**Final Exercise**

Tutor in charge of this HW: Shir Maimon. Questions will be answered through the Moodle forum.

Read **carefully** the following instructions, and don’t hesitate to reach us (preferably through the forum) if something is not clear.

**Moodle instructions:**

1. Before submission, you need to create a team through the “Team Work” window. You can team with different students for every HW, but even if you don’t change your team – you still need to do this step for every assignment.
2. Submit the .m file directly to the Moodle. Only one member of the team need to submit – but **all** of you should make sure that you see your submitted file in your Moodle!

**General Matlab instructions:**

1. **Do not submit a script that does not run properly**. To check that your script run without errors, make sure your workspace is cleared, and then run the script (press F5). If you are reading data, avoid full paths (“C:\My\_computer\HW\data.mat”), as they will return an error when you run the code on a different machine (i.e. the computer of the TA who checks your HW ☺).
2. Name your script "hw5\_< ID number 1>\_< ID number 2>.m".
3. Start your script with a header – a few lines of comments which describe your script. The header should contain **all** of this information: your names, IDs, HW number and the Matlab version (e.g. (R2021a) you wrote your code on.
4. Divide your code into **sections** according to the questions, or any other reasonable division (use the '%%' sign). At the beginning of each question, write its number and a short explanation of your answer – e.g. what the code does, etc.
5. If the question involves printing a result to the console, you should **also** print the question number: You can use “*disp('\*\*\* Question ## \*\*\*')*”.
6. **Document** your work with comments in the script (use the '%' sign). Over-documentation is better than under-documentation: we want to see that you understand how your code is working at each step.
7. Give your variables **meaningful names**.
8. For every question that has a defined output (a number, a figure, etc. – not an “open question”), **read carefully** the HW instructions and make sure that your output is **exactly** what we asked, and what you would expect it to be. This is very important also for you in your “debugging” procedure. For example, if you expect a scalar output but your code returns a matrix (and vice versa), you clearly have a mistake somewhere… ☺
9. **Do not** use "magic numbers" – **every** number that appears in your script should be assigned to a variable, preferably at the beginning of the code section.
10. Do not calculate anything implicitly, but make your code calculate everything explicitly. For example, if you have a vector with the values [2, 5, 3, 1] and you need to find its largest value, you **cannot** write “*disp(5)*”.
11. **Avoid** “code duplication”! Code duplication is copying & pasting lines of code with minimal changes. Instead, use matrix and vector operations, and loops.

**Introduction**

In this assignment you will explore a dataset from the 2020 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.
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 (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.
   4. Find the two settlements that had the highest correlations between their voting pattern, and display their names to the command line. Do the same for the two settlements that had the lowest correlations.
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. 'replicates' option should be 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 plot.
   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 the two elections:
   1. Find which parties participated in all three elections
   2. Create a scatter plots of votes in between all pairs of elections
   3. Compute the correlation between the elections and print it in the title of the graphs above.
   4. Fit a regression line and add it to the figures above.
   5. 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. In 2019b GESHER and AVODA were merged and then they merged with MERETZ in 2020. Let’s explore it compared to 2019a:
   1. Subplot 1: Plot a bar graph with the sum of votes for GESHER, AVODA and MERETZ in 2019a, AVODA+GESHER, MERETZ in 2019b and AVODA+GESHER+MERETZ in 2020. Add horizontal line specifying the votes threshold in Israel (3.25%) of election 2020.
   2. Is the sum larger than its parts? (AVODA+GESHER+MERETZ in 2020b >GESHER, AVODA and MERETZ in 2019a). write in the title of the graph.
   3. Subplot 2: Find if the union of AVODA+GESHER+MERETZ changed something the voting pattern of the “Likud-settlements”: find the settlements with the most votes for the Likud (top 100 in percentage) and measure weather the union of AVODA+GESHER+MERETZ in those settlements in election 2020 is more the sum of votes for AVODA,GESHER and MERETZ in 2019a. plot a bar graph similar to q9a just for those settlements. NOTE- there are small differences in the settlements order, but let’s assume it is the same because it is hard to work with the Hebrew data.
4. 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 another interesting comparison between them.