UNIVERSITY OF MALAYA

Group Project Report

WIA2005 - Algorithm Design & Analysis

Semester 2 2021

Tutorial Group: 1

Group Number: 3

Lecturer: Dr. Asmiza binti Abdul Sani

Group Members:

No	Name	Matrics Number
1	Raja Zareef Firdaus bin Raja Azman Nahar (Leader)	17207394
2	Muhammad Adi Safwan bin Mohd Hasmadi	17204337
3	Muhammad Haziq bin Hilmi	17206486
4	Thinaraj A/L A Muttiah	U2005319
5	Afifah binti Abdul Halim	17203106
6	Fatima Zahra binti Ahmad Ariff	17207077

Table of Content

Table of Content 2

Introduction 3

Problem 1 3

Problem 2 3

Problem 3 3

Description 4

PROBLEM 1 4

Step 1: Extracting the words from the articles 4

Step 2: Cleaning the data 4

Step 3: Sentiment Analysis 4

Step 4: Plotting 4

Sentiment Analysis Time Complexity 4

PROBLEM 2 5

Step 1: Finding distribution Center 5

Weiszfeld's algorithm Time complexity 6

Step 2: Finding the optimal route 6

Christofides Approach Time Complexity 6

PROBLEM 3 7

Problem 3 Time Complexity 8

The program code: source code and snapshots of input/output 9

Problem 1: 9

Problem 2: 10

Problem 3: 13

Conclusion 14

References 15

Introduction

Problem 1

In this problem, we are required to do sentiment analysis on 5 countries based on their local economy and social situation. We divided this problem into a few phases. In the first phase, we did the data collection where five of us will randomly choose a country and then find 5 articles on that country that are related to its local economy and social situation. Next, we move on to the brainstorming phase, where we find ways to approach the problem. During the brainstorming, we use Github and Microsoft Teams as our main discussion platforms. With the help of an online forum and a few Youtube videos, we manage to come to a solution that is to use Textblob to do the sentiment analysis. Upon finishing the program for this problem, we get into the final phase which is the testing. Here we test the program that we had with the articles we had collected before and also find the time complexity of the program created.

Problem 2

In this problem we are required to find a local distribution center for each of the countries that we chose before and also find the optimal delivery route for it. To solve this problem, we divided our teams into 2, one to brainstorm on subproblem 1 and another on subproblem 2. After a week of brainstorming, we move on to the discussion phase, where each team will present their findings. Team subproblem 1 manages to come up with a solution to find a center and have some ideas on how to find the optimal delivery. Team subproblem 2 presents some ideas on how to approach subproblem 2 and after some discussion, we agreed to use the Team subproblem 1 approach as it will be much easier to implement. Next, is the development phase. We developed our programs and did the testing. We also managed to get the time complexity for each of the subproblems.

Problem 3

In this problem, we are required to calculate the probability of a country that has a good local economic and social situation with the lowest optimal delivery. To solve this, we start with a brainstorming phase. We discuss how to find the probability for each country given that we have their sentiment analysis and cost of delivery. We managed to come up with a solution to tackle this problem and move to the next phase, program development. Here, we developed the program for this problem and did the testing. Upon solving the problem, we also come up with a summary in a ranking from the most recommended countries to the least recommended countries to have an expansion. The time complexity for this problem also is calculated.

Description

PROBLEM 1

Step 1: Extracting the words from the articles

Textify the websites, & then manually copy pasting only the necessary paragraphs into a notepad (txt file)

Step 2: Cleaning the data

Removed the punctuations, stopwords (counting them as well) & lowering all letters to lowercase. For stopwords, we combined nltk's stopwords library with the online stopwords from what Dr shared.

Step 3: Sentiment Analysis

Used TextBlob built-in library to do sentiment analysis. TextBlob is a Python (2 and 3) library for processing textual data. It provides a consistent API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, and more. Basically how TextBlob does the sentiment analysis is it takes an input called a blob (set of strings from our text files) then it compares each word within the blob to its internal library/dictionary or lexicon (similar to how a trie function). Each word in the library has its designated sentiment score (closer to 1 means more positive & closer to -1 means more negative). Every time it finds a match to the word, it totals up the sentiment scores, and then at the end, it will average the total scores with the total number of words within the blob, giving the final output which is the overall sentiment score of the blob in the range between -1 to 1.
The main default sentiment calculation is defined in _text.py.
The lexicon it refers to is in en-sentiment.xml

Step 4: Plotting

A bar chart is created to visualize the sentiment score of each country. The bar chart is chosen to represent this because the data is categorical & not continuous (i.e. the x-axis are the countries. In contrast, other charts like histograms are for continuous & numerical data in which the data can aggregate into bins (range groups), this creates a continuous gapless chart. In addition, a bar chart is perfect for showing the contrast between the countries, which aids in decision-making. Only the sentiment score of the countries are plotted as we believe that that metric is enough for proceeding analysis as the ratio between the positive & negative sentiments is taken into account in the overall sentiment score.

Sentiment Analysis Time Complexity

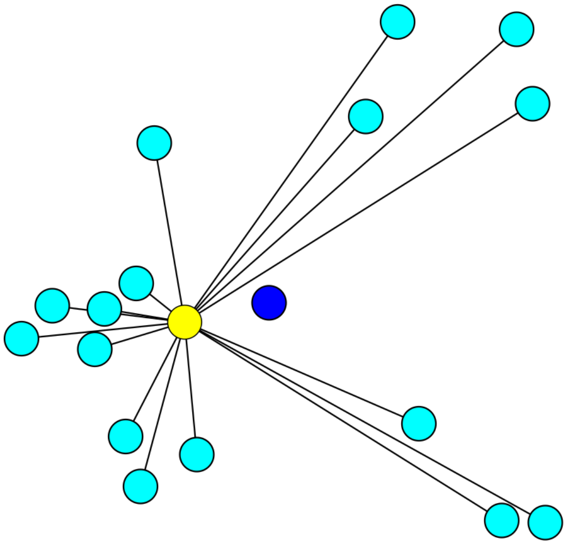
O(n); n is the no of words in the blob. To make it more efficient & faster, stop words & punctuations are removed as they do not contribute to the overall sentiment score.

PROBLEM 2

Step 1: Finding distribution Center

# Weiszfeld’s Algorithm

The distribution center is found using a geometric approach. The stores in a country are regarded as a series of points. In order to find the most suitable store to be the distribution center, that is, the store that is located in the middle of all of the other stores in the country, the geometric median must be found. In geometry, the geometric median of a discrete set of sample points in a Euclidean space is the point minimizing the sum of distances to the sample points.



