# Lab2

Yonnatan Lourie, Eitan Zimmerman

4/21/2022



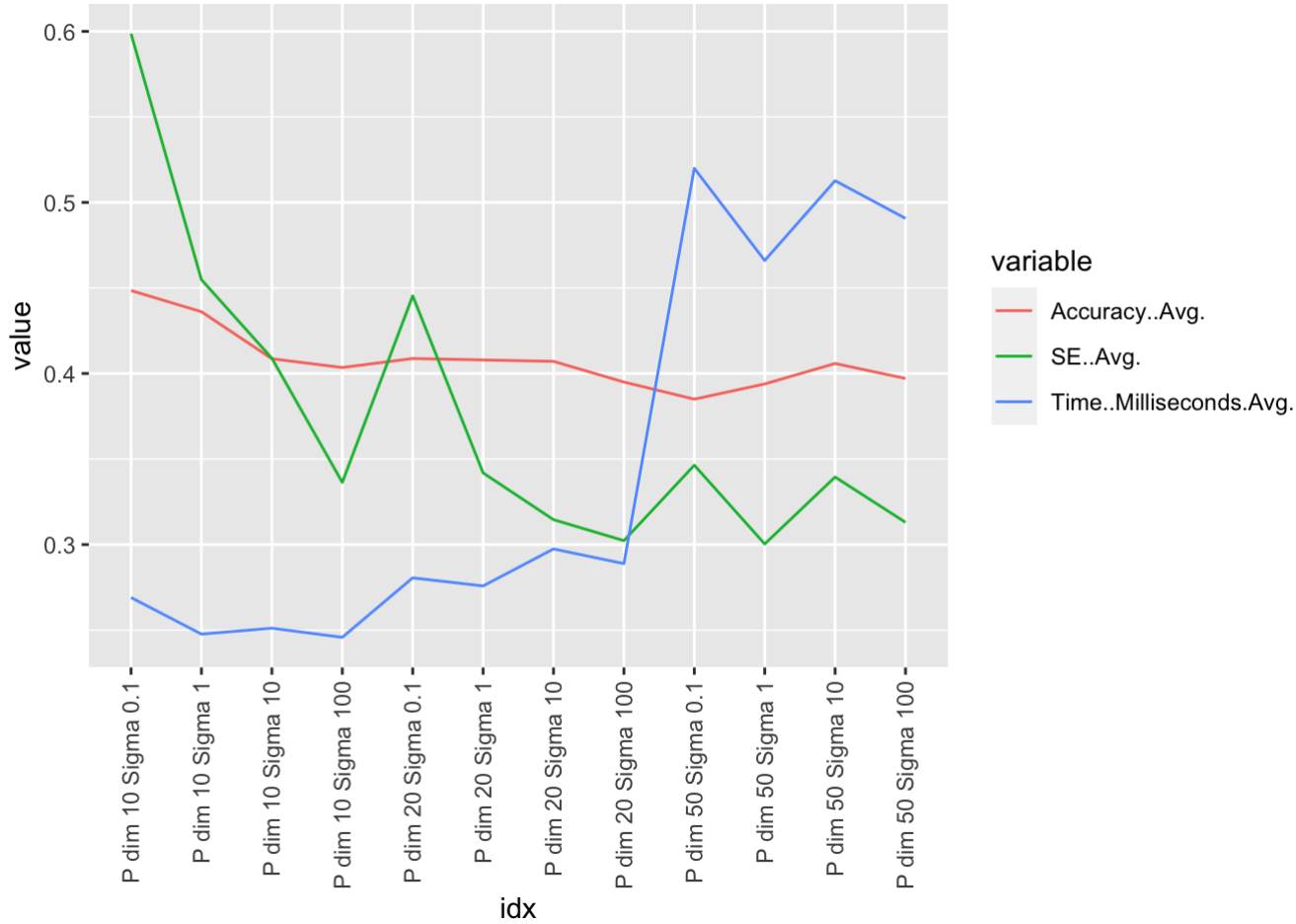
# Lab 2

## Question 1

### Functions

| | Accuracy (Avg) | SE (Avg) |
|---|---|---|
| P dim 10 Sigma 0.1 | 0.4484722 | 0.0059863 |
| P dim 10 Sigma 1 | 0.4361111 | 0.0045483 |
| P dim 10 Sigma 10 | 0.4086111 | 0.0040891 |
| P dim 10 Sigma 100 | 0.4034722 | 0.0033636 |
| P dim 20 Sigma 0.1 | 0.4087500 | 0.0044536 |
| P dim 20 Sigma 1 | 0.4079167 | 0.0034189 |
| P dim 20 Sigma 10 | 0.4070833 | 0.0031455 |
| P dim 20 Sigma 100 | 0.3950000 | 0.0030229 |
| P dim 50 Sigma 0.1 | 0.3850000 | 0.0034638 |
| P dim 50 Sigma 1 | 0.3938889 | 0.0030035 |
| P dim 50 Sigma 10 | 0.4058333 | 0.0033947 |
| P dim 50 Sigma 100 | 0.3970833 | 0.0031299 |

As expected, there are 2 main trends discovered in our simulation. 1. Increasing sigma (e.g variance), making our data to be more spread out, which makes the clusters less distinguishable.We can see this when looking at the accuracy and SE scores for P=10. Both metrics decreases when increasing sigma's value. 