**Final Exercise**

Tutor in charge of this HW: Tamir Eliav. E-mail your solutions to: [matlab.wis@gmail.com](mailto:matlab.wis@gmail.com)

**Instructions**

1. The mail subject should be: **'matlab intro 2021b final'**. Attach only the script file to your mail.
2. Do not submit a script that does not run properly. Before submission make sure to run the entire script from scratch.
3. Name your script "hw\_final\_< ID number 1>\_< ID number 2>.m".
4. Divide your code into **sections** (according to the questions or any other reasonable division). Use the '%% ' sign
5. Give your variables **meaningful names**.
6. **Document** your work with comments in the script. (use the '%' sign).
7. Any number that appears in your script should be assigned to a variable (i.e. do NOT use "magic numbers").
8. Always use **matrix calculations** when applicable, do *not* use unnecessary loop.
9. Do *not* copy-paste code sections to do similar calculations, use loops when necessary.
10. Add appropriate labels/titles/legends to your figures.
11. Look at the documentation of the relevant functions you should use, and explore the function options before using it.
12. Use the following header and questions separator:

%%

% Final HW solution

% <your name & T.Z.>

% for example:

% Final HW solution

% Israel Israeli 123456789

%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% <short description of what your script is doing for

% example: this script load experiments data and

% analyzes it. The output is the statistics

% calculations results.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Q1

% <Document the code: briefly describe the question and how you solved it>

disp('\*\*\* Question 1 \*\*\*');

<Q1 output>

disp(' ');

%% Q2

% <Document the code: briefly describe the question and how you solved it>

disp('\*\*\* Question 2 \*\*\*');

<Q2 output>

disp(' ');

**Introduction**

In this assignment you will explore a dataset from the elections in Israel. You will have to load the data, process it, explore different aspects of it using different visualizations and draw some conclusions.

Make sure to write **generic code**, so your code should work the same also when applying it for all other election data sets.

**Questions**

1. Load the data: read the data in the file ‘*Kneset\_result\_2020a.xlsx*' to a table.  
   Data description:   
   - Each row contains the elections results from one settlement.   
   - First 6 columns are self-explanatory.  
   - Columns 7 and above are the number of votes for each party.  
   - For those who don't know Hebrew, you can ignore the Hebrew names of settlements, it is not an essential part of the assignment.
2. Plot basic elections results:
   1. Figure should be divided to 4 subplots.
   2. Subplot 1: bar plot of total number of votes (in counts), per party, over all settlements. Add horizontal line specifying the votes threshold in Israel (3.25%). Don’t forget to add labels!
   3. Subplot 2: same as previous bar plot, but now yaxis should be presented in a log scale.
   4. Subplot 3: pie chart of the total number of votes (in percentage), per party, over all settlements. For the first 5 biggest parties add their name next to their corresponding slice and also offset their slice (see doc pie).
   5. Subplot 4: in this subplot you should add text that summarize some basic voting stats:
      1. Total registered voters
      2. Total voters
      3. Total voting rate (in percentage)
      4. Total valid/invalid votes
      5. Votes threshold, both the definition in parentage (3.25%) and in vote counts.
3. Find the top 10 settlements that had the highest percentage of valid votes. Do the same for the settlements with the lowest percentage. Display to the command line the names of those parties you found. (If there are several settlements with the same percentage choose the first ones in the least and ignore the others).
4. Explore voting pattern correlations:
   1. Using corr function (see doc corr), calculate the correlation between the voting pattern in each settlement and the general (total) voting pattern (i.e. summed over all settlements).
   2. Find the top 10 settlement that had the highest correlation to the general voting pattern, and display their names to the command line. Find and display also the 10 settlements with the lowest correlations.
   3. Using corr function (see doc corr), calculate the correlation between the voting pattern in each settlement to all other settlement, i.e. all pairs of settlements.
   4. Find the pair of settlements that had the highest correlations between their voting patterns, and display their names to the command line. Do the same for the pair of settlements that had the lowest correlations.
   5. IMORTANT NOTE: use array operations to calculate the correlations. Do NOT use for-loops! (see doc corr)
5. Cluster the voting data to groups according to voting pattern. Each settlement is a sample, and the features are the different parties. Each data point (sample) is a vector of voting *counts* per settlement.
   1. Display in the command line how many samples and how many features you have in the data.
   2. Use kmeans clustering algorithm (see doc kmeans) to find groups in the data.
   3. Explore results using the following 3 distance metrics: 'sqeuclidean', 'cosine', 'correlation'.
   4. Choose the 'replicates' option to be a value between 10 and 100.
   5. Explore results using different number of clusters (k), from 2 clusters to 10 clusters.
   6. For each distance metric find the optimal k using the *silhouette* function, as follows: Generate a figure that contains 9 *silhouette* subplots (see doc silhouette), one for each k. The 10th subplot should show the grand average silhouette value over all points, per k. Mark the optimal k with a red asterisk and add text next to it with the optimal k number. Make sure to add relevant titles and labels to all subplots.
   7. Can you explain why using the 'sqeuclidean' created different results? How can you fix it?

\* Use 'rng(0)' before running each call to kmeans to fix the randomization (i.e. have the same results when re-running the same code again).

1. For the next parts, choose the clustering results with optimal k using the 'correlation' distance metric.
2. Create a figure to plot the clustering results:
   1. Figure should have 4 subplots.
   2. Subplot 1: pie chart showing the percentage of data points in each cluster.
   3. Subplot 2: stem plot (see doc stem) of voting pattern for each cluster (use different colors for each cluster). Add the general voting pattern with a thick black line. Voting patterns should be in percentages. Don't forget to add legends and labels.
   4. Subplot 3: histogram (see doc histogram) of correlations values to total voting pattern (from question 3). Plot several histograms (on top of each other), one for each cluster, use the same colors for the clusters as you used in the previous plots.
   5. Subplot 4: a 3D plot (see doc plot3), with the following axes (each point is a settlement):
      1. Number of votes
      2. Voting rate (in percentage)
      3. Correlation to the total voting pattern.

Color the points according to the cluster they belong to (use the same colors as before).

Set the view angle of the 3D plot to the optimal view of the data.

* 1. Select two clusters and try to explain their results using the figure you created. What is different between those groups?

1. Check that your code until now works also with the results of 2019a (*‘Kneset\_result\_2019a.xlsx’*) and 2019b (*‘Kneset\_result\_2019b.xlsx’* ) Your code should be generic and work with this different data sets. (You don’t need to submit this part, but you should check it!).
2. Comparison between elections:
   1. Load the data from all three elections.
   2. Find which parties participated in all three elections
   3. Create scatter plots of votes in between all pairs of elections.
   4. Compute the correlation between the elections and print it in the title of the graphs above.
   5. Fit a regression line and add it to the figures above.
   6. Create a bar graph of all three elections (one on top of the other) only for the relevant parties similar to the graph you made in q2b.
3. Creative part (bonus 5 point): Here you can choose an analysis you think will be interesting to do. You can choose to do an analysis just on one of the election or do an interesting comparison between them.