CPSC 340 A1

Chuan Shao Foo

TOTAL POINTS

53.5 / 67

QUESTION 1

Linear Algebra Review 16 pts

1.1 Basic Operations 5 / 7

- ✓ **- 1 pts 1.1.1** ▯
- **- 1 pts** 1.1.2 ▯
- **- 1 pts** 1.1.3 ▯
- **- 1 pts** 1.1.4 ▯
- **- 1 pts** 1.1.5 ▯
- **- 1 pts** 1.1.6 ▯
- ✓ **- 1 pts 1.1.7** ▯
- **- 0 pts** ✓

1.2 Matrix Algebra Rules 8 / 9

- ✓ **+ 1 pts True**
- ✓ **+ 1 pts False**
- ✓ **+ 1 pts False**
- ✓ **+ 1 pts False**
- ✓ **+ 1 pts True**
- ✓ **+ 1 pts False**
- **+ 1 pts** True
- ✓ **+ 1 pts True**
- ✓ **+ 1 pts True**
- **+ 0 pts** No answer

QUESTION 2

Probability Review 9 pts

2.1 Rules of Probability 3 / 4

- **- 0 pts** Correct
- **- 1 pts** 2.1.1 incorrect
- ✓ **- 1 pts 2.1.2 incorrect**
- **- 1 pts** 2.1.3(a) incorrect
- **- 1 pts** 2.1.3(b) incorrect
- **- 4 pts** Incorrect / blank

2.2 Bayes Rule and Conditional Probability 5 / 5

- ✓ **- 0 pts Correct**
- **- 1 pts** 2.2.1 incorrect
- **- 1 pts** 2.2.2 incorrect
- **- 1 pts** 2.2.3 incorrect
- **- 1 pts** 2.2.4 incorrect
- **- 1 pts** 2.2.5 incorrect
- **- 5 pts** Incorrect / blank
- **- 0.5 pts** partially incorrect

QUESTION 3

Calculus Review 11 pts

3.1 One-variable derivatives 2.5 / 3

- **- 0 pts** Correct
- **- 1 pts** 3.1.1 Incorrect
- **- 1 pts** 3.1.2 Incorrect
- **- 1 pts** 3.1.3 Incorrect
- ✓ **- 0.5 pts 3.1.3 Not Simplified**

3.2 Multivariable derivatives 5 / 5

- ✓ **- 0 pts Correct**
- **- 0.5 pts** 3.2.1 Minor error
- **- 1 pts** 3.2.1 Incorrect
- **- 0.5 pts** 3.2.2 Minor error
- **- 1 pts** 3.2.2 Incorrect
- **- 0.5 pts** 3.2.3 Minor error
- **- 1 pts** 3.2.3 Incorrect
- **- 0.5 pts** 3.2.4 Minor error
- **- 1 pts** 3.2.4 Incorrect
- **- 0.5 pts** 3.2.5 Minor error
- **- 1 pts** 3.2.5 Incorrect

3.3 Derivatives of code 3 / 3

- ✓ **- 0 pts Correct**

- **1 pts** Grad1 incorrect
- **1 pts** Grad2 incorrect
- **1 pts** Grad3 incorrect

QUESTION 4

Algorithms and Data Structures Review

13 pts

4.1 Trees 2 / 2

✓ - **0 pts Correct**
- **2 pts** incorrect
- **1 pts** #nodes incorrect
- **1 pts** #leaves incorrect

4.2 Common Runtimes 5 / 7

- **1 pts** 1 incorrect
- **1 pts** 2 incorrect
- **1 pts** 3 incorrect
- **1 pts** 4 incorrect
- **1 pts** 5 incorrect

✓ - **1 pts 6 incorrect**
✓ - **1 pts 7 incorrect**
- **0 pts** All correct

4.3 Running times of code 3 / 4

- **1 pts** 1 incorrect
- **1 pts** 2 incorrect
- **1 pts** 3 incorrect
- **1 pts** 4 incorrect

✓ - **1 pts 5 incorrect**
- **0 pts** Click here to replace this description.

