STAT 341: Assignment 1

DUE: Friday, May 24, 2024 by 5:00pm EST

NOTES

Your assignment must be submitted by the due date listed at the top of this document, and it must be submitted electronically in .pdf format via Crowdmark. This means that your responses for different questions should begin on separate pages of your .pdf file. Note that your .pdf solution file must have been generated by R Markdown. Additionally:

- For mathematical questions: your solutions must be produced by LaTeX (from within R Markdown). Neither screenshots nor scanned/photographed handwritten solutions will be accepted – these will receive zero points.
- For computational questions: R code should always be included in your solution (via code chunks in R Markdown). If code is required and you provide none, you will receive zero points.
- For interpretation questions: plain text (within R Markdown) is required. Text responses embedded as comments within code chunks will not be accepted.

Organization and comprehensibility is part of a full solution. Consequently, points will be deducted for solutions that are not organized and incomprehensible. Furthermore, if you submit your assignment to Crowdmark, but you do so incorrectly in any way (e.g., you upload your Question 2 solution in the Question 1 box), you will receive a 5% deduction (i.e., 5% of the assignment's point total will be deducted from your point total).

QUESTION 1: Basic R Calculations [12 points]

R is an excellent replacement for a graphing calculator, or even Wolfram Alpha. Solve the following expressions using R, and provide the code you used to do it.

(a) [1 point]

$$5^7$$

(b) [1 point]

$$\log_2(21)$$

(c) [1 point]

$$\sum_{x=1}^{1000} \frac{\cos(x+\pi)}{x^2}$$

(d) [1 point]

$$21 \mod 2$$

In the first lecture we talked about how `apply` functions can be used as an alternative to using `for` loops in R. Implement the tasks in parts (e) and (f) using an appropriate call to `apply` (instead of a `for` loop) with the following matrix `A`.

```
A <- matrix(data = c(5, 6, 6, 3, 4, 6, 7, 10, 6, 4, 4, 5, 6, 2, 7, 9),
    nrow = 4, byrow = FALSE)
```

(e) [2 points] Calculate the median of the values in each row of `A`, and the mean of the values in each column of `A`.

(f) [2 points] Determine how many entries in each row of `A` are divisible by 5. Note: you may find it helpful to write *your own* function to pass into the `FUN` input.

Later in the course, you will use more sophisticated techniques to minimize functions $f(x)$ or find solutions to the equation $f(x) = 0$. In part (g), solve this type of equation by naively evaluating $f(x)$ at various x values. Get the `x` values using the `seq(..., by=dx)` function. Notes:

- Using an extremely small `dx` will use a lot of computational resources, so press escape if a computation is taking too long and try something more moderate.
- You can check your work with calculus. Your answers should be within 0.001 of the correct answers.

(g) [2 points] Approximate the value of x that minimizes $f(x) = x^4 + x^3 + 2x^2 + 3x + 4$ over $-2 \leq x \leq 2$. Return the $(x, f(x))$ combination for this approximate solution.

(h) [2 points] Write (and show the code for) a function called `nearest_neighbour()` which takes inputs `v` (a vector of numbers to search) and `x` (the number whose nearest neighbour in `v` you want to find). This function should output the position(s) of the element(s) in `v` closest to `x`. Call this function as shown below and provide the output.

```
nearest_neighbour(v = c(7, 10, 5, 10, 14, 2, 11, 8, 13, 8), x = 9)
```

QUESTION 2: Investigating the Proportion of Non-Negatives [10 points]

Suppose we're studying a population $\mathcal{P}$ in which some numeric variate y is measured on every unit $u \in \mathcal{P}$. The population may then be described as $\mathcal{P} = \{y_1, y_2, \ldots, y_N\}$. Suppose further that interest lies in determining what proportion of the units in $\mathcal{P}$ have non-negative variate values. The following proportion attribute measures exactly this:

$$a(\mathcal{P}) = \frac{1}{N} \sum_{u \in \mathcal{P}} I_{[0,\infty)}(y_u)$$

where $I_{[0,\infty)}(y_u) = 1$ if $y_u \geq 0$, and 0 otherwise.

(a) [2 points] Derive whether the proportion attribute $a(\mathcal{P})$ is location invariant, location equivariant, or neither.

(b) [2 points] Derive whether the proportion attribute $a(\mathcal{P})$ is scale invariant, scale equivariant, or neither.

(c) [2 points] Derive whether the proportion attribute $a(\mathcal{P})$ is replication invariant, replication equivariant, or neither.

(d) [2 points] Derive the sensitivity curve $SC(y; a(\mathcal{P}))$ for the proportion attribute above. For this question you may assume the population $\mathcal{P} = \{y_1, y_2, \ldots, y_{N-1}\}$ has $N-1$ elements so that the population with y injected into it has N elements: $\{y_1, y_2, \ldots, y_{N-1}, y\}$.

(e) [2 points] For the population `pop = rnorm(1000)` (with seed set using `set.seed(341)`), plot the sensitivity curve for the proportion $a(\mathcal{P})$ for $y \in [-5, 5]$. Note that it may be helpful to write a simple function to calculate this proportion given a vector of numbers. This may then be used together with the `sc()` function from class. Make sure your plot has an informative title and axis labels. Hint: you may use this plot to ensure your derivation in (d) is correct.

QUESTION 3: k-Nearest Neighbour Classifier [10 points]

The *k-nearest neighbour* algorithm (often abreviated as "kNN") is perhaps the simplest machine learning method that exists. In this question you will write a function that implements a basic, 1-dimensional version of kNN.

