Assignment 4: Advanced Topics

Submission

Description: My chosen SDG in my submission is ***Reduced Inequalities***.

Task 1: Dimensionality Reduction

- **Dataset:** Income level, income inequality, population at risk of poverty or dependent on basic social security by region in the dwelling-population, 2023.

Note: The dataset contains 421 observations (rows) which are different regions across Finland and 4 distinct variables (columns) which are: Dwelling population, persons; At risk of poverty rate (threshold 60 % of median); At risk of poverty rate of children in dwelling population; Gini coefficient, gross income.

This is the partial dataset that I can show because I cannot upload this big dataset into this PDF:

Income level, income inequality, population at risk of poverty or dependent on basic social security in the dwelling-population by Year, Region and Information

	Dwelling population, persons	At risk of poverty rate (threshold 60 % of median)	At risk of poverty rate of children in dwelling population	Gini coefficient, gross income
2023				
WHOLE COUNTRY	5,478,794	13.4	12.2	33.0
Akaa	16,075	11.4	9.5	27.5
Alajärvi	8,852	16.4	19.3	28.4
Alavieska	2,381	18.4	22.5	35.9
Alavus	10,733	15.4	14.6	29.0
Asikkala	7,806	13.8	15.4	30.7
Askola	4,617	8.3	10.2	29.7
Aura	3,915	9.1	7.7	24.8
Brändö	417	12.2	...	35.6
Eckerö	913	13.1	...	27.8
Enonkoski	1,281	17.9	17.4	31.1
Enontekiö	1,680	17.9	19.9	29.6
Espoo	307,563	10.6	10.4	37.6
Eura	10,955	11.6	12.9	28.9

Using the filter function when accessing the link to the dataset below will create the above dataset.

- **Link to the dataset:** https://pxdata.stat.fi/PxWeb/pxweb/en/StatFin/StatFin__tjt/statfin_tjt_pxt_14ww.px/
- **Visualization tools:** Python with Matplotlib, Seaborn libraries, and Scikit learn.

Visualization: *Apply PCA, MDS, and t-SNE to 421 observations and then color them by 5 groups. I group 421 observations into 5 different groups as follows:*

```python
# Categorize regions
def categorize_region(region):
    if region == 'WHOLE COUNTRY':
        return 'Whole Country'
    elif region.startswith('MK'):
        return 'Regions (MK)'
    elif region.startswith('HVA'):
        return 'Wellbeing Services Counties (HVA)'
    elif region.startswith('SK'):
        return 'Subregions (SK)'
    else:
        return 'Municipalities'
```