QUESTION 5

Summary Statistics and Data Visualization 12 pts

5.1 Summary Statistics 4 / 5

- **0 pts** Correct

✓ - **1 pts Incorrect explanation for better measure of the mode isn't reliable, or missing explanation**
- **1 pts** Computed statistics separately for each dataset
- **1 pts** Missing max/min values
- **5 pts** Incorrect or no answer

5.2 Data Visualization 3 / 6

- **0 pts** Correct

✓ - **1 pts No explanations**
✓ - **2 pts 2 incorrect matches**
- **1 pts** 1 incorrect match
- **3 pts** 3 incorrect matches
- **6 pts** Incorrect or missing answer

5.3 Decision Surfaces 1 / 1

✓ - **0 pts Correct**
- **1 pts** Missing answer or incorrect

QUESTION 6

Decision Trees 6 pts

6.1 Equality vs. Inequality Splitting Rules 1 / 1

✓ - **0 pts Correct**
- **1 pts** Incorrect/blank
- **0.5 pts** Partial answer

6.2 Decision Stump Implementation 2 / 2

✓ - **0 pts Correct**
- **1 pts** Incorrect/Missing error
- **2 pts** Incorrect/Blank

6.3 Decision Tree to Program 0 / 2

- **0 pts** Correct
- **0.5 pts** Incorrect/Missing 37.669
- **0.5 pts** Incorrect/Missing -96.0901
- **0.5 pts** Incorrect/Missing -115.5776
- **0.5 pts** Incorrect Order

✓ - **2 pts Incorrect/Blank**
- **1 pts** Not using the hard coded values

6.4 Constructing Decision Trees 1 / 1

✓ - **0 pts Correct**
- **1 pts** Incorrect/Blank
- **0.5 pts** Incorrect/missing error
- **0.5 pts** Incorrect/missing graph

1 Linear Algebra Review

For these questions you may find it helpful to review these notes on linear algebra:
`http://www.cs.ubc.ca/~schmidtm/Documents/2009_Notes_LinearAlgebra.pdf`

1.1 Basic Operations

Use the definitions below,

$$\alpha = 2, \quad x = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}, \quad y = \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix}, \quad A = \begin{bmatrix} 3 & 2 & 2 \\ 1 & 3 & 1 \\ 1 & 1 & 3 \end{bmatrix}, \quad I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and use x_i to denote element i of vector x. Evaluate the following expresions, showing at least one intermediate step of work:

1. $\|x\|$ (Euclidean norm of x).

   Answer: $\|x\| = \sqrt{0^2 + 1^2 + 2^2} = 5$

2. $\alpha(x + y)$ (vector addition and scalar multiplication).

   Answer: $(\text{x+y}) = \begin{bmatrix} 3 \\ 5 \\ 7 \end{bmatrix}$

   $\alpha(x + y) = \begin{bmatrix} 6 \\ 10 \\ 14 \end{bmatrix}$

3. $x^T y = \sum_{i=1}^{n} x_i y_i$ (inner product).

   Answer: The statement is true. x^T is a 1x3 matrix, and y is a 3x1 matrix, which produces a 1x1 matrix which sums up all the multiplication of x_i and y_i

   $x^T y = \begin{bmatrix} 0 * 3 + 1 * 4 + 2 * 5 \end{bmatrix} = [14]$

4. xy^T (outer product).

   Answer: $y^T = \begin{bmatrix} 3 & 4 & 5 \end{bmatrix}$

   $xy^T = \begin{bmatrix} 0 & 0 & 0 \\ 3 & 4 & 5 \\ 6 & 8 & 10 \end{bmatrix}$

5. Ax (matrix-vector multiplication).

   Answer: $\text{Ax} = \begin{bmatrix} 3 * 0 + 2 * 1 + 2 * 2 \\ 1 * 0 + 3 * 1 + 1 * 2 \\ 1 * 0 + 1 * 1 + 3 * 2 \end{bmatrix} = \begin{bmatrix} 6 \\ 5 \\ 7 \end{bmatrix}$

6. $x^T A x$ (quadratic form).

   Answer: $x^T A x = \begin{bmatrix} 0 & 1 & 2 \end{bmatrix} * \begin{bmatrix} 6 \\ 5 \\ 7 \end{bmatrix} = \begin{bmatrix} 19 \end{bmatrix}$

7. Solve for a vector v that satisfies $(I_3 - xx^T)v = y$ (linear system).

Answer: $xx^T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 2 & 4 \end{bmatrix}$

$I_3 - xx^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -2 \\ 0 & -2 & 3 \end{bmatrix}$

$(I_3 - xx^T)^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -3/4 & -1/2 \\ 0 & -1/2 & 0 \end{bmatrix}$

$v = (I_3 - xx^T)^{-1} y = \begin{bmatrix} 3 \\ -5.5 \\ -2 \end{bmatrix}$

1.1 Basic Operations 5 / 7

- ✓ - 1 pts **1.1.1**
- - 1 pts 1.1.2
- - 1 pts 1.1.3
- - 1 pts 1.1.4
- - 1 pts 1.1.5
- - 1 pts 1.1.6
- ✓ - 1 pts **1.1.7**
- - 0 pts ✓

1.2 Matrix Algebra Rules

Assume that $\{x, y, z\}$ are $n \times 1$ column vectors and $\{A, B, C\}$ are $n \times n$ real-valued matrices, 0 is the zero matrix of appropriate size, and I is the identity matrix of appropriate size. State whether each of the below is true in general (you do not need to show your work).