[Diagram 1: A simple illustration of Problem 2 Step 1. The yellow circle is the geometric median in a Euclidean space.]

To find the geometric median, Weiszfeld's algorithm is used. Below is the formula of Weiszfeld’s algorithm.

$$y_{i+1} = \left(\sum_{j=1}^{m} \frac{x_j}{\|x_j - y_i\|} \right) \bigg/ \left(\sum_{j=1}^{m} \frac{1}{\|x_j - y_i\|} \right).$$

It is a form of an iteratively re-weighted sum of squares. This algorithm defines a set of weights that are inversely proportional to the distances from the current estimate to the sample points and creates a new estimate that is the weighted average of the sample according to these weights.

Weiszfeld's algorithm Time complexity

O(N*M/K) where N is the number of points, M is the dimension and K is the error. In this solution, M is set to 2 and the error is set to 0.0000000001 or $1X10^{-15}$.

Step 2: Finding the optimal route

# Christofides Approach

Due to the number of cities we have to traverse, we opted to use an approximation algorithm to solve this rather than choosing a greedy algorithm. This is because the scope of the problem is close to the Traveling Salesman Problem (TSP). The goal is not to find an exact solution (as it can require an immense computational power to solve), but to verify whether it falls within a certain proportion of the optimal solution, hence the name "approximation".

Created by Nicos Christofides in the late 1970s, it is a multistep algorithm that guarantees its solution to the TSP will be within 3/2 of the optimal solution.

The steps:

1. Create an undirected minimum spanning tree from the original graph, G
2. Create a set of nodes with odd degrees
3. Create a minimum perfect matching of the odd degrees we found.
4. Combine (3) & (1) to form an Eulerian Cycle.
5. Remove any excess visits to cities we have by creating a Hamiltonian path.

### Christofides Approach Time Complexity

Since the algorithm is multistep in nature, its running time and complexity varies based on the running time of its components. The upper bound on computations for Christofides' Algorithm is roughly **O(|V|^4),** significantly better than any of the exact solutions approaches.

PROBLEM 3

The probability of a country that has a good local economic and social situation with the lowest optimal delivery.

Step 1: Calculating the probability

First, we calculated the probability of a country with a good local economic and social situation and the probability of the country with the lowest optimal delivery. For instance, we set two variables as:

G = Good local economic and social situation

H = Lowest optimal delivery

The formula we used to find the probability of a country with a good local economic and social situation is:

P(G) = score / total score

Whereas to find the probability of a country with lowest optimal delivery is:

P(H) = 1 - (cost / total cost)

We subtract it from 1 because currently, we are trying to find out the lowest optimal delivery cost. The probability should be higher if the delivery cost is lesser. If we perform the calculation without subtracting from 1, the calculated probability will show that the country with a higher cost has a higher probability to be chosen, which fails to prove our question’s requirement. Therefore, we reversed the probability by subtracting from 1, which is proven logical.

Then, we have to find out the combined probability of both cases. Since both cases are independent from each other, we have to use the formula “P(A and B) = P(A) x P(B)” which in our case is:

P (G and H) = P(G) x P(H)

Step 2: Sorting the probability to find the countries’ ranking

Secondly, we have to sort the calculated probabilities in descending order. The countries with a higher probability will have a higher ranking. Shell sort is used to sort the country's ranking. This is because the probability we calculated will have long and unlimited decimal places. From the sorting algorithms we learned before, shell sort will be the best for sorting this type of value. But after the presentation, it is explained to us that bubble sort is a better option than shell sort as the data is small.

Problem 3 Time Complexity

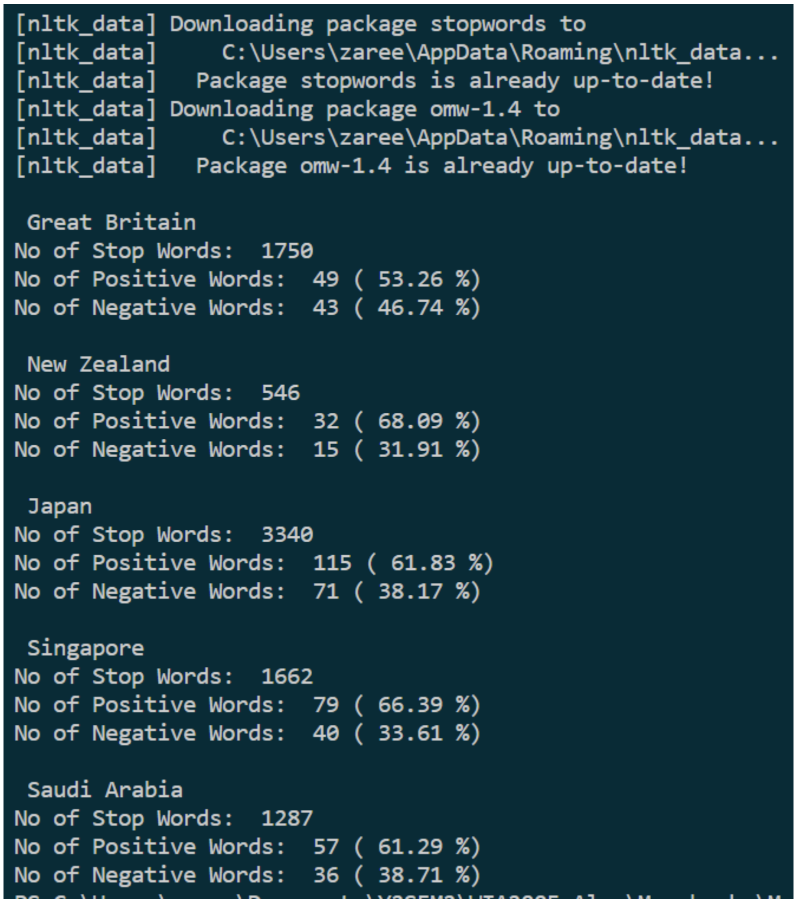
Probability:
The time complexity for calculating the probability will be T(n) = **O(n)**, where n is the number of countries.

Shell sort:
Best Case Complexity: **O(n log n)**
Average Case Complexity: slightly worse than O(n log n) to O(n^2), depending on the gap sequence
Worst Case Complexity: **O(n^2)**, where n is the number of countries.