2. Increasing the dimensions only by padding the vector with 0 making the different observation more similar as the dimensions grows. For p=50 the vectors are variate only in quarter of their individual entries, the mean of each centroids is highly effected by the 0 and hence the clustering task is much harder. We can see how the trend described in 1 (change of sigma) has little to no effect in the case of p=50.




Note that we normalized the milliseconds (multiply by 100x) so the plot will be in the same scales.

# Question 2

## Pre processing

Some technical preprocessing to make sure both data sets have the citie's name in English.

We used a mapping from code to name downloaded from The NBS.

The code below will contain the technical steps of:

1. Sampling 20 cities from the covid data sets.
2. Finding their matching data in the demographic data.
3. Creating the clustering and dendogram objects that will use us to make some plots and comparisons.

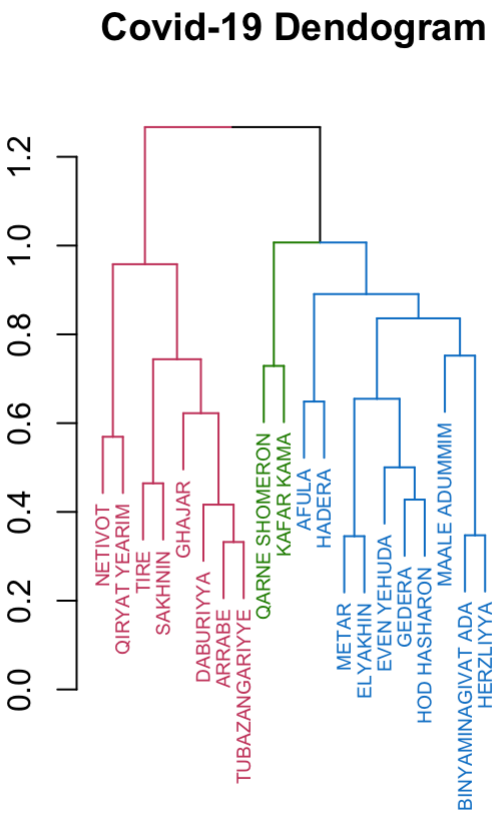
In the code section above we created the hierarchical clusters for both covid and the demographic data. We thought on the best way to create the distance $NxN$ matrix for the samples so they will be best suit for the comparison we will make after between the two.