1. $x^T x = \|x\|^2$.

   Answer: True, it is simply x_i^2 summed without the root

2. $x^T x = x x^T$.

   Answer: False, the matrix size would be different

3. $(x - y)^T (y - x) = \|x\|^2 - 2x^\top y + \|y\|^2$.

   Answer: False, the sign for the right hand side is meant to be negative

4. $AB = BA$.

   Answer: False, matrix multiplication is not commutative

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

   Answer: True, matrix multiplication is distributive over addition

6. $(AB)^T = A^T B^T$.

   Answer: False, the reverse for $B^T A^T$ would however be true

7. $x^T A y = y^T A^T x$.

   Answer: False, the right hand side of the equation is missing an overall transpose

8. $A^n = (A^n)^T$ for any non-negative integer n if A is symmetric.

   Answer: True, n symmetric matrices multiplied gives a symmetric matrix which can be transposed and give the same result

9. $A^\top A = I$ if the columns of A are orthonormal.

   Answer: True, that is the definition of orthogonal matrices

1.2 Matrix Algebra Rules 8 / 9

- ✓ + **1 pts** True
- ✓ + **1 pts** False
- ✓ + **1 pts** False
- ✓ + **1 pts** False
- ✓ + **1 pts** True
- ✓ + **1 pts** False
- \+ **1 pts** True
- ✓ + **1 pts** True
- ✓ + **1 pts** True
- \+ **0 pts** No answer

2 Probability Review

For these questions you may find it helpful to review these notes on probability:
`http://www.cs.ubc.ca/~schmidtm/Courses/Notes/probability.pdf`
And here are some slides giving visual representations of the ideas as well as some simple examples:
`http://www.cs.ubc.ca/~schmidtm/Courses/Notes/probabilitySlides.pdf`

2.1 Rules of probability

Answer the following questions. You do not need to show your work.

1. Consider two events A and B such that $\Pr(A, B) = 0$ (they are mutually exclusive). If $\Pr(A) = 0.4$ and $\Pr(A \cup B) = 0.95$, what is $\Pr(B)$? Note: $p(A, B)$ means "probability of A and B" while $p(A \cup B)$ means "probability of A or B". It may be helpful to draw a Venn diagram.

   Answer: 0.55

2. Instead of assuming that A and B are mutually exclusive ($\Pr(A, B) = 0$), what is the answer to the previous question if we assume that A and B are independent?

   Answer: 0.55

3. You are offered the opportunity to play the following game: first your opponent rolls a 4 - sided dice and records the outcome r_1. Then you roll a $(5 - r_1)$ - sided dice and record the outcome r_2. Your payout is $r_1 + r_2 - 1$ dollars. You can enter the game either before or after your opponent's turn.

   - If you enter *after* your opponent's turn you know r_1. What is a fair price for a ticket in this case, i.e., what is the expected payout as a function of r_1?

     Answer: E(payout) $= 2 + 0.5r_1$ for $0 < r_1 < 5$

   - If you enter *before* your opponent's turn you do not know r_1. What is the expected payout now?

     Answer: 3.25 from summing and dividing equally

2.1 Rules of Probability 3 / 4

- **- 0 pts** Correct
- **- 1 pts** 2.1.1 incorrect
- ✓ **- 1 pts 2.1.2 incorrect**
- **- 1 pts** 2.1.3(a) incorrect
- **- 1 pts** 2.1.3(b) incorrect
- **- 4 pts** Incorrect / blank

2.2 Bayes Rule and Conditional Probability

Answer the following questions. You do not need to show your work.

Suppose a drug test produces a positive result with probability 0.95 for drug users, $P(T = 1|D = 1) = 0.95$. It also produces a negative result with probability 0.99 for non-drug users, $P(T = 0|D = 0) = 0.99$. The probability that a random person uses the drug is 0.0001, so $P(D = 1) = 0.0001$.

1. What is the probability that a random person would test positive, $P(T = 1)$?

   Answer: $P(T = 1) = P(T = 1|D = 1) * P(D = 1) + P(T = 1|D = 0) * P(D = 0) = 0.010094$

2. In the above, do most of these positive tests come from true positives or from false positives?

   Answer: Most of them come from false positives

3. What is the probability that a random person who tests positive is a user, $P(D = 1|T = 1)$?

   Answer: $P(D = 1|T = 1) = P(T = 1|D = 1) * P(D = 1)/P(T = 1) = 0.009411531$

4. Suppose you have given this test to a random person and it came back positive, are they likely to be a drug user?

   Answer: No

5. What is one factor you could change to make this a more useful test?

   Answer: A way could be to increase the number of drug users :)

2.2 Bayes Rule and Conditional Probability 5 / 5

- ✓ **- 0 pts** Correct
- **- 1 pts** 2.2.1 incorrect
- **- 1 pts** 2.2.2 incorrect
- **- 1 pts** 2.2.3 incorrect
- **- 1 pts** 2.2.4 incorrect
- **- 1 pts** 2.2.5 incorrect
- **- 5 pts** Incorrect / blank
- **- 0.5 pts** partially incorrect

3 Calculus Review

For these questions you may find it helpful to review these notes on calculus:
`http://www.cs.ubc.ca/~schmidtm/Courses/Notes/calculus.pdf`

3.1 One-variable derivatives

Answer the following questions. You do not need to show your work.