1. **PCA once without standardization and once with standardization:**

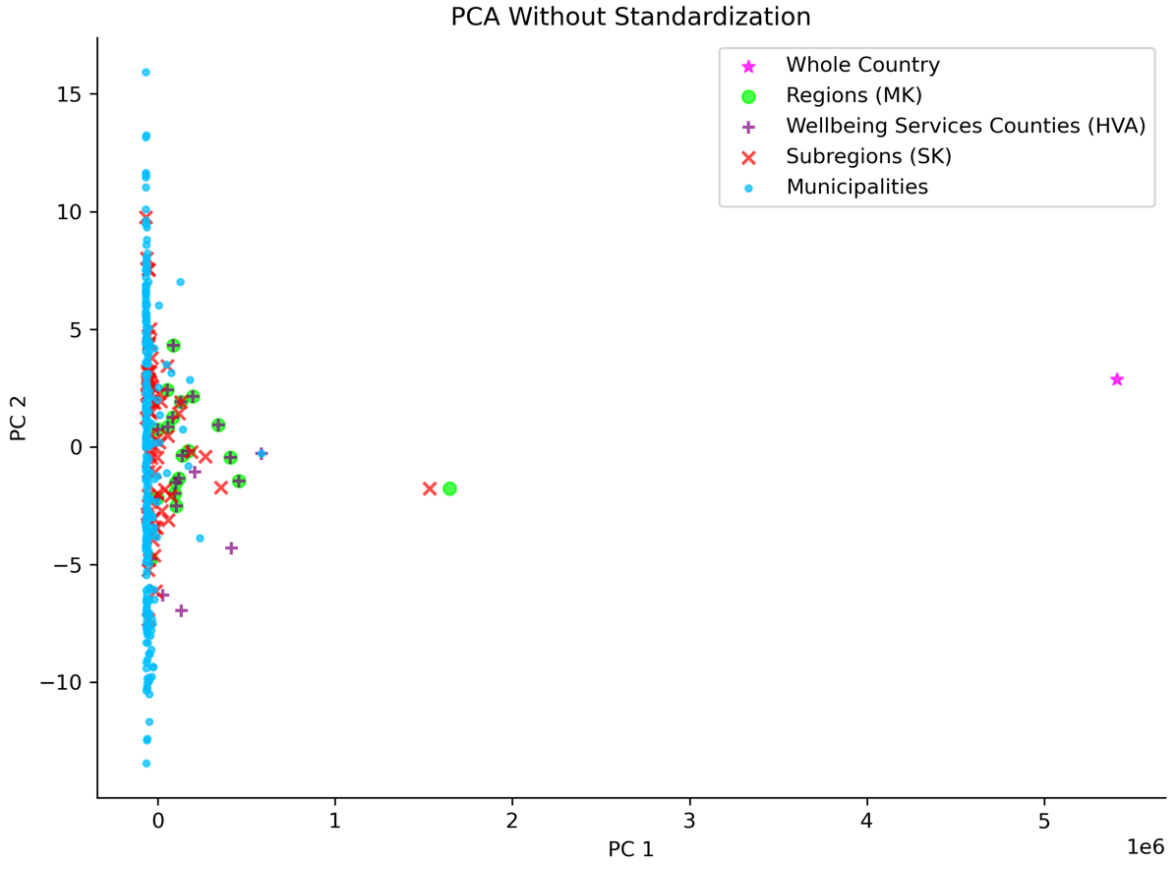


Figure 1: *PCA without standardization*

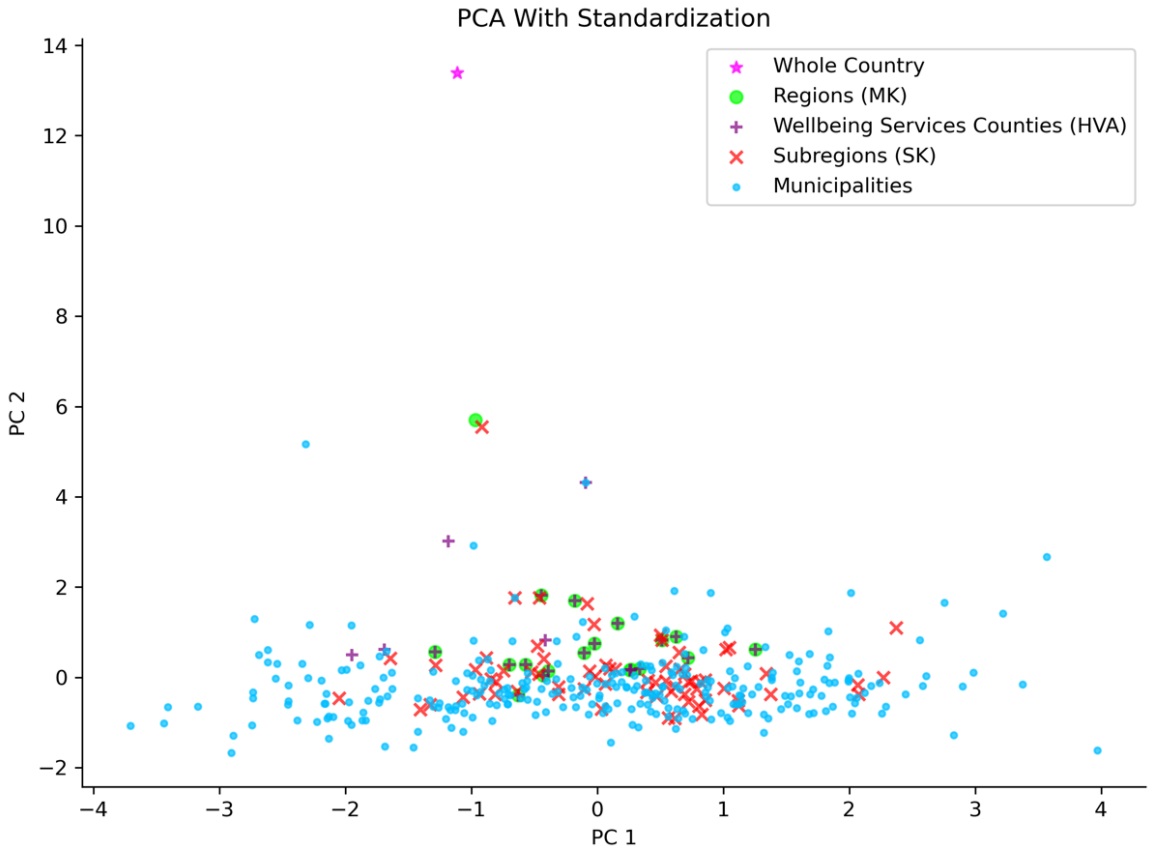


Figure 2: *PCA with standardization*

For ***Figure 1***, I use the method *PCA(n_components = 2)* and data *raw, non-standardized data (X)*. To visualize this PCA, I use configures as follows:

```
# Define marker styles and colors for categories
category_styles = {
    'Whole Country': ('magenta', '*'),
    'Regions (MK)': ('lime', 'o'),
    'Wellbeing Services Counties (HVA)': ('purple', '+'),
    'Subregions (SK)': ('red', 'x'),
    'Municipalities': ('deepskyblue', '.')
}
```

- Alpha: 0.7 for slight transparency
- Axes: Labeled as "*PC 1*" and "*PC 2*"
- Spines: Top and right spines removed for cleaner appearance
- Figure: *plt.subplots(figsize=(8, 6))*
- Save figure: *plt.savefig('pca_non_std.png', bbox_inches='tight', dpi=300)*

For ***Figure 2***, I use the method *PCA(n_components = 2)* and Standardized data (*X_scaled* using *StandardScaler()*).
To visualize this PCA, I use configures as follows:

- Marker styles and colors for categories: Same as the ***Figure 1***
- Alpha: 0.7 for slight transparency
- Axes: Labeled as "*PC 1*" and "*PC 2*"
- Spines: Top and right spines removed for cleaner appearance
- Figure: *plt.subplots(figsize=(8, 6))*
- Save figure: *plt.savefig('pca _std.png', bbox_inches='tight', dpi=300)*

PCA is a linear dimensionality reduction method which projects data onto principal components that maximize the variance. It will rotate the space such that data becomes maximally aligned with axes, then take an orthogonal projection. It assumes linear relationships between variables, which may oversimplify complex, nonlinear patterns in socioeconomic data like poverty rates and Gini coefficients. However, PCA is stable and computationally efficient method which provides a optimal way to visualize high-dimensional data in two dimensions.

Data preprocessing: We see that the ***Figure 1*** uses raw data which has variables with vastly different scales like "*Dwelling population, persons*" ranges from hundreds to millions, while "*At risk of poverty rate*" and "*Gini coefficient, gross income*" are percentages from 0% to 100%. PCA without standardization is dominated by the variable with largest variance, which is dwelling population in this case. This means that PC 1 is heavily influenced by population size, causing regions with large populations (such as "Whole Country" at approximately PC 1 ≈ 5e6) to be extreme outliers, while small municipalities cluster near PC 1 ≈ 0. This obscures patterns in poverty and inequality metrics, making it harder to interpret relationships relevant to SDG 10.
For ***Figure 2*** (with standardization), using *StandardScaler()* standardizes the variables to have zero mean and unit variance. This allows Gini coefficients and poverty rates to have more significant influence on projection, where regions and municipalities are more evenly distributed. This visualiztion gives a better plot where PC 1 and PC 2 reflect a close relationship between factors rather than just the population size in ***Figure 1***. This preprocessing step is really important for analysis of inequality metrics.

Initialization and Hyperparameters: PCA does not require ***initialization*** or ***hyperparameters*** such as random seeds because it is a deterministic method based on eigenvalue decomposition. The only parameter we need is *n_components = 2*, which reduces the data to two dimensions for visualization.

Description and Interpretation of Visualizations:
For ***Figure 1***, the scatter plot shows a wide spread along PC 1 (x-axis), ranging from 0 to about 5e6, with "Whole Country" (pink star) as a significant outlier at PC 1 ≈ 5e6. This reflects its large population (5,478,794 persons). Most municipalities (blue dots) cluster near PC 1 ≈ 0, with small variations along PC 2 (y-axis, ranging from -10 to 15). Regions (MK) (green circles), Wellbeing Services Counties (HVA) (purple pluses), and Subregions (SK) (red crosses) are scattered between PC 1 ≈ 0 and PC 1 ≈ 2e, with some outliers (a green circle at PC1 ≈ 1.8e5, likely MK01 Uusimaa with a population of 1,716,624). This visualization is not very good for SDG 10 because the population variable overshadows the inequality metrics. Smaller municipalities, despite having varying poverty rates, are compressed near the origin, making it hard to identify regions with high inequality or poverty. Progress toward SDG 10 cannot be effectively assessed here, as the plot does not highlight disparities in poverty or income inequality.