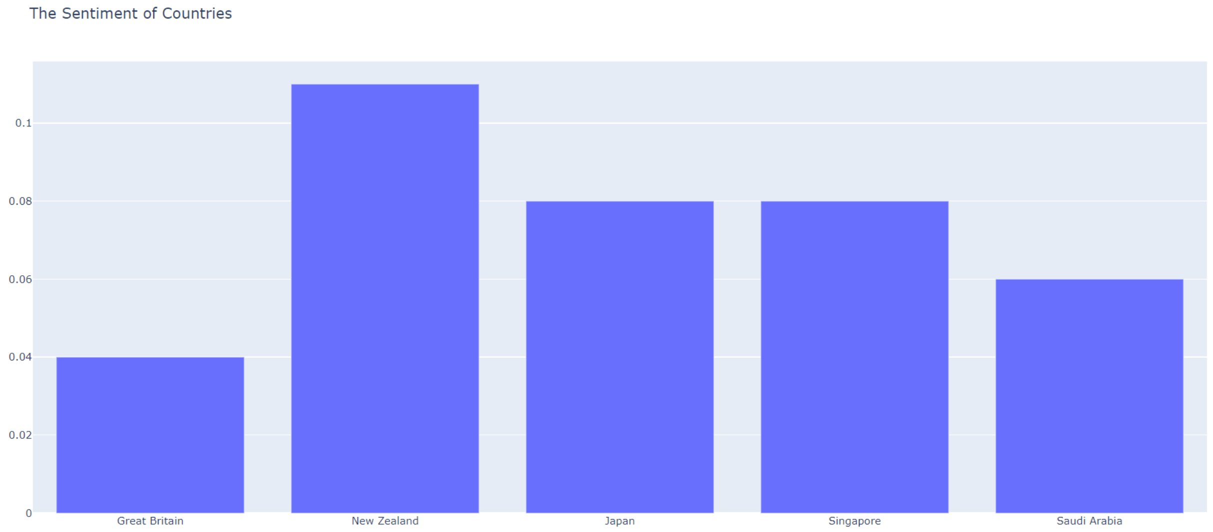
The program code: source code and snapshots of input/output

Source Codes: https://github.com/zareefrj/MoonBucks

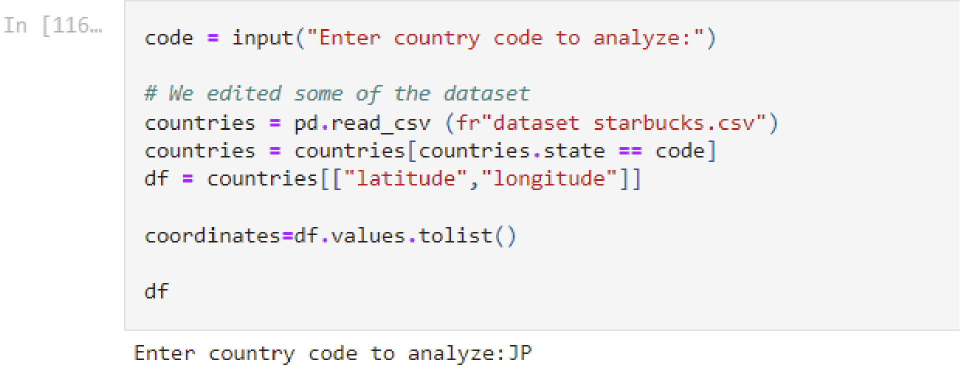
Problem 1:



```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\zaree\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package omw-1.4 to
[nltk_data]     C:\Users\zaree\AppData\Roaming\nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!

 Great Britain
No of Stop Words:  1750
No of Positive Words:  49 ( 53.26 %)
No of Negative Words:  43 ( 46.74 %)

 New Zealand
No of Stop Words:  546
No of Positive Words:  32 ( 68.09 %)
No of Negative Words:  15 ( 31.91 %)

 Japan
No of Stop Words:  3340
No of Positive Words:  115 ( 61.83 %)
No of Negative Words:  71 ( 38.17 %)

 Singapore
No of Stop Words:  1662
No of Positive Words:  79 ( 66.39 %)
No of Negative Words:  40 ( 33.61 %)

 Saudi Arabia
No of Stop Words:  1287
No of Positive Words:  57 ( 61.29 %)
No of Negative Words:  36 ( 38.71 %)
```

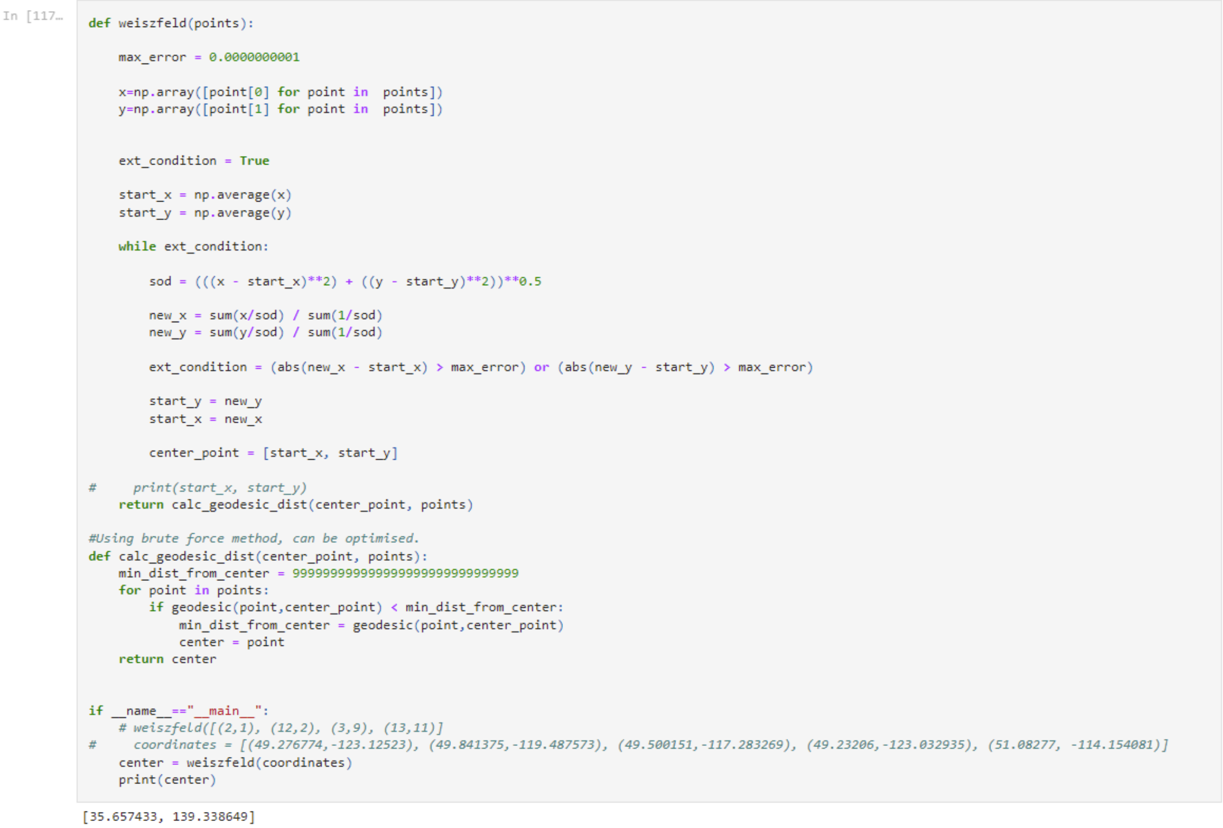


Problem 2:



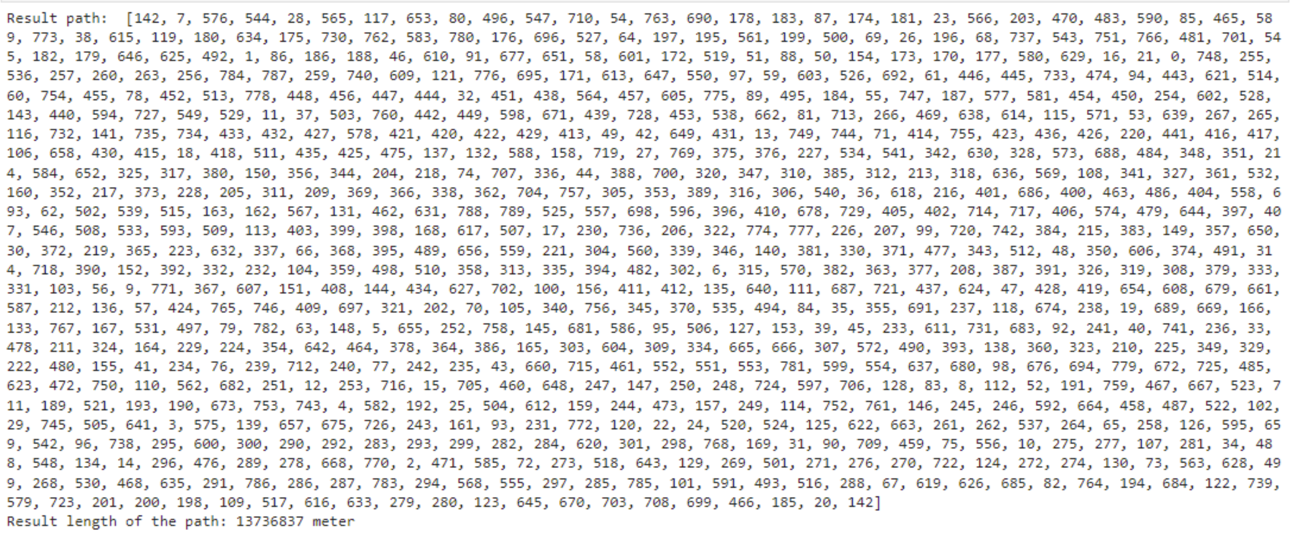
```
In [116…

code = input("Enter country code to analyze:")

# We edited some of the dataset
countries = pd.read_csv (fr"dataset starbucks.csv")
countries = countries[countries.state == code]
df = countries[["latitude","longitude"]]

coordinates=df.values.tolist()

df
```

```
Enter country code to analyze:JP
```

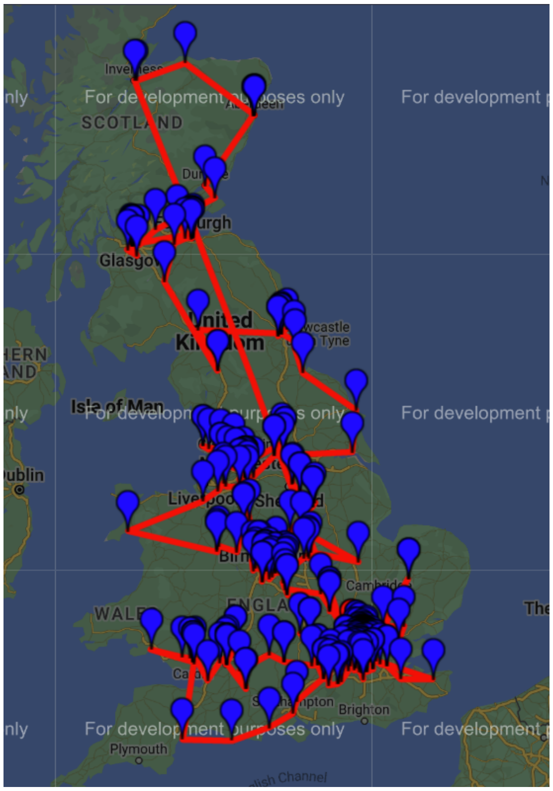


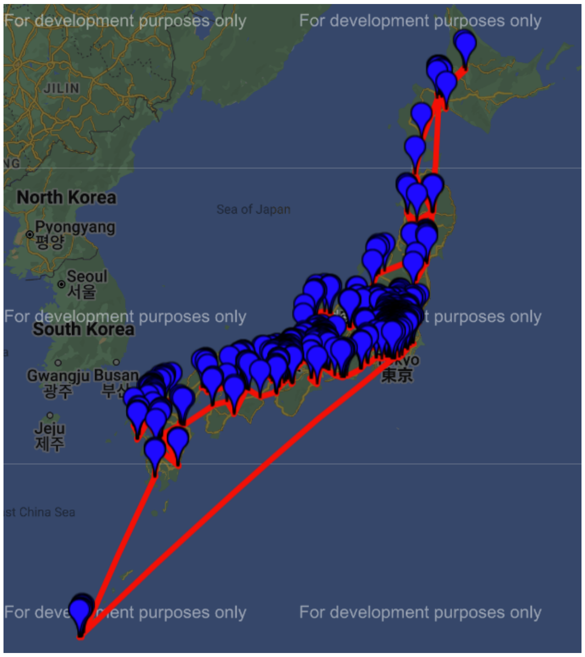
```
In [117…

def weiszfeld(points):

    max_error = 0.0000000001

    x=np.array([point[0] for point in  points])
    y=np.array([point[1] for point in  points])


    ext_condition = True

    start_x = np.average(x)
    start_y = np.average(y)

    while ext_condition:

        sod = (((x - start_x)**2) + ((y - start_y)**2))**0.5

        new_x = sum(x/sod) / sum(1/sod)
        new_y = sum(y/sod) / sum(1/sod)

        ext_condition = (abs(new_x - start_x) > max_error) or (abs(new_y - start_y) > max_error)

        start_y = new_y
        start_x = new_x

        center_point = [start_x, start_y]

#     print(start_x, start_y)
    return calc_geodesic_dist(center_point, points)

#Using brute force method, can be optimised.
def calc_geodesic_dist(center_point, points):
    min_dist_from_center = 999999999999999999999999999999
    for point in points:
        if geodesic(point,center_point) < min_dist_from_center:
            min_dist_from_center = geodesic(point,center_point)
            center = point
    return center


if __name__=="__main__":
    # weiszfeld([(2,1), (12,2), (3,9), (13,11)]
#     coordinates = [(49.276774,-123.12523), (49.841375,-119.487573), (49.500151,-117.283269), (49.23206,-123.032935), (51.08277, -114.154081)]
    center = weiszfeld(coordinates)
    print(center)
```