1. Find the derivative of the function $f(x) = 3x^2 - 2x + 5$.

   Answer: $6x - 2$

2. Find the derivative of the function $f(x) = x^2 \cdot \exp(x)$.

   Answer: $\exp(x) \cdot (x^2 + 2x)$

3. Let $p(x) = \frac{1}{1+\exp(-x)}$ for $x \in \mathbb{R}$. Compute the derivative of the function $f(x) = x - \log(p(x))$ and simplify it by using the function $p(x)$.

   Answer: $1 - p'(x)/p(x) = 1 - \exp(-x) \cdot p(x)$

Note that in this course we will use $\log(x)$ to mean the "natural" logarithm of x, so that $\log(\exp(1)) = 1$. Also, observe that $p(x) = 1 - p(-x)$ for the final part.

3.1 One-variable derivatives 2.5 / 3

- **0 pts** Correct
- **1 pts** 3.1.1 Incorrect
- **1 pts** 3.1.2 Incorrect
- **1 pts** 3.1.3 Incorrect

✓ **- 0.5 pts 3.1.3 Not Simplified**

3.2 Multi-variable derivatives

Compute the gradient $\nabla f(x)$ of each of the following functions. You do not need to show your work.

1. $f(x) = x_1^2 + \exp(x_2)$ where $x \in \mathbb{R}^2$.

   Answer: $\begin{bmatrix} 2x_1 \\ \exp(x_2) \end{bmatrix}$

2. $f(x) = \exp(x_1 + x_2x_3)$ where $x \in \mathbb{R}^3$.

   Answer: $\begin{bmatrix} \exp(x_1 + x_2x_3) \\ \exp(x_1 + x_2x_3) \cdot x_3 \\ \exp(x_1 + x_2x_3) \cdot x_2 \end{bmatrix}$

3. $f(x) = a^T x$ where $x \in \mathbb{R}^2$ and $a \in \mathbb{R}^2$.

   Answer: $\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$

4. $f(x) = x^\top Ax$ where $A = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$ and $x \in \mathbb{R}^2$.

   Answer: $\begin{bmatrix} 4x_1 - 2x_2 \\ 2x_2 - 2x_1 \end{bmatrix}$

5. $f(x) = \frac{1}{2}\|x\|^2$ where $x \in \mathbb{R}^d$.

   Answer: x, the result of the right hand side is simply $1/2(x_1^2 + x_2^2 + ... + x_d^2)$

Hint: it is helpful to write out the linear algebra expressions in terms of summations.

3.2 Multivariable derivatives **5 / 5**

- ✓ **- 0 pts** **Correct**
- **- 0.5 pts** 3.2.1 Minor error
- **- 1 pts** 3.2.1 Incorrect
- **- 0.5 pts** 3.2.2 Minor error
- **- 1 pts** 3.2.2 Incorrect
- **- 0.5 pts** 3.2.3 Minor error
- **- 1 pts** 3.2.3 Incorrect
- **- 0.5 pts** 3.2.4 Minor error
- **- 1 pts** 3.2.4 Incorrect
- **- 0.5 pts** 3.2.5 Minor error
- **- 1 pts** 3.2.5 Incorrect

3.3 Derivatives of code

For these questions you may find it helpful to review this list of useful Julia commands:
`http://www.cs.ubc.ca/~schmidtm/Courses/Notes/juliaCommands.txt`

The zip file `a1.zip` contains a Julia file named `grads.jl` which defines several functions that take in a vector as input. Complete the functions `grad1`, `grad2`, and `grad3` (which compute the gradients of `func1`, `func2`, and `func3`). Include the code in the PDF file for this section, and also in your zip file.