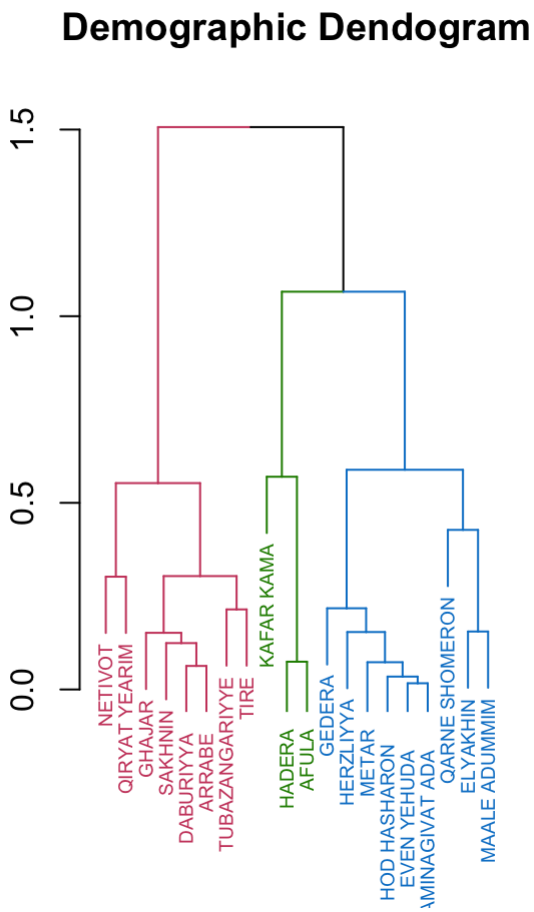
Covid Data - For the dissimilarity matrix we used the cosines distance metric based only on the scores columns of the data. We chose the filter the data to these only as we believed they are capturing in a good way the rest of the data as well (e.g the scores themselves are calculated with respect to other measures like number of cases, vaccination etc..) For our dissimilarity matrix we used the cosines distance. As described here (link to article) the cosines distance calculates the angle between the 2 vectors (the observations in our case),what makes this distance metric a good measure of "style". That is, it will score 2 vectors that "go" in the same direction as close to each other, even if their size is

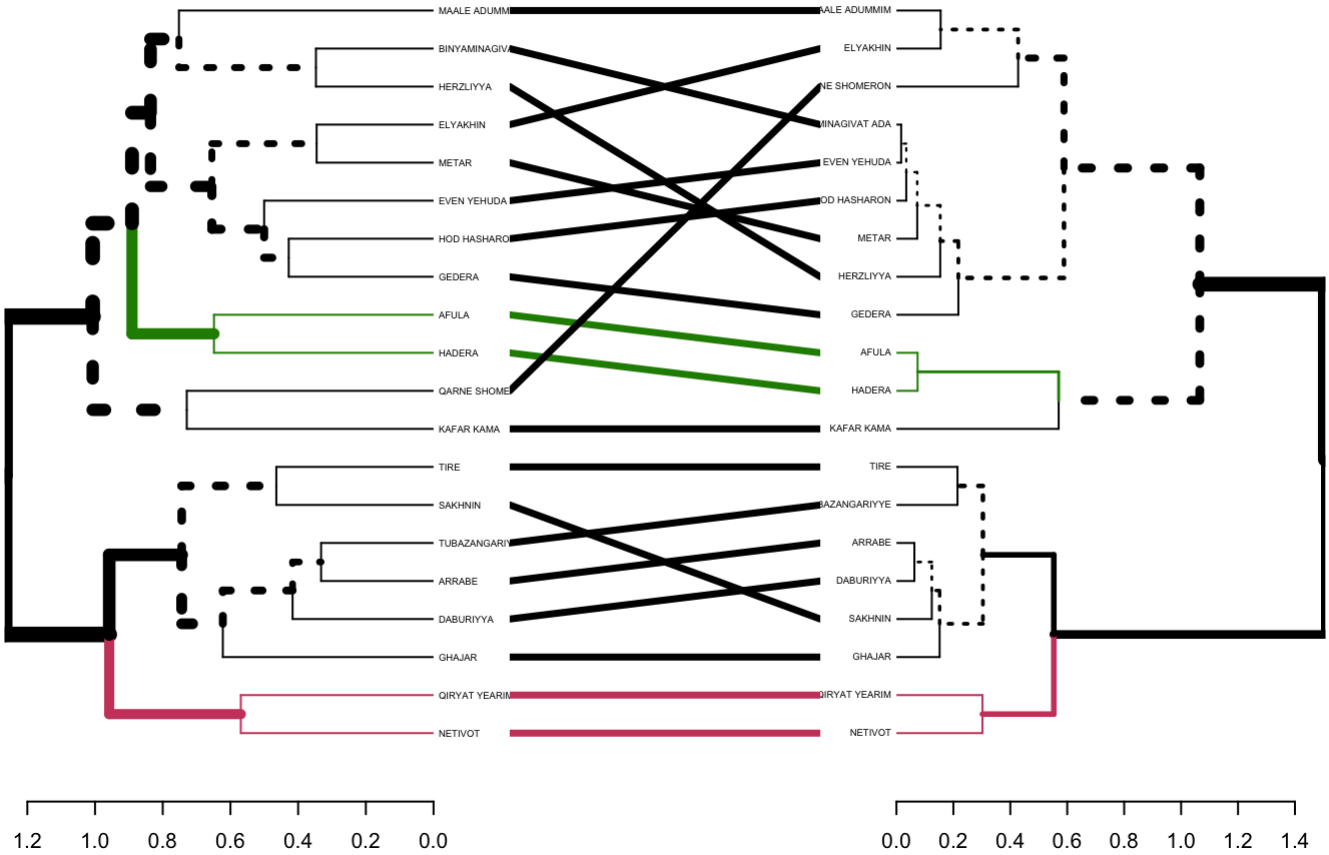
different. In our case, we will cluster similar trends in time represents by the score of different cities even if they had different scores values. We thought it will be a good metric to expose some interesting clusters.