Description of 1D Classification

Suppose we have a response variate y and an explanatory variate x, and for a unit $u \in \mathcal{P}$ we believe knowing x_u will help us predict y_u. For simplicity, here we'll assume the response variate y is binary, and $y_u = 1$ implies unit u belongs to "class 1" and $y_u = 0$ implies unit u belongs to "class 2". The problem of prediction in this setting is referred to as *classification*; on the basis of x_u, we seek to classify unit u as "class 1" ($y_u = 1$) or "class 2" ($y_u = 0$).

Description of 1D kNN

To make this determination, the kNN method simply finds the k units in $\mathcal{P}$ whose x values are closest to x_u. It then predicts $\hat{y}_u = 1$ if 50% or more of the k-nearest neighbours are in class 1, and it therefore predicts $\hat{y}_u = 0$ if less than 50% of the k-nearest neighbours are in class 1 (i.e., when more than 50% of the k-nearest neighbours belong to class 2). In this way, unit u is given the same classification as the majority of its k-nearest neighbours.

Your Task

Write a function called `knn()` that performs this classification for a population $\mathcal{P} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$. Your function should take only the following inputs:

- `x`: a vector of explanatory variate values for each of the units in $\mathcal{P}$.
- `y`: a vector of response variate values for each of the units in $\mathcal{P}$.
- `k`: the number of neighbours to consider when performing the majority-vote classification rule.

The function should output a vector of 1's and 0' representing the predicted classes for each unit $u \in \mathcal{P}$.

Apply your function to the data found in `q3data.csv`, and set `k=3`. Note that a correct implementation of kNN in this case yields classifications with the following *confusion matrix.* This matrix counts:

- the number of times 1's were correctly classified as 1's (true positives, TP)
- the number of times 1's were incorrectly classified as 0's (false negatives, FN)
- the number of times 0's were correctly classified as 0's (true negatives, TN)
- the number of times 0's were incorrectly classified as 1's (false positives, FP)

		Prediction	
		$\hat{y}_u = 1$	$\hat{y}_u = 0$
Truth	$y_u = 1$	3	3
	$y_u = 0$	1	14

QUESTION 4: Kendrick vs. Drake [20 points]

As sometimes happens in hip-hop, artists feud and attack one another lyrically through so-called diss tracks. Recently, a longstanding feud between Toronto rapper Drake and Compton, California's Kendrick Lamar has boiled over culminating in a series of diss tracks exchanged over the past few weeks. Kendrick's most recent song "Not Like Us" broke the record of the most single-day streams for a hip-hop song, a record previously held by Drake. This is being called the defining hip-hop beef of the 21st century. It is therefore timely, and perhaps interesting, to explore the relative popularity of these artists. The file `kendrick_v_drake.csv` contains information on $N = 269$ songs (128 are Kendrick's and 141 are Drake's); the variates recorded for each song are summarized in the table below. Note this information was extracted from YouTube Music.

Variate	Description
`Title`	The song's title.
`Artist`	The song's artist(s).
`Plays`	The number of times the song has been played (on YouTube Music as of May 10, 2024).
`Album`	The album the song appeared on.
`Duration`	The duration of the song, recorded as `mm:ss`.

In the questions that follow, you will (via multiple methods and modalities) compare the popularity of Kendrick vs. Drake based on the number of plays of their songs (on YouTube Music).

(a) [4 points] Read the data into R and calculate the `summary()` of `Plays` for each artist separately. Additionally, construct side-by-side boxplots of `Plays/1000000` by artist. Ensure your plot has an informative title and axis labels.

(b) [4 points] Construct an individuals scatter plot of `Plays/1000000` with the individual values jittered. Ensure the points are different colours depending on whether the song is Drake's or Kendrick's, and jittered sufficiently to distinguish them. Also ensure your plot has an informative title and axis labels, and a legend to distinguish the two different colours of points.

(c) [2 points] Summarize the informal insights you draw from the visualizations and numeric summaries constructed in parts (a) and (b).

(d) [3 points] If one artist is drastically more popular than the other, one would expect that `Plays` serves as a proxy for popularity, and that knowledge of the number of `Plays` a song has would be enough to predict its artist. You will investigate this supposition by using kNN to classify songs as "Kendrick" ($y = 1$) or "Drake" ($y = 0$) based on $x =$ `Plays/1000000`. Using your `knn()` function from Question 3 with `k=5`, predict the artist of each song based on how many millions of times it's been played. Provide the confusion matrix associated with this classification. Structure this matrix like the one shown below, but with TP, TN, FP, FN replaced with their actual counts.

		Prediction	
		$\hat{y}_u = 1$	$\hat{y}_u = 0$
Truth	$y_u = 1$	TP	FN
	$y_u = 0$	FP	TN

(e) [4 points] The quality of the classification depends on the number of neighbours k specified in the algorithm. In this question you will consider `k=2:200`, and identify the value of k that appears to maximize the *accuracy* of the classification. Note that

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

is the proportion of songs correctly classified. In particular, run your `knn()` algorithm for `k=2:200`, each time calculating (and saving) the corresponding accuracy value. Then construct a plot of *accuracy* vs. k.

(f) [3 points] Based on the plot from part (e), comment on:

- which values of k appear optimal;
- the impact of considering too many and too few neighbours (i.e., very large k and very small k);
- whether there seems to be a noticeable difference in song plays between the two artists.

.