Hint: for many people it's easiest to first understand on paper what the code is doing, then compute the gradient, and then translate this gradient back into code. We've given you `func0` and `grad0` as an example. Also, we've provided the function `numGrad` which approximates the gradient numerically to help you debug. Below is an example of using these functions:

Answer:

```
function grad1(x)
    n = length(x);
    g = zeros(n);
    for i in 1:n
        # Put gradient code here
        g[i] = x[i] - i
    end
    return g
end
```

```
function grad2(x)
    n = length(x);
    g = zeros(n);
    # Put gradient code here
    for i in 1:n
        g[i] = i
    end
    return g
end
```

```
function grad3(x)
    # Put gradient code here
    g = 2*max.(x,0)
    return g
end
```

Note: do not worry about the distinction between row vectors and column vectors here. For example, if the correct answer is a vector of length 5, we'll accept vectors of size 5×1 or 1×5. In future assignments we will be more careful to always use column vectors.

3.3 Derivatives of code 3 / 3

- ✓ **- 0 pts** **Correct**
- **- 1 pts** Grad1 incorrect
- **- 1 pts** Grad2 incorrect
- **- 1 pts** Grad3 incorrect

4 Algorithms and Data Structures Review

For these questions you may find it helpful to review these notes on big-O notation:
`http://www.cs.ubc.ca/~schmidtm/Courses/Notes/bigO.pdf`

4.1 Trees

Answer the following questions You do not need to show your work.

1. What is the maximum number of *leaves* you could have in a binary tree of depth l?

   Answer: 2^l

2. What is the maximum number of *internal nodes* (excluding leaves) you could have in a binary tree of depth l?

   Answer: $2^l - 1$

Note: we'll use the standard convention that the leaves are not included in the depth, so a tree with depth 1 has 3 nodes with 2 leaves.

4.1 Trees 2 / 2

- ✓ **- 0 pts** Correct
- **- 2 pts** incorrect
- **- 1 pts** #nodes incorrect
- **- 1 pts** #leaves incorrect

4.2 Common Runtimes

Answer the following questions using big-O notation You do not need to show your work.

1. What is the cost of running the mergesort algorithm to sort a list of n numbers?

   Answer: $O(nlogn)$

2. What is the cost of finding the third-largest element of an unsorted list of n numbers?

   Answer: $O(n)$

3. What is the cost of finding the smallest element greater than 0 in a *sorted* list with n numbers?

   Answer: $O(logn)$

4. What is the cost of finding the value associated with a key in a hash table with n numbers? (Assume the values and keys are both scalars.)

   Answer: $O(1)$

5. What is the cost of computing the matrix-vector product Ax when A is $n \times d$ and x is $d \times 1$?

   Answer: $O(nd)$

6. What is the cost of computing the quadratic form $x^T Ax$ when A is $d \times d$ and x is $d \times 1$?

   Answer: $O(d^3)$

7. How does the answer to the previous question change if A has only z non-zeroes? (You can assume $z \geq d$)

   Answer: $O(d^3)$, it does not change as the iteration for the matrix multiplication is the same. Unless if the non-zeroes positions are known

4.2 Common Runtimes 5 / 7

- **- 1 pts** 1 incorrect
- **- 1 pts** 2 incorrect
- **- 1 pts** 3 incorrect
- **- 1 pts** 4 incorrect
- **- 1 pts** 5 incorrect
- ✓ **- 1 pts 6 incorrect**
- ✓ **- 1 pts 7 incorrect**
- **- 0 pts** All correct

4.3 Running times of code

Included in `a1.zip` is file named `bigO.jl`, which defines several functions that take an integer argument n. For each function, state the running time as a function of n, using big-O notation.

Answer: func1: $O(n)$
func2:$O(n)$
func3:$O(1)$
func4:$O(n^2)$
func5:$O(n^2)$

4.3 Running times of code 3 / 4

- **- 1 pts** 1 incorrect
- **- 1 pts** 2 incorrect
- **- 1 pts** 3 incorrect
- **- 1 pts** 4 incorrect
- ✓ **- 1 pts** **5 incorrect**
- **- 0 pts** Click here to replace this description.

5 Summary Statistics and Data Visualization

The file *a1.zip* contains estimates of the influenza-like illness percentage over 52 weeks on 2005-06 by Google Flu Trends in a comma-separated values (CSV) file. You can open this with Excel or other spreadsheet programs; the first row gives the abbreviation of the region names for each column, and each row gives the estimate for a week. After you change to the a0 directory, you can load this data in Julia using:

```
using DelimitedFiles
dataTable = readdlm("fluTrends.csv",',',')
```

This creates an two-dimensional array of type "Any" populated with all the information in the CSV file.

5.1 Summary Statistics

Report the following statistics: the minimum, maximum, mean, median, and mode of all values across the dataset. In light of the above, is the mode a reliable estimate of the most "common" value? Describe another way we could give a meaningful "mode" measurement for this (continuous) data.

Hint: Since the first row of the CSV file is just the names of the columns, we can create a matrix X containing the data stored as real numbers using:

```
X = real(dataTable[2:end,:])
```

You can make Julia display the matrix X using

```
@show X
```

The *show* macro can be used to display the result of any expression, like showing the tenth row of X:

```
@show X[10,:]
```

Note that this can be run inside functions, so it's helpful for debugging.