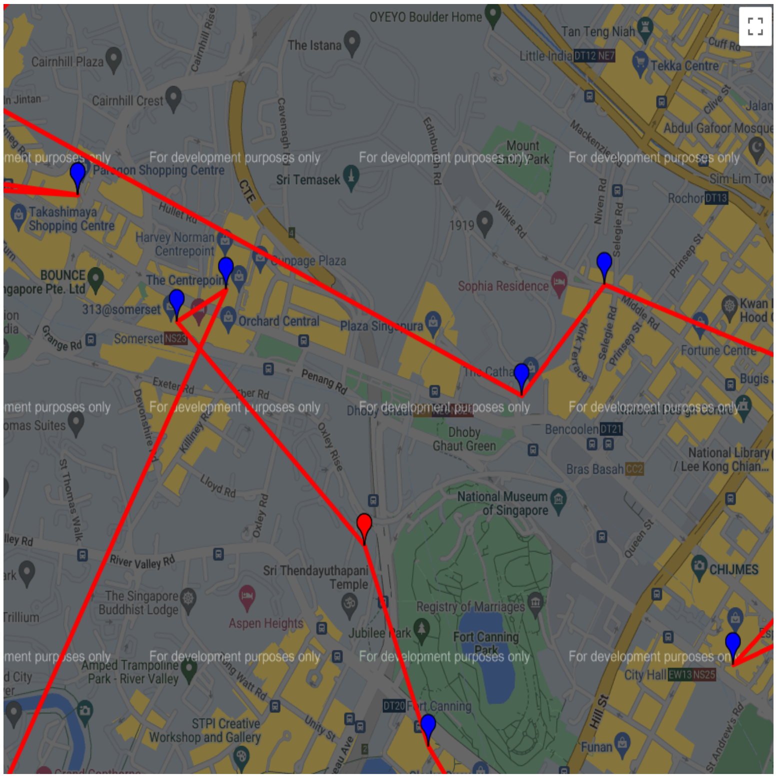
```
[35.657433, 139.338649]
```

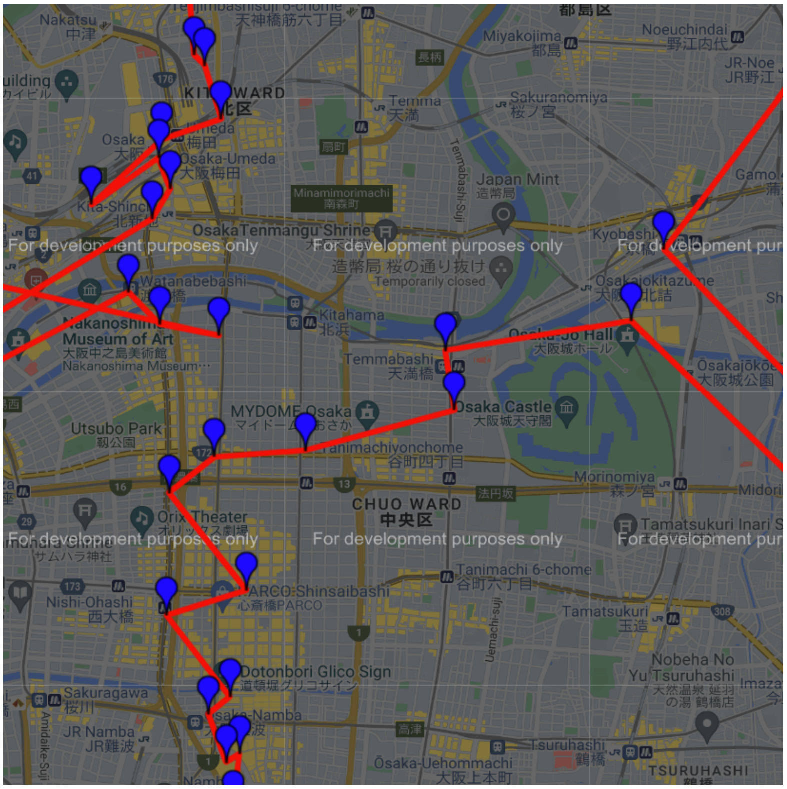


```
Result path:  [142, 7, 576, 544, 28, 565, 117, 653, 80, 496, 547, 710, 54, 763, 690, 178, 183, 87, 174, 181, 23, 566, 203, 470, 483, 590, 85, 465, 58
9, 773, 38, 615, 119, 180, 634, 175, 730, 762, 583, 780, 176, 696, 527, 64, 197, 195, 561, 199, 500, 69, 26, 196, 68, 737, 543, 751, 766, 481, 701, 54
5, 182, 179, 646, 625, 492, 1, 86, 186, 188, 46, 610, 91, 677, 651, 58, 601, 172, 519, 51, 88, 50, 154, 173, 170, 177, 580, 629, 16, 21, 0, 748, 255,
536, 257, 260, 263, 256, 784, 787, 259, 740, 609, 121, 776, 695, 171, 613, 647, 550, 97, 59, 603, 526, 692, 61, 446, 445, 733, 474, 94, 443, 621, 514,
60, 754, 455, 78, 452, 513, 778, 448, 456, 447, 444, 32, 451, 438, 564, 457, 605, 775, 89, 495, 184, 55, 747, 187, 577, 581, 454, 450, 254, 602, 528,
143, 440, 594, 727, 549, 529, 11, 37, 503, 760, 442, 449, 598, 671, 439, 728, 453, 538, 662, 81, 713, 266, 469, 638, 614, 115, 571, 53, 639, 267, 265,
116, 732, 141, 735, 734, 433, 432, 427, 578, 421, 420, 422, 429, 413, 49, 42, 649, 431, 13, 749, 744, 71, 414, 755, 423, 436, 426, 220, 441, 416, 417,
106, 658, 430, 415, 18, 418, 511, 435, 425, 475, 137, 132, 588, 158, 719, 27, 769, 375, 376, 227, 534, 541, 342, 630, 328, 573, 688, 484, 348, 351, 21
4, 584, 652, 325, 317, 380, 150, 356, 344, 204, 218, 74, 707, 336, 44, 388, 700, 320, 347, 310, 385, 312, 213, 318, 636, 569, 108, 341, 327, 361, 532,
160, 352, 217, 373, 228, 205, 311, 209, 369, 366, 338, 362, 704, 757, 305, 353, 389, 316, 306, 540, 36, 618, 216, 401, 686, 400, 463, 486, 404, 558, 6
93, 62, 502, 539, 515, 163, 162, 567, 131, 462, 631, 788, 789, 525, 557, 698, 596, 396, 410, 678, 729, 405, 402, 714, 717, 406, 574, 479, 644, 397, 40
7, 546, 508, 533, 593, 509, 113, 403, 399, 398, 168, 617, 507, 17, 230, 736, 206, 322, 774, 777, 226, 207, 99, 720, 742, 384, 215, 383, 149, 357, 650,
30, 372, 219, 365, 223, 632, 337, 66, 368, 395, 489, 656, 559, 221, 304, 560, 339, 346, 140, 381, 330, 371, 477, 343, 512, 48, 350, 606, 374, 491, 31