Demographic Data - for the demographic data we disregarded the vehicle columns data, as well as the population and the income. We chose to disregard the population and income as we believe they will create some unwanted connection in our matter, For example we wouldn't want big cities to be clustered together necessarily. We wanted our clusters to represent the cities economical features and priorities and wealth (pct_dgree and pct_woman_income for example) as we believed these features would be better correlated with COVID-19 spreading. For the dissimilarity function we used the cosines distance as well for the same reasons.

Plotting The 2 dendograms colored by K=3 cluster cut.






We can see that when looking at the 3 clusters the similarities are relatively high for the sample chosen. As expected, low socioeconomically cities are clustered together both in the demographic and the Covid dendograms which support the evidence of the same Covid-19 spreading trends we have seen for Arab and Orthodox cities. The higher socioeconomically cities are also clustered together in both dendograms as expected. But there are still some diffrences when looking at the dendograms. Firstly, the height and distinguish between the clusters are way more significant in the Demographic dendogram, that is, it is “easier” to cluster cities based on their demographic data than Covid spreading. This is expected as well as the demographic data is way more deterministic (e.g less flexible) while Covid-19 trends are more prone to be changed rapidly and without any defined structure. Another thing we can spot is that in the lower level (cluster 6+), the two dendograms are less similar. A good explanation for this is that demographic data can be still very different between cities that have same Covid-19 perception. For example, Savyon, being the richest city in Israel, is still far away from Givattaim in terms of their demographic data but their approach towards Covid pandemic is more about the same (both population are well educated, involved politically etc..)

To calculate the correlation between the 2 dendograms we will use the Baker’s Gamme coefficient. From the the dendeExtend doc’s the coefficient defined as:

> *It is calculated by taking two items, and see what is the highest possible level of k (number of cluster groups created when cutting the tree) for which the two item still belongs to the same tree. That k is returned, and the same is done for these two items for the second tree. There are n over 2 combinations of such pairs of items from the items in the tree, and all of these numbers are calculated for each of the two trees. Then, these two sets of numbers (a set for the items in each tree) are paired according to the pairs of items compared, and a Spearman correlation is calculated.*