Julia has a mean and median function available, if you inxclude the Statistics package. This package does not have a mode command, so I've included one in 'misc.jl'.

Answer: minimum(X) = 0.352
maximum(X) = 4.862
mean(X) = 1.324625
median(X) = 1.1589999999999998
mode(X) = 0.77
The mode might not be the most reliable estimate of the most "common" value, as the result is continuous, the most frequent may not truly represent the most "common". A good estimate would be the mean or median instead, to show the most "average" value.

5.1 Summary Statistics **4 / 5**

- **- 0 pts** Correct
- ✓ **- 1 pts** **Incorrect explanation for better measure of the mode isn't reliable, or missing explanation**
- **- 1 pts** Computed statistics separately for each dataset
- **- 1 pts** Missing max/min values
- **- 5 pts** Incorrect or no answer

5.2 Data Visualization

Consider the figure below. The figure contains the following plots, in a shuffled order:

1. A histogram showing the distribution of all values in the matrix X.
2. A boxplot grouping data by weeks, showing the distribution across regions for each week.
3. A scatterplot between the two regions with highest correlation.
4. A single histogram showing the distribution of *each* column in X.
5. A scatterplot between the two regions with lowest correlation.
6. A plot containing the weeks on the x-axis and the percentages for each region on the y-axis.

Match the plots (labeled A-F) with the descriptions above (labeled 1-6), with an extremely brief (a few words is fine) explanation for each decision.

Answer: Apologies for missing the plots, it was not included in the tex file.
1: Plot D
2: Plot B
3: Plot F
4: Plot C
5: Plot E
6: Plot A

Hint: you can generate similar plots by adding the Plots package. To add this package and make a plot, run the following from the Julia REPL:

```
using Pkg # Loads the package manager
Pkg.add("Plots") # Only needs to be done once  (installs a Julia-callable Python build)
using Plots # Do this once per session
plot(1:52,X[:,1]) # Plot the first row
```

To generate similar-looking plots you can use the functions 'plot', 'scatter', 'histogram', and 'boxplot' (which is from the StatsPlot package).

5.2 Data Visualization **3 / 6**

- **- 0 pts** Correct
- ✓ **- 1 pts** **No explanations**
- ✓ **- 2 pts** **2 incorrect matches**
- **- 1 pts** 1 incorrect match
- **- 3 pts** 3 incorrect matches
- **- 6 pts** Incorrect or missing answer

5.3 Decision Surfaces

Consider the figure below, which plots a set of two-dimensional training examples and the decision surface produced by a "neural network" classifier (a model we'll see later in the course). How many training examples has the neural network mis-classified? (This figure is best viewed in colour.)

Answer: 17, they are the circles in the green region and the crosses in the blue region combined

5.3 Decision Surfaces 1 / 1

✓ **- 0 pts** **Correct**

- 1 pts Missing answer or incorrect

6 Decision Trees

If you run the file *example_decisionStump.jl*, it will load a dataset containing longtitude and latitude data for 400 cities in the US, along with a class label indicating whether they were a "red" state or a "blue" state in the 2012 election.[1] Specifically, the first column of the variable X contains the longitude and the second variable contains the latitude, while the variable y is set to 1 for blue states and 2 for red states. After it loads the data, it plots the data and then fits two simple classifiers: a classifier that always predicts the most common label (1 in this case) and a decision stump that discretizes the features (by rounding to the nearest integer) and then finds the best equality-based rule (i.e., check if a feature is *equal* to some value). It reports the training error with these two classifiers, then plots the decision areas made by the decision stump. The plot should look like this:

Note that these functions use the "JLD" package for loading the data and the "Plots" package to do the plotting. You can install these packages using:

```
using Pkg
Pkg.add("JLD")
Pkg.add("Plots")
```

6.1 Equality vs. Inequality Splitting Rules

In class we discussed splitting rules based on inequalities rather than equalities. Is there a type of feature where it makes sense to use an equality-based splitting rule?

Answer: Features that have binary outcomes, such as True and False would make sense for a equality-based splitting rule. This would be straight forward and does not require any inequality signs.

[1] The cities data was sampled from `http://simplemaps.com/static/demos/resources/us-cities/cities.csv`. The election information was collected from Wikipedia.