4, 718, 390, 152, 392, 332, 232, 104, 359, 498, 510, 358, 313, 335, 394, 482, 302, 6, 315, 570, 382, 363, 377, 208, 387, 391, 326, 319, 308, 379, 333,
331, 103, 56, 9, 771, 367, 607, 151, 408, 144, 434, 627, 702, 100, 156, 411, 412, 135, 640, 111, 687, 721, 437, 624, 47, 428, 419, 654, 608, 679, 661,
587, 212, 136, 57, 424, 765, 746, 409, 697, 321, 202, 70, 105, 340, 756, 345, 370, 535, 494, 84, 35, 355, 691, 237, 118, 674, 238, 19, 689, 669, 166,
133, 767, 167, 531, 497, 79, 782, 63, 148, 5, 655, 252, 758, 145, 681, 586, 95, 506, 127, 153, 39, 45, 233, 611, 731, 683, 92, 241, 40, 741, 236, 33,
478, 211, 324, 164, 229, 224, 354, 642, 464, 378, 364, 386, 165, 303, 604, 309, 334, 665, 666, 307, 572, 490, 393, 138, 360, 323, 210, 225, 349, 329,
222, 480, 155, 41, 234, 76, 239, 712, 240, 77, 242, 235, 43, 660, 715, 461, 552, 551, 553, 781, 599, 554, 637, 680, 98, 676, 694, 779, 672, 725, 485,
623, 472, 750, 110, 562, 682, 251, 12, 253, 716, 15, 705, 460, 648, 247, 147, 250, 248, 724, 597, 706, 128, 83, 8, 112, 52, 191, 759, 467, 667, 523, 7
11, 189, 521, 193, 190, 673, 753, 743, 4, 582, 192, 25, 504, 612, 159, 244, 473, 157, 249, 114, 752, 761, 146, 245, 246, 592, 664, 458, 487, 522, 102,
29, 745, 505, 641, 3, 575, 139, 657, 675, 726, 243, 161, 93, 231, 772, 120, 22, 24, 520, 524, 125, 622, 663, 261, 262, 537, 264, 65, 258, 126, 595, 65
9, 542, 96, 738, 295, 600, 300, 290, 292, 283, 293, 299, 282, 284, 620, 301, 298, 768, 169, 31, 90, 709, 459, 75, 556, 10, 275, 277, 107, 281, 34, 48
8, 548, 134, 14, 296, 476, 289, 278, 668, 770, 2, 471, 585, 72, 273, 518, 643, 129, 269, 501, 271, 276, 270, 722, 124, 272, 274, 130, 73, 563, 628, 49
9, 268, 530, 468, 635, 291, 786, 286, 287, 783, 294, 568, 555, 297, 285, 785, 101, 591, 493, 516, 288, 67, 619, 626, 685, 82, 764, 194, 684, 122, 739,
579, 723, 201, 200, 198, 109, 517, 616, 633, 279, 280, 123, 645, 670, 703, 708, 699, 466, 185, 20, 142]
Result length of the path: 13736837 meter
```

