplying by a matrix gives the same matrix.)

Homework Page 8

... a matrix gives the same matrix.)

5. Symmetric Matrix: A square matrix equal to its transpose: $A^T = A$

$$A_{ij} = A_{ji} \ \forall i, j$$

Ex: $\begin{bmatrix} 1 & 7 & 3 \\ 7 & 4 & -5 \\ 3 & -5 & 6 \end{bmatrix}$

There are well-known matricies that do speeific transformations on spacial vectors.

Exercise 3.14:

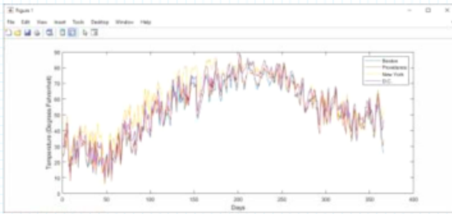
$DM = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{bmatrix}$

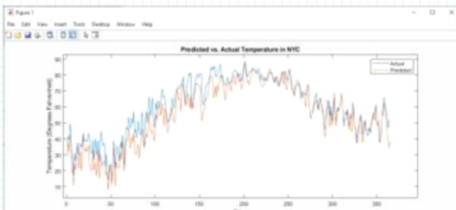
2) $\begin{bmatrix} 3 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

3) $\begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix}$

Exercise 3.16:

1. T is 4x7670. Each city is one row, each column represents one sample.
2. Done, dimensions are right.





MATLAB Command Window   Page 1

```
>> load temps.mat
>> size(T)

ans =

           4        7670

>> fprintf("Each city is one row, each column is one data point.\n")
Each city is one row, each column is one data point.
>> t1 = T(1,:);
>> t2 = T(2,:);
>> t3 = T(3,:);
>> t4 = T(4,:);
>> mean(t1)

ans =

   51.7667

>> mean(t2)

ans =

   51.9140

>> mean(t3)

ans =

   58.4365

>> mean(t4)

ans =

   55.9451

>> printf("Boston and Providence presumably have nearly the same weather, so the must be
t1 and t2.\n")
Check for missing argument or incorrect argument data type in call to function 'printf'.

>> fprintf("Boston and Providence presumably have nearly the same weather, so the must be
t1 and t2.\n")
Boston and Providence presumably have nearly the same weather, so the must be t1 and t2.
>> fprintf("*they must be t1 and t2\n")
*they must be t1 and t2
>> fprintf("The question mentioned geography, so I assume the theory is that temprature
is strictly inversely correlated with latitude.\n")
The question mentioned geography, so I assume the theory is that temprature is strictly
inversely correlated with latitude.
>> fprintf("Sorry, temperature is inversely correlated with distance from the equator,
```

MATLAB Command Window Page 2

```
positively correlated with latitude. (North = cold, South = warm)")
Sorry, temperature is inversely correlated with distance from the equator, positively
correlated with latitude. (North = cold, South = warm)>>
>> fprintf("Ergo, t1 = Boston, t2 = Providence (very similar to Boston because they're
very close), t3 = NYC, t4 = DC")
Ergo, t1 = Boston, t2 = Providence (very similar to Boston because they're very close),
t3 = NYC, t4 = DC>>
>> boston = t1; providence = t2; nyc = t3; dc = t4;
>> max(boston)

ans =

   96.7000

>> min(boston)

ans =

    5.7000

>> plot(boston)
>> plot(T)
>> plot(boston, nyc, providence, dc)
>> plot(boston); hold on;
>> plot(nyc);
>> plot(providence);
>> plot(dc)
>> legend("Boston", "New York", "Providence", "D.C.")
>> fprintf("Oops! Forgot to limit it to the last year! Restarting...\n");
Oops! Forgot to limit it to the last year! Restarting...
>> hold off;
>> plot(boston); hold on; plot(providence); plot(nyc); plot(dc);
>> hold off;
>> plot(boston(:,end-365):); hold on;
 plot(boston(:,end-365):); hold on;
                       ↑
Error: Invalid expression. When calling a function or indexing a variable, use
parentheses. Otherwise, check for mismatched delimiters.

>> plot(boston(:,-365):); hold on;
 plot(boston(:,-365):); hold on;
                    ↑
Error: Invalid expression. When calling a function or indexing a variable, use
parentheses. Otherwise, check for mismatched delimiters.

>> plot(boston(:,(end-365):); hold on;
 plot(boston(:,(end-365):); hold on;
                         ↑
Error: Invalid expression. When calling a function or indexing a variable, use
parentheses. Otherwise, check for mismatched delimiters.
```