6.1 Equality vs. Inequality Splitting Rules 1 / 1

- ✓ **- 0 pts** Correct
- **- 1 pts** Incorrect/blank
- **- 0.5 pts** Partial answer

6.2 Decision Stump Implementation

The file *decisionstump.jl* contains a function that finds the best decision stump using the equality rule ("decisionStumpEquality"), and then returns a function that can apply this decision stump to new data. Instead of discretizing the data and using a rule based on testing an equality for a single feature, we want to check whether a feature is above a threshold and split the data accordingly (this is the more sane approach, which we discussed in class). Add a new function "decisionStump" to *decision_stump.jl* that finds the best inequality-based rule, and report the updated error you obtain by using inequalities instead of discretizing and testing equality.

Answer: Error with inequality-rule decision stump: 0.25

Hint: you may want to start by copy/pasting the contents of of the "decisionStumpEquality" function and then make modifications from there. Note that you should remove the calls to the "round" function for the inequality case. Make sure that you maintain the same input/output format in your function, since otherwise subsequent questions will not work (it should produce a plot that divides the US into a northern blue and a southern red area). If you are new to Julia, you may also want to look at*majorityPredictor.jl* to get an idea of the syntax in a simpler case.

```julia
function decisionStump(X,y)
    # Fits a decision stump based on inequality

    # Get the size of the data matrix
    (n,d) = size(X)

    # Initialize the "best rule" with the baseline rule (no split)
    y_mode = mode(y)
    minError = sum(y .!= y_mode);
    splitVariable = [];
    splitValue = [];
    splitYes = y_mode;
    splitNo = [];

    # Search for the best rule
    # (Uses O(n^2d) approach to keep code simple)
    yhat = zeros(n)
    for j in 1:d
        # Try unique values of column as split values
        for val in unique(X[:,j])

            # Test whether each object satisfies inequality
            yes = X[:,j] .>= val

            # Find correct label on both sides of split
            y_yes = mode(y[yes])
            y_no = mode(y[.!yes])

            # Make predictions
            yhat[yes] .= y_yes
            yhat[.!yes] .= y_no

            # Compute error
            trainError = sum(yhat .!= y)

            # Update best rule
            if trainError < minError
                minError = trainError
                splitVariable = j
                splitValue = val
                splitYes = y_yes
                splitNo = y_no
            end
        end
    end

    # Now that we have the best rule,
    # let's build our splitting function
    function split(Xhat)
        (t,d) = size(Xhat)
        if isempty(splitVariable)
            return fill(true,t)
        else
            return (Xhat[:,splitVariable] .>= splitValue)
        end
    end

    # Now that we have the best rule,
    # let's build our predict function
    function predict(Xhat)
        (t,d) = size(Xhat)
        yes = split(Xhat)
        yhat = fill(splitYes,t)
        if any(.!yes)
            yhat[.!yes] .= splitNo
        end
        return yhat
    end

    return StumpModel(predict,split,isempty(splitNo))
end
```

6.2 Decision Stump Implementation 2 / 2

✓ **- 0 pts** **Correct**

- 1 pts Incorrect/Missing error

- 2 pts Incorrect/Blank

6.3 Decision Tree to Program

Once your *decisionStump* function is finished, the script *example_decisionTree* will be able to fit a decision tree of depth 2 to the same dataset (which results in a lower training error). Look at how the decision tree is stored and how the (recursive) *predict* function works. Using the same splits as the fitted depth-2 decision tree, write out what an alternate version of the predict function would be for classifying one training example as a simple program using if/else statements (as in the first slide of L3 that has the title "Decision Trees").

Hint: you can use use the "@show" macro to print the values of various expressions during the execution of a .jl file.

Answer:

```
function depth2DecisionTree(X, y)
    # Split from the top
    (n,d) = size(X)

    # Top level splitting
    splitModel = decisionStump(X,y)

    # yes represents where the model will be split at the top level
    yes = splitModel.split(X)

    # Bottom level splitting
    yesModel = decisionStump(X[yes,:],y[yes])
    noModel = decisionStump(X[.!yes,:],y[.!yes])

    # Combining both top and bottom level to return a GenericModel
    function predict(Xhat)
        (t,d) = size(Xhat)
        yhat = zeros(t)

        yes = splitModel.split(Xhat)

        yhat[yes] = yesModel.predict(Xhat[yes,:])
        yhat[.!yes] = noModel.predict(Xhat[.!yes,:])
        return yhat
    end

    return GenericModel(predict)
end
```