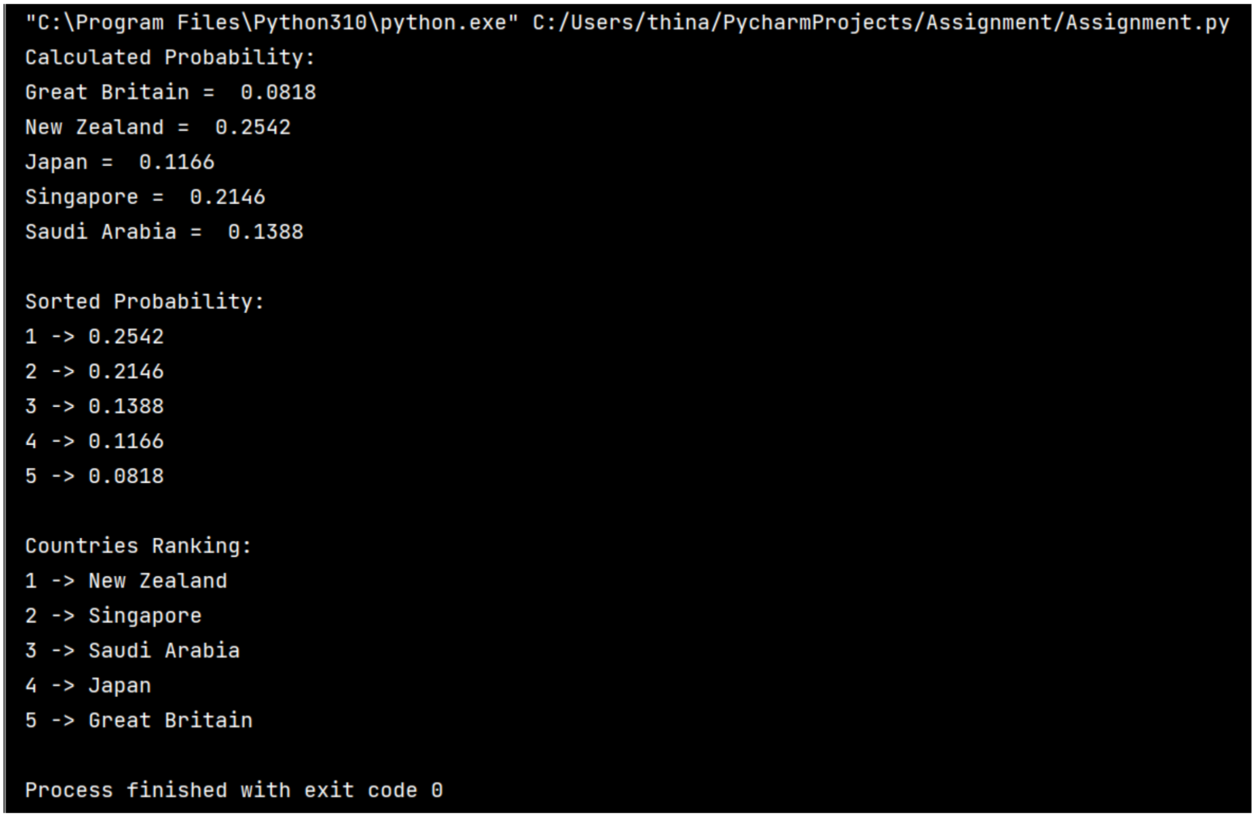




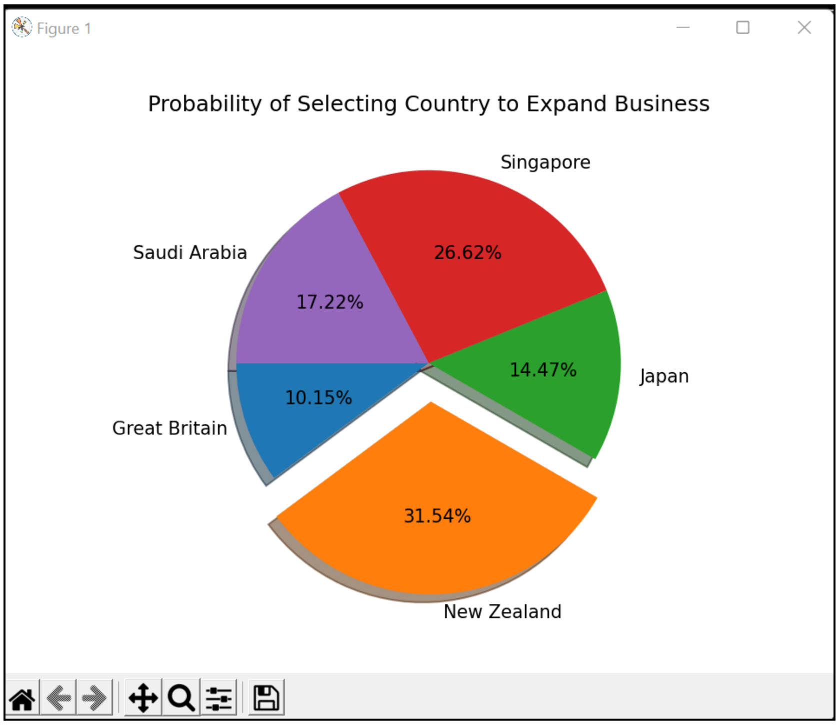

OYEYO Boulder Home
The Istana
Tan Teng Niah
Little India
Tekka Centre
Cairnhill Plaza
Cairnhill Crest
Paragon Shopping Centre
Sri Temasek
Abdul Gafoor Mosque
Rochor
Takashimaya Shopping Centre
Harvey Norman Centrepoint
Orchard Central
1919
Sophia Residence
Plaza Singapura
The Centrepoint
313@somerset
Somerset
Bencoolen
Dhoby Ghaut Green
Bras Basah
National Museum of Singapore
National Library / Lee Kong Chian...
Sri Thendayuthapani Temple
The Singapore Buddhist Lodge
Aspen Heights
Registry of Marriages
Fort Canning Park
Jubilee Park
CHIJMES
City Hall
Amped Trampoline Park - River Valley
STPI Creative Workshop and Gallery
Fort Canning
Funan
For development purposes only


Nakatsu 中津
天神橋筋六丁目
都島区
Miyakojima 都島
Noeuchindai 野江内代
KITA WARD 北区
Umeda 梅田
Osaka-Umeda 大阪梅田
Temma 天満
Sakuranomiya 桜ノ宮
Minamimorimachi 南森町
Japan Mint 造幣局
Kita-Shinchi 北新地
Osaka Tenmangu Shrine
Kyobashi
造幣局 桜の通り抜け
Temporarily closed
Watanabebashi
Osakajokitazume 大阪城北詰
Nakanoshima Museum of Art 大阪中之島美術館
Kitahama 北浜
Osaka-jo Hall 大阪城ホール
Temmabashi 天満橋
Osaka Castle 大阪城天守閣
Osakajokoen 大阪城公園
Utsubo Park 靱公園
MYDOME Osaka マイドーム おおさか
Tanimachiyonchome 谷町四丁目
Morinomiya 森ノ宮
CHUO WARD 中央区
Orix Theater オリックス劇場
Tamatsukuri Inari
Nishi-Ohashi 西大橋
PARCO Shinsaibashi 心斎橋PARCO
Tanimachi 6-chome 谷町六丁目
Tamatsukuri 玉造
Nobeha No Yu Tsuruhashi
Dotonbori Glico Sign 道頓堀グリコサイン
Sakuragawa 桜川
Osaka-Namba
JR Namba JR難波
Osaka-Uehommachi 大阪上本町
Tsuruhashi 鶴橋
TSURUHASHI
For development purposes only

Problem 3:



```
"C:\Program Files\Python310\python.exe" C:/Users/thina/PycharmProjects/Assignment/Assignment.py
Calculated Probability:
Great Britain =  0.0818
New Zealand =  0.2542
Japan =  0.1166
Singapore =  0.2146
Saudi Arabia =  0.1388

Sorted Probability:
1 -> 0.2542
2 -> 0.2146
3 -> 0.1388
4 -> 0.1166
5 -> 0.0818

Countries Ranking:
1 -> New Zealand
2 -> Singapore
3 -> Saudi Arabia
4 -> Japan
5 -> Great Britain

Process finished with exit code 0
```



Conclusion

We have attempted various ways to solve each problem. In problem 1, we were required to do sentiment analysis on 5 countries based on their local economy and social situation where this problem requires us to determine the likelihood of a country having a good local economic and social situation. For problem 2, we are required to locate a local distribution facility in each of the countries that we previously selected, as well as the most efficient delivery route. Problem 3 requires us to calculate the probability of a country having a good local economic and social situation with the lowest optimal delivery. According to our findings, New Zealand has the greatest sentiment score, followed by Singapore, Japan, Saudi Arabia and the United Kingdom. This is due to the fact that New Zealand has the highest positive word percentage compared to the other countries. Based on our result in problem 2, we deduce that Weiszfeld’s Algorithm and the Christofides Approach is the most optimal solution to solve this problem. For the outcome in problem 3, we ranked which country is the best to expand the MoonBucks’ business starting with New Zealand, Singapore, Saudi Arabia, Japan and Great Britain.