MATLAB Command Window Page 3

```
>> plot(boston(:,(end-365):)); hold on;
 plot(boston(:,(end-365):)); hold on;
                         ↑
Error: Invalid expression. When calling a function or indexing a variable, use
parentheses. Otherwise, check for mismatched delimiters.

>> plot(boston(:,(end-365)); hold on;
 plot(boston(:,(end-365)); hold on;
                          ↑
Error: Invalid expression. When calling a function or indexing a variable, use
parentheses. Otherwise, check for mismatched delimiters.

>> plot(boston(:,-365):; hold on;
 plot(boston(:,-365):; hold on;
                    ↑
Error: Invalid expression. When calling a function or indexing a variable, use
parentheses. Otherwise, check for mismatched delimiters.

>> plot(boston(:,end:-365); hold on;
 plot(boston(:,end:-365); hold on;
                        ↑
Error: Invalid expression. When calling a function or indexing a variable, use
parentheses. Otherwise, check for mismatched delimiters.

>> plot(boston(:,end-365:); hold on;
 plot(boston(:,end-365:); hold on;
                       ↑
Error: Invalid expression. When calling a function or indexing a variable, use
parentheses. Otherwise, check for mismatched delimiters.

>> plot(boston(:,end-365:)); hold on;
 plot(boston(:,end-365:)); hold on;
                       ↑
Error: Invalid expression. When calling a function or indexing a variable, use
parentheses. Otherwise, check for mismatched delimiters.

>> plot(boston(:,-365:)); hold on;
 plot(boston(:,-365:)); hold on;
                    ↑
Error: Invalid expression. When calling a function or indexing a variable, use
parentheses. Otherwise, check for mismatched delimiters.

>> plot(boston(:,-365:end)); hold on;
Index in position 2 is invalid. Array indices must be positive integers or logical
values.

>> plot(boston(:,end-365:end)); hold on;
>> fprintf("There we go! (I know we could have used size() and subtracted, but that's a
code smell.")
There we go! (I know we could have used size() and subtracted, but that's a code smell.>>
>> plot(providence(:,end-365:end)); hold on;
```

MATLAB Command Window

Page 4

predict_data =

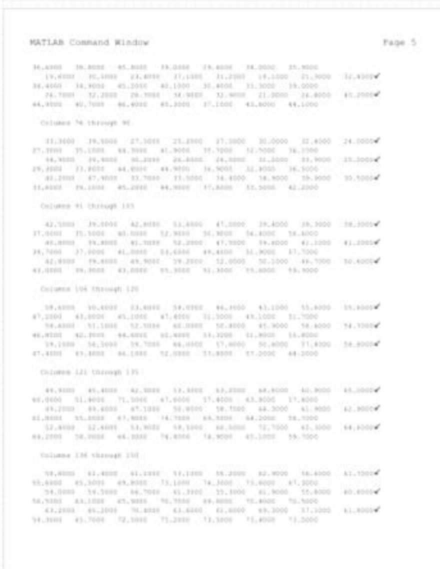
Columns 1 through 15

Columns 16 through 30

Columns 31 through 45

Columns 46 through 60

Columns 61 through 75



MATLAB Command Window

Page 5

Columns 76 through 90

Columns 91 through 105

Columns 106 through 120

Columns 121 through 135

Columns 136 through 150

MATLAB Command Window Page 6

Columns 151 through 165

[illegible]

Columns 166 through 180

[illegible]

Columns 181 through 195

[illegible]

Columns 196 through 210

[illegible]

Columns 211 through 225

[illegible]

Columns 226 through 240

[illegible]

MATLAB Command Window Page 7

[illegible]

Columns 241 through 255

[illegible]

Columns 256 through 270

[illegible]

Columns 271 through 285

[illegible]

Columns 286 through 300

[illegible]

Columns 301 through 315

[illegible]

Columns 316 through 330

[illegible]

Homework Page 13

MATLAB Command Window Page 8

```
[illegible numeric output]

  Columns [illegible] through [illegible]

[illegible numeric output]

  Columns [illegible] through [illegible]

[illegible numeric output]

  Columns [illegible] through [illegible]

[illegible numeric output]

>> size(predict_data)

ans =

     1   366

>> predict_data = T([1 2 4],end-364:end);
>> fprintf("[illegible]")
[illegible]>> size(predict_data)

ans =

     1   365

>> predict_result = transpose(transpose(predict_data) * [illegible]);
Error using *
Incorrect dimensions for matrix multiplication. Check that the number of columns in the
first matrix matches the number of rows in the second matrix. To
perform elementwise multiplication, use '.*'.

>> predict_result = transpose(transpose(predict_data) * [illegible]);
```

MATLAB Command Window Page 9

```
>> size(predict_result)

ans =

     1   365

>> hold on;
>> hold off;
>> plot(ayo)
>> hold on;
>> title("Predicted vs. Actual Temperature in [illegible]");
>> fprintf("Forgot to limit to last year... restarting")
Forgot to limit to last year... restarting>> holf off;
Unrecognized function or variable 'holf'.

Did you mean:
>> hold off;
>> plot(ayo(:,end-364:end)); hold on;
>> plot(predict_result)
>> title("Predicted vs. Actual Temperature in [illegible]");
>> label("Actual", "Predicted")
Check for missing argument or incorrect argument data type in call to function 'label'.

>> legend("Actual", "Predicted")
>> xlabel("Days")
>> ylabel("Temperature (Degrees Fahrenheit)")
>>
```

Homework Page 14