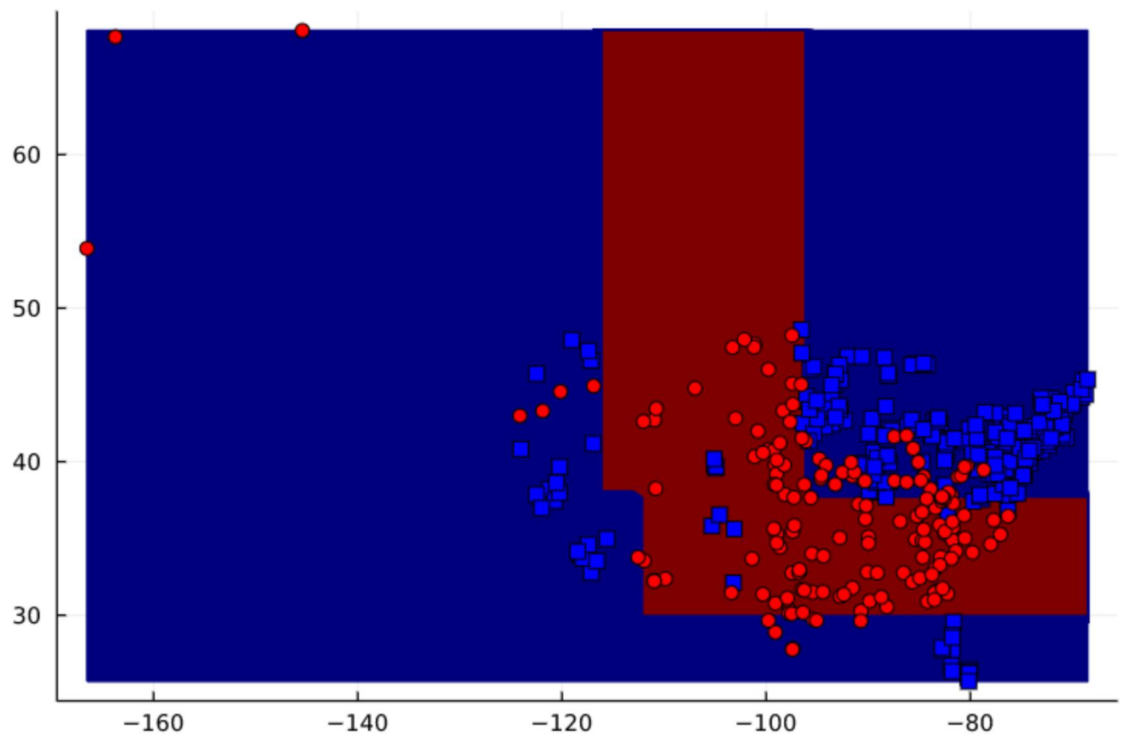
6.3 Decision Tree to Program **0 / 2**

- **- 0 pts** Correct
- **- 0.5 pts** Incorrect/Missing 37.669
- **- 0.5 pts** Incorrect/Missing -96.0901
- **- 0.5 pts** Incorrect/Missing -115.5776
- **- 0.5 pts** Incorrect Order
- ✓ **- 2 pts Incorrect/Blank**
- **- 1 pts** Not using the hard coded values

6.4 Constructing Decision Trees

Report the training error generated by fitting a decision tree of depth 3, and hand in the corresponding decision surface plot.

Answer: Error with depth-3 decision tree: 0.133



HAVE YOU DOUBLE CHECKED THAT YOU'RE FOLLOWING ALL THE ASSIGNMENT SUBMISSION INSTRUCTIONS POSTED ON PIAZZA???

6.4 Constructing Decision Trees 1 / 1

✓ **- 0 pts** **Correct**

- 1 pts Incorrect/Blank

- 0.5 pts Incorrect/missing error

- 0.5 pts Incorrect/missing graph

CPSC 340 Assignment 1 (due Friday September 16 at 11:55pm)

Commentary on Assignment 1: CPSC 340 is tough because it combines knowledge and skills across several disciplines. To succeed in the course, you will need to know or very quickly get up to speed on:

- Math to the level of the course prerequisites: linear algebra, multivariable calculus, some probability.
- Basic Julia programming, and the ability to translate from math to programming and back.
- Statistics, algorithms, and data structures to the level of the course prerequisites.
- Some basic LaTeX skills so that you can typeset equations and submit your assignments.

The purpose of this assignment is to make sure you are prepared for this course. We anticipate that each of you will have different strengths and weaknesses, so don't be worried if you struggle with *some* aspects of the assignment. But if you find this assignment to be very difficult overall, that is a sign that you may not be prepared to take CPSC 340 at this time. Future assignments will be more difficult than this one (and probably around the same length).

Questions 1-4 are on review material, that we expect you to know coming into the course. The rest is related to the first few lectures.

IMPORTANT!!!!! Before proceeding, please carefully read the homework instructions posted on Piazza.

You may receive a 50% deduction on the assignment if you don't follow these instructions.

We use blue to highlight the deliverables that you must answer/do/submit with the assignment.

You may also want to read the answers to this Quora question as motivation:
`https://www.quora.com/Why-should-one-learn-machine-learning-from-scratch-rather-than-just-learning-to-use-the-available-libraries`

Basic Information

1. Name: Foo Chuan Shao
2. Student ID: 30321277