References

Akiyoshi, T. (2022, January 10). *The Japanese Economy in 2022: Overcoming Obstacles to Keep Recovery on Track*. Nippon.com. https://www.nippon.com/en/in-depth/d00778/

Alam, K. (2022, January 9). *New Zealand Economy-CPI, GDP, Trade Balance*. InvestoPower. https://www.investopower.com/new-zealand-economy/

Amnesty International. (2018). *Saudi Arabia Archives*. Amnesty International. https://www.amnesty.org/en/location/middle-east-and-north-africa/saudi-arabia/#research

Britannica. (2022). Saudi Arabia - Government and society | Britannica. In *Encyclopædia Britannica*. https://www.britannica.com/place/Saudi-Arabia/Government-and-society

Chew, P. K. H., Young, J. L., & Tan, G. P. K. (2019). Racism and the Pinkerton syndrome in Singapore: effects of race on hiring decisions. *Journal of Pacific Rim Psychology*, *13*. https://doi.org/10.1017/prp.2019.9

Chua, B. H. (2021, September). *Race, racism and racial privilege in Singapore*. The Straits Times. https://www.straitstimes.com/opinion/race-racism-and-racial-privilege-in-singapore

Empire of Japan | Britannica. (2022). In *Encyclopædia Britannica*. https://www.britannica.com/place/Empire-of-Japan/additional-info#contributors

Gordon, A. (2018). *Social Conflict and Control, Protest and Repression (Japan)*. https://encyclopedia.1914-1918-online.net/pdf/1914-1918-Online-social_conflict_and_control_protest_and_repression_japan-2018-05-02.pdf

Hays, J. (2012). *SINGAPOREAN SOCIETY | Facts and Details*. Factsanddetails.com. https://factsanddetails.com/southeast-asia/Singapore/sub5_7b/entry-3735.html

Heritage. (2022). *New Zealand Economy: Population, GDP, Inflation, Business, Trade, FDI, Corruption*. Heritage.org; Index of Economic Freedom. https://www.heritage.org/index/country/newzealand

Human Rights Watch. (2021, December 3). *World Report 2022: Rights Trends in Singapore*. Human Rights Watch. https://www.hrw.org/world-report/2022/country-chapters/singapore

International Trade Administration. (2021). *Saudi Arabia - Market Overview*. International Trade Administration | Trade.gov. https://www.trade.gov/knowledge-product/saudi-arabia-market-overview

Loria, S. (2020). *TextBlob: Simplified Text Processing — TextBlob 0.16.0 documentation*. Readthedocs.io. https://textblob.readthedocs.io/en/dev/

Matthew, R., Cherukupalli, D., Pusich, K., & Zhao, K. (2022). *Travelling Salesman Algorithms*. Github.io. https://cse442-17f.github.io/Traveling-Salesman-Algorithms/#christofides_header

Mourdoukoutas, P. (2022). *3 Economic Challenges Facing Japan in 2022*. Investopedia. https://www.investopedia.com/articles/investing/123015/3-economic-challenges-japan-faces-2016.asp

Neoh, A., Wilder-Smith, M., Chen, H., & Chase, C. *An Evaluation of the Traveling Salesman Problem*. California State Polytechnic University, Pomona. https://doi.org/http://localhost/files/8g84mp499

New Zealand Immigration. (2021, March 17). *Stability & security*. New Zealand Now. https://www.newzealandnow.govt.nz/invest-innovate-in-new-zealand/nzs-business-environment/stability-security

OECD. (2022). *New Zealand Economic Snapshot - OECD*. Oecd.org. https://www.oecd.org/economy/new-zealand-economic-snapshot/

Partington, R. (2022, January 2). *What does 2022 hold for the UK economy and its households?* The Guardian; The Guardian. https://www.theguardian.com/business/2022/jan/02/what-does-2022-hold-for-the-uk-economy-and-its-households

PESTLEanalysis Contributor. (2015, March 17). *PESTLE Analysis of Singapore*. PESTLE Analysis. https://pestleanalysis.com/pestle-analysis-of-singapore/

Retsediv. (2017, November 18). *Retsediv/ChristofidesAlgorithm*. GitHub. https://github.com/Retsediv/ChristofidesAlgorithm

Roth, K. (2021, January 13). *World Report 2021: Rights Trends in United Kingdom*. Human Rights Watch. https://www.hrw.org/world-report/2021/country-chapters/united-kingdom

Schumacher, A. (2015). *TextBlob Sentiment: Calculating Polarity and Subjectivity*. Planspace.org. https://planspace.org/20150607-textblob_sentiment/

Smith, E. (2022, February 11). *UK economy grew 7.5% in 2021, mostly recovering from its pandemic plunge*. CNBC; CNBC. https://www.cnbc.com/2022/02/11/uk-economy-grew-7point5percent-in-2021-mostly-recovering-from-its-pandemic-plunge.html

Social Mobility Commission. (2021). *State of the Nation 2021: Social Mobility and the Pandemic*. https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1003977/State_of_the_nation_2021_-_Social_mobility_and_the_pandemic.pdf

The Heritage Foundation. (2022). *Japan Economy: Population, Facts, GDP, Business, Trade, Unemployment*. Heritage.org; Index of Economic Freedom. https://www.heritage.org/index/country/japan

The World Bank. (2022). *Saudi Arabia's Economic Update — April 2022*. The World Bank. https://www.worldbank.org/en/country/saudiarabia/publication/economic-update-april-2022

Todman, W. (2019). *Is Saudi Arabia Undergoing a Social Revolution?* Csis.org. https://www.csis.org/analysis/saudi-arabia-undergoing-social-revolution

UK economy latest. (2021, January 25). Ons.gov.uk; Office for National Statistics. https://www.ons.gov.uk/economy/economicoutputandproductivity/output/articles/ukeconomylatest/2021-01-25#:~:text=UK%20GDP%20is%20estimated%20to%20have%20fallen%20by%200.1%25%20in%20March%202022,-12%20May%202022&text=In%20March%202022%2C%208%20of,10.7%25%20from%20February%202020

Üster, H., & Love, R. F. (2000). The convergence of the Weiszfeld algorithm. *Computers & Mathematics with Applications*, *40*(4-5), 443–451. https://doi.org/10.1016/s0898-1221(00)00172-3

Wikipedia Contributors. (2022, June 20). *Economy of New Zealand*. Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/Economy_of_New_Zealand