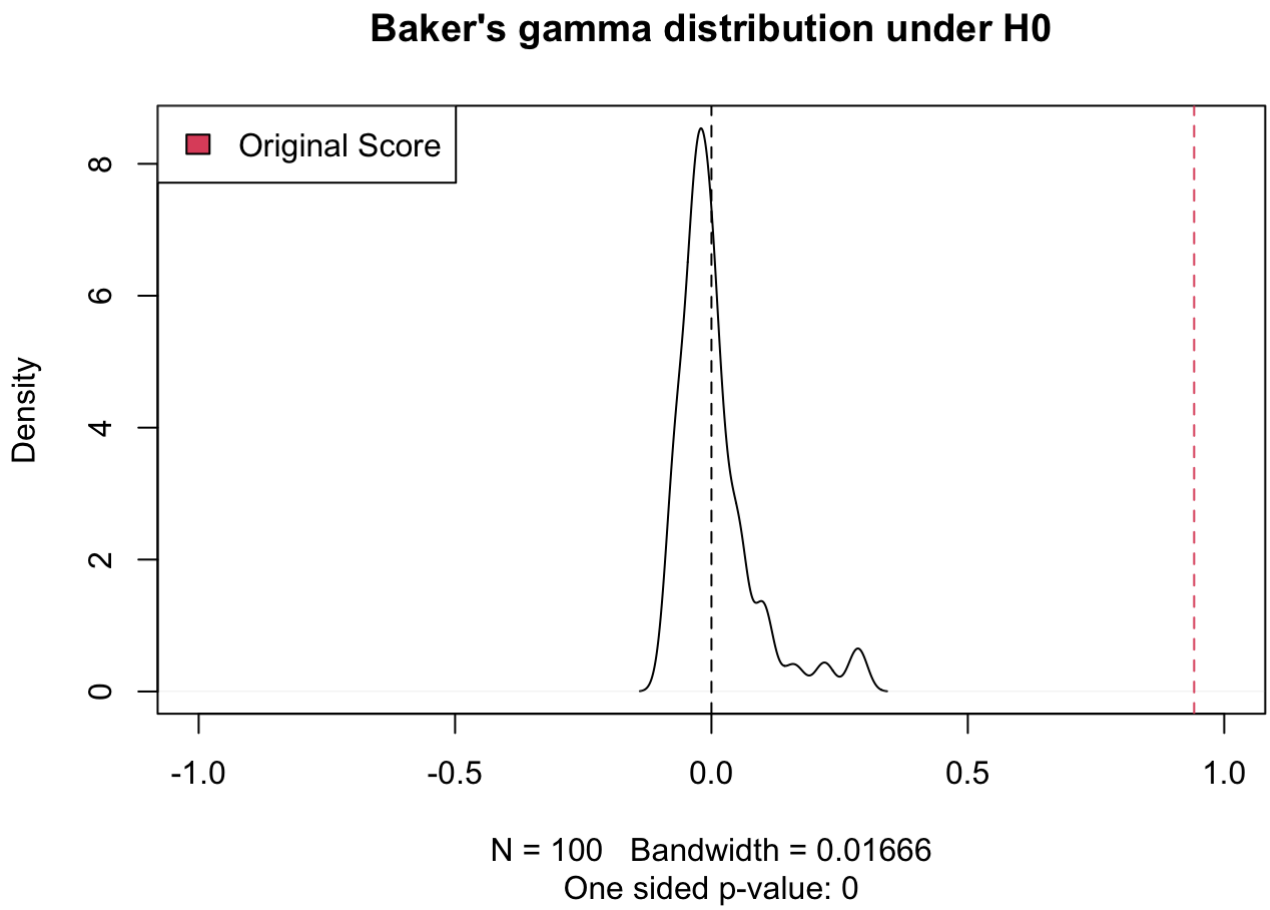
Hence Baker's coefficient takes values in (-1,1) the same as Spearman correlation. higher (absolute score) means the dendograms are similar while values near 0 means they aren't statistically similar.

To be able to define the P-value we will need to use permutation test. The test purpose is to check that our baker's score is significant, that is, depends on the topological order of the dendogrmas. To check this, we will need to permutate the labels of one of the trees many times, calculating the Baker's gamma each time to receive a distribution if the Baker's index under the null hypothesis - Our fixed tree topology.

The bakers gamma correlation between the Covid and Demographic dendograms is:

```
## [1] 0.9413166
```

```
## [1] 0
```



We ran 100 permutation tests to get the distribution above Running it we can see that the similarity between our dendograms is significant (P-value~0). and we can conclude that our trees are significantly similar as defined by Baker's index.

# Question 3

In this question we will create the K-means algorithm from scratch for the genes data and will use Shiny App to allow the user to interact and explore the data.

The codes is divided in 3 main function:

1. PreProcess - the get_data and preprocess functions are preforming these actions on our data
    a. Log transform
    b. Taking the top 200 features with the highest variance across examples
    c. Transposing our data so every row is a tissue and every column is some gene.
    d. Scaling the data and correcting the names of the genes.
2. my_kmeans - k-means algorithm written from scratch by us. Returns the clusters, their centers and the WSS score vector for each iteration.

The shiny app we made allows the user to interact with the data in 2 ways after preforming the clustering:

1. Feature by Feature plot - the user can plot any 2 features side by side in a scatter plots, colored by the clusters.
2. PCA plot - the user can decide to plot the first 2 components of the PCA process colored by clusters.

In both cases the user can define the number of clusters which will immediately effect the plots. (from 2 to 9, any more than this doesn't make really sense as we have only 20 observations).

To run the app you should run the whole block below. The get_data function may take a few seconds to run.