For ***Figure 2***, the scatter plot is more compact, with PC 1 ranging from -4 to 4 and PC2 from -2 to 14. "Whole Country" (pink star) is no longer an extreme outlier and is positioned at (PC 1 ≈ -1, PC 2 ≈ 13), suggesting its socioeconomic data (Gini = 33%, poverty rate = 13.4%) is distinct but not dominated by population. Municipalities (blue dots) are spread across the plot, forming a dense cluster around (PC 1 ≈ 0, PC 2 ≈ 0), with some outliers (a blue dot at PC 1 ≈ 4, possibly Kauniainen with a high Gini of 47.9% and low poverty rate of 5.9%). Regions (MK), Wellbeing Services Counties (HVA), and Subregions (SK) are interspersed among municipalities, with some clustering. This visualiztion reflects a combination of poverty rates and Gini coefficients, as standardization ensures all variables contribute equally.

The plot reveals significant variation in inequality and poverty across Finland. Municipalities such as Kauniainen (low poverty, high Gini) and Juuka (high poverty, moderate Gini) illustrate contrasting extremes, showing that inequality is not consistent. This suggests that while some regions have low poverty, they may still have high income inequality, which is a concern for SDG 10. Many municipalities cluster around the center (PC 1 ≈ 0, PC 2 ≈ 0), suggesting that a large number of regions have similar socioeconomic data (moderate poverty and Gini values). This indicates a baseline level of equality in many areas, but outliers highlight areas needing attention.

2. MDS on the non-standardized data with two different random seeds:

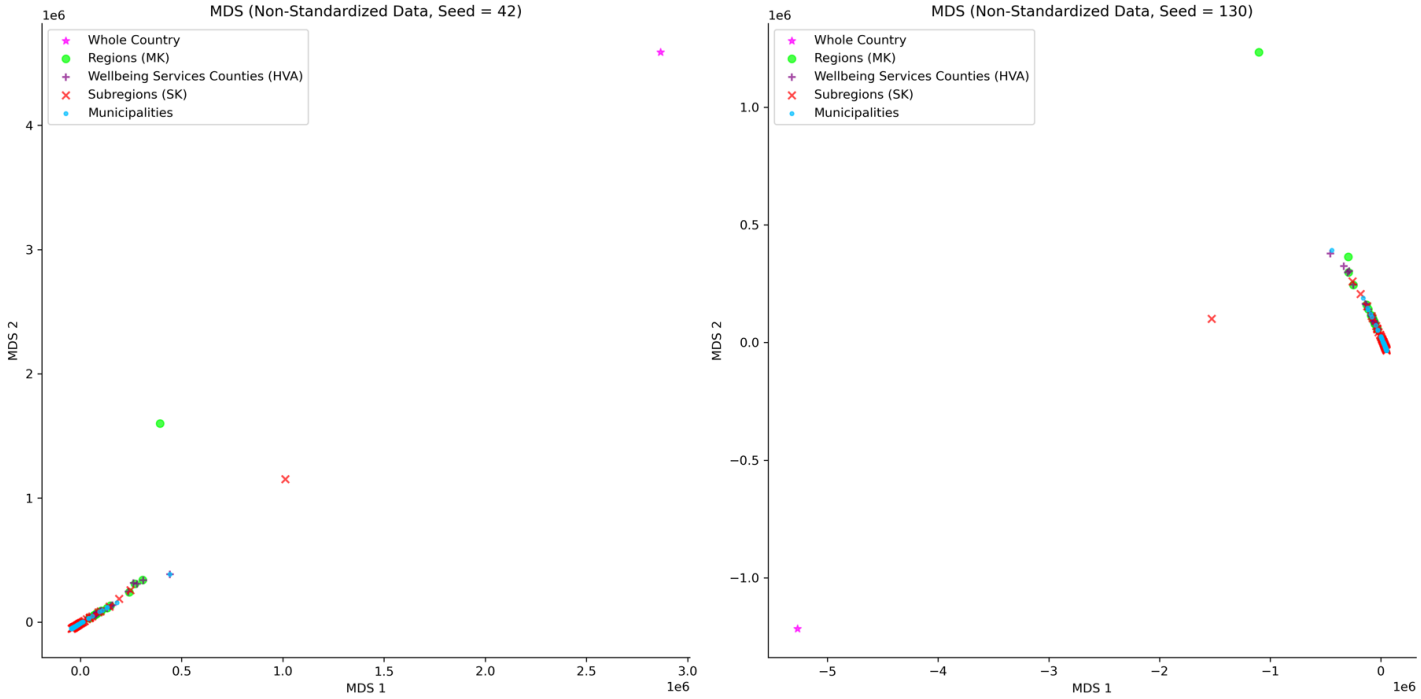


Figure 3: *MDS with random seed 42 and 130*

For *the left visualization* of ***Figure 3***, , I use the method *MDS(n_components = 2, random_state = 42)* and data *raw, non-standardized data (X)*. To visualize this PCA, I use configures as follows:

- Marker styles and colors for categories: Same as the ***Figure 1***
- Alpha: 0.7 for slight transparency
- Axes: Labeled as "*MDS 1*" and "*MDS 2*"
- Spines: Top and right spines removed for cleaner appearance
- Figure: *fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(16, 8))* (shared with the right plot)
- Save figure: *plt.savefig('mds_non_std_comparison.png', bbox_inches='tight', dpi=300)*

For *the right visualization* of ***Figure 3***, , I use the method *MDS(n_components = 2, random_state = 130)* and data *raw, non-standardized data (X)*. To visualize this PCA, I use configures as follows:

- Marker styles and colors for categories: Same as the ***Figure 1***
- Alpha: 0.7 for slight transparency
- Axes: Labeled as "*MDS 1*" and "*MDS 2*"
- Spines: Top and right spines removed for cleaner appearance
- Figure: *fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(16, 8))* (shared with the left plot)
- Save figure: *plt.savefig('mds_non_std_comparison.png', bbox_inches='tight', dpi=300)*

Multi-Dimensional Scaling (MDS) is a non-parametric dimensionality reduction technique that is used to approximate pairwise distances between points in the original high-dimensional space when projecting to a lower-dimensional space. Unlike PCA, which maximizes variance, MDS focuses on preserving the distances between data points and does not consider other types of relationships between data points. This means that MDS has limited scalability and layout is often dominated by large distances. This method is very sensitive to outliers and noise in the data, which can affect the interpretability of the projection. The goal of MDS is to minimize the stress function, which measures the difference between the distances in the original space and the distances in the lower-dimensional space. [1]

Data preprocessing: We see that the data is not standardized, meaning variables like "*Dwelling population, persons*" dominate the distance calculations over other variables like "*At risk of poverty rate*" and "*Gini coefficient, gross income*". This results in a visualization where the spread is driven by population size rather than socioeconomic factors. For instance, the "Whole Country" point is an extreme outlier in both plots (MDS 1 ≈ 3e6 in the left plot and MDS 1 ≈ -5e6 in the right one), showing its large population rather than its inequality metrics. Therefore, we find that standardization is an important step when applying MDS method.

Initialization and Hyperparameters: MDS uses an iterative numercial optimization algorithms (application of gradient descent) to minimize the stress function. The *random_state* parameter controls the initial configuration of data points, which can lead to different visualizations. In the left plot (seed = 42), the "Whole Country" point is at the far right (MDS 1 ≈ 3e6) while in the right plot (seed = 130), this data point is at the far left (MDS 1 ≈ -5e6). This indicates a reflection of the layout, but the relative distances between points should be preserved. The default settings for MDS such as *n_components = 2, Euclidean distance metric* are applied in this visualization. In general, while the relative positions of points are consistent (municipalities remain clustered, "Whole Country" remains an outlier), the orientation changes, which can affect visual interpretation.

Description and Interpretation of Visualizations:

For *the left visualization,* the scatter plot shows a wide spread along MDS1 (x-axis), ranging from 0 to 3e6, with "Whole Country" (pink star) as a significant outlier at MDS1 ≈ 3e6, reflecting its large population (5,478,794 persons). Municipalities (blue dots) cluster near the origin (MDS 1 ≈ 0 and MDS 2 ≈ 0), with a dense concentration and overlapping with other groups. Regions (MK) (green circles), Wellbeing Services Counties (HVA) (purple pluses), and Subregions (SK) (red crosses) varies between MDS 1 ≈ 0 and MDS 1 ≈ 0.5, with some outliers. The spread along MDS 1 is dominated by population size, with little variation along MDS 2 (y-axis, ranging from 0 to 4). This visualization is not very useful for assessing progress toward SDG 10 because the population

variable dominates the layout. Smaller municipalities are compressed near the origin, making it difficult to identify regions with high inequality or poverty.

For *the right visualization,* the scatter plot has MDS 1 ranging from -5e6 to 0. "Whole Country" (pink star) is now at MDS 1 ≈ -5e6, reflecting its large population. Municipalities (blue dots) cluster around MDS 1 ≈ 0, MDS 2 ≈ 0, with a similar dense concentration. The spread along MDS 1 is still driven by population, with MDS 2 ranging from -1 to 1, showing minimal variation in the second dimension. This visualization is relatively the same as the left one, since changing the random seed only changes the orientation. The population size continues to obscure the other patterns. The outlier does not provide any meaningful insight into national inequality trends in Finland because its location is influenced by population.

Therefore, both MDS plots fail to effectively highlight inequalities in poverty and