CP421 Assignment 1

Nathaniel Carr

Note: In some cases, I have neglected to write out long summations, since it would be too wide for the page and make it hard to see what I'm actually doing. I've instead used a sigma summation notation. An example of this is found in 1.a, where I could have typed all 27 data points with a + between them, but it would exceed the page width and formatting it on separate lines would be confusing.

1.a)
Let A be the array of ages in sorted order.

The mean is:

$$\frac{\sum_{i=1}^{27} A[i]}{27} = \frac{809}{27} \approx 30$$

The median is the 13th item (mid-point): 25.

1.b)
This data is bimodal, since it has 2 numbers that appear at the highest frequency: 25 and 35.

1.c)
The midrange of this data is:

$$\frac{70+13}{2} = 41.5$$

1.d)
Since the median is found at the 14th item, we look at the median of the 13 items to the left and 13 items to the right of the 14th item to find Q1 and Q3, respectively. Thus,

$$Q1 = 20$$
$$Q3 = 35$$

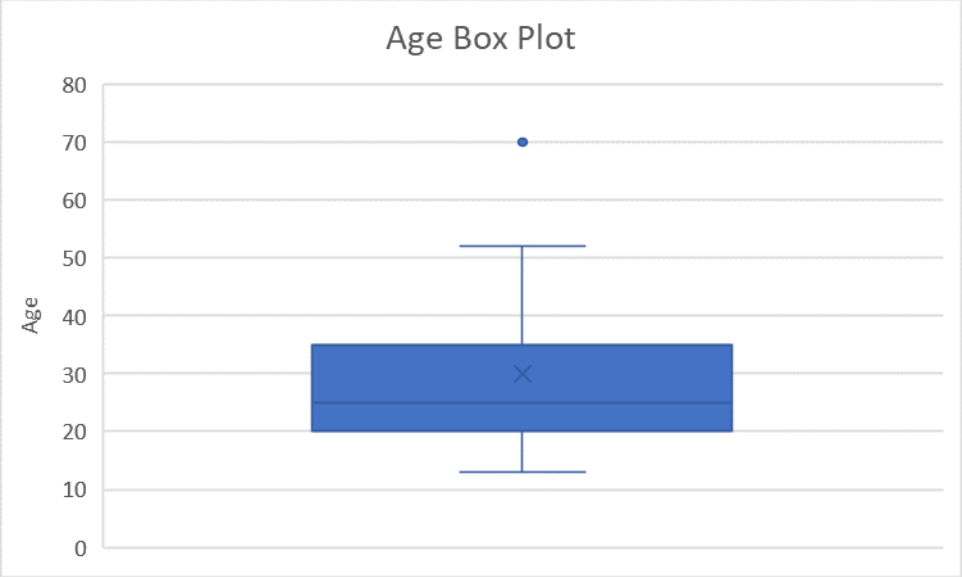
1.e)

$$Minimum = 13$$
$$Q1 = 20$$
$$Median = 25$$
$$Q3 = 35$$
$$Maximum = 70$$

1.f)
Note that 70 is an outlier because:

$$IQR = 35 - 20 = 15$$

$1.5(15) + 35 = 57.5$
$70 > 57.5$



2.a)
Using Euclidean distance:
$d(x_1, x) = \sqrt{(1.5 - 1.4)^2 + (1.7 - 1.6)^2} \approx 0.1414$
$d(x_2, x) = \sqrt{(2.0 - 1.4)^2 + (1.9 - 1.6)^2} \approx 0.6708$
$d(x_3, x) = \sqrt{(1.6 - 1.4)^2 + (1.8 - 1.6)^2} \approx 0.2828$
$d(x_4, x) = \sqrt{(1.2 - 1.4)^2 + (1.5 - 1.6)^2} \approx 0.2236$
$d(x_5, x) = \sqrt{(1.5 - 1.4)^2 + (1.0 - 1.6)^2} \approx 0.6083$

Since $similarity = 1 - dissimilarity$, the ranking of this data by similarity with the query using Euclidean distance (in ascending order of similarity): x_1, x_4, x_3, x_5, x_2.

Using cosine similarity:
$\|x_1\| = \sqrt{1.5^2 + 1.7^2} = \sqrt{5.14}$
$\|x_2\| = \sqrt{2.0^2 + 1.9^2} = \sqrt{7.61}$
$\|x_3\| = \sqrt{1.6^2 + 1.8^2} = \sqrt{5.80}$
$\|x_4\| = \sqrt{1.2^2 + 1.5^2} = \sqrt{3.69}$
$\|x_5\| = \sqrt{1.5^2 + 1.0^2} = \sqrt{3.25}$
$\|x\| = \sqrt{1.4^2 + 1.6^2} = \sqrt{4.52}$

$\cos(x_1, x) = \frac{1.5(1.4)+1.7(1.6)}{\sqrt{5.14}(\sqrt{4.52})} \approx 0.99999$
$\cos(x_2, x) = \frac{2.0(1.4)+1.9(1.6)}{\sqrt{7.61}(\sqrt{4.52})} \approx 0.9958$
$\cos(x_3, x) = \frac{1.6(1.4)+1.8(1.6)}{\sqrt{5.80}(\sqrt{4.52})} \approx 0.99997$

$\cos(x_4, x) = \frac{1.2(1.4)+1.5(1.6)}{\sqrt{3.69}(\sqrt{4.52})} \approx 0.9990$

$\cos(x_5, x) = \frac{1.5(1.4)+1.0(1.6)}{\sqrt{3.25}(\sqrt{4.52})} \approx 0.9654$

Ranking by similarity with the query using cosine similarity: x_1, x_3, x_4, x_2, x_5.

2.b)

For all x_i:

$norm_{old}{}^2 = A_1{}^2 + A_2{}^2$

$1 = \frac{A_1{}^2}{norm_{old}{}^2} + \frac{A_2{}^2}{norm_{old}{}^2}$

$1 = \left(\frac{A_1}{norm_{old}}\right)^2 + \left(\frac{A_2}{norm_{old}}\right)^2$

$1 = \left(\frac{A_1}{\sqrt{A_1{}^2+A_2{}^2}}\right)^2 + \left(\frac{A_2}{\sqrt{A_1{}^2+A_2{}^2}}\right)^2$

Consequently, for all A_j of x_i:

$A_{j_{new}} = \frac{A_j}{\sqrt{A_1{}^2+A_2{}^2}}$

The normalized data set is:

	A_1	A_2
x	0.6585	0.7526
x_1	0.6616	0.7498
x_2	0.7250	0.6887
x_3	0.6644	0.7474
x_4	0.6247	0.7809
x_5	0.8321	0.5547

Repeating the Euclidean distance calculations:

$d(x_1, x) = \sqrt{(0.6616 - 0.6585)^2 + (0.7498 - 0.7526)^2} \approx 0.0042$

$d(x_2, x) = \sqrt{(0.7250 - 0.6585)^2 + (0.6887 - 0.7526)^2} \approx 0.0922$

$d(x_3, x) = \sqrt{(0.6644 - 0.6585)^2 + (0.7474 - 0.7526)^2} \approx 0.0079$

$d(x_4, x) = \sqrt{(0.6247 - 0.6585)^2 + (0.7809 - 0.7526)^2} \approx 0.0441$

$d(x_5, x) = \sqrt{(0.8321 - 0.6585)^2 + (0.5547 - 0.7526)^2} \approx 0.2633$

Ranking by similarity with the query using Euclidean distance: x_1, x_3, x_4, x_2, x_5.

3.a)

The number of baskets some item, x, appears in can be calculated by $\frac{100}{x}$.

The highest value of x for which $\frac{100}{x}$ is greater than or equal to the support threshold of 5 is 20, so every number below 21 is frequent and all others are infrequent.

The following set of items are frequent:
$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20\}$

3.b)
5 and 7 are found together in the following baskets: 35, 70.
2, 5, and 7 are found together in the following baskets: 70.

$\frac{1}{2} = 0.5$

The confidence of {5, 7} → {2} is 0.5.

2, 3, and 4 are found together in the following baskets: 12, 24, 36, 48, 60, 72, 84, 96.
2, 3, 4, and 5 are found together in the following baskets: 60.

$\frac{1}{8} = 0.125$

The confidence of {2, 3, 4} → {5} is 0.125.

4)
The source code (as well as the log.txt, which contains the results gathered by running the program), is included in the zip file as miner.py. The code is well-commented, should any of the following description be unclear. The itertools built-in Python library is used to calculate some combinations of sets. The time library was used to check runtime.

miner.py first collects all lines from the browsing.txt file, then breaks each line into a list of items. The baskets in the file are searched one-by-one, adding new items to a Python dictionary object and incrementing the frequency of each found item every time it is seen.

These items are iterated over to prune out any items that appear fewer than 100 times.

Each combination of the frequent 1-itemsets is created and validated according to the k-1 X k-1 candidate generation rule before being added to a new dictionary of candidate 2-itemsets.

The baskets in the file are searched one-by-one (making every possible 2-combination in every basket), and the frequencies of any candidate 2-itemsets found are increased appropriately.

The candidate 2-itemsets are pruned.

Association rules are mined for 2-itemsets by generating every possible partition of each frequent 2-itemset and searching the frequent 1-itemsets to determine the frequency of the left side of each rule. These rules are added to a list.

Each combination of frequent 2-itemsets is created and validated according to the k-1 X k-1 candidate generation rule before being added to a new dictionary of candidate 3-itemsets.

The baskets in the file are searched one-by-one (making every possible 3-combination in every basket), and the frequencies of any candidate 3-itemsets found are increased appropriately.

The candidate 3-itemsets are pruned.

Association rules are mined for 3-itemsets by generating every possible partition of each frequent 3-itemset and searching the frequent 1-itemsets and 2-itemsets to determine the frequency of the left side of each rule. These rules are added to a list.

Finally, the discovered rules and frequent itemsets are logged in a file. Rules are sorted in descending order of confidence, then ascending order lexicographically.

4.a)
1668 2-itemset association rules were calculated from 1334 frequent 2-itemsets, and the following 5 rules had the highest confidence:

1.{DAI93865} -> {FRO40251} - frequency: 208, confidence = 1.0
2. {GRO85051} -> {FRO40251} - frequency: 1213, confidence = 0.999176276771005
3. {GRO38636} -> {FRO40251} - frequency: 106, confidence = 0.9906542056074766
4. {ELE12951} -> {FRO40251} - frequency: 105, confidence = 0.9905660377358491
5. {DAI88079} -> {FRO40251} - frequency: 446, confidence = 0.9867256637168141

4.b)
1398 3-itemset association rules were calculated from 233 frequent 3-itemsets, and the following 5 rules had the highest confidence (there are 15 3-itemset association rules with a 1.0 confidence, but these 5 are sorted lexicographically by left side, as well):

1. {DAI23334, ELE92920} -> {DAI62779} - frequency: 143, confidence = 1.0
2. {DAI31081, GRO85051} -> {FRO40251} - frequency: 102, confidence = 1.0
3. {DAI55911, GRO85051} -> {FRO40251} - frequency: 133, confidence = 1.0
4. {DAI62779, DAI88079} -> {FRO40251} - frequency: 117, confidence = 1.0
5. {DAI75645, GRO85051} -> {FRO40251} - frequency: 395, confidence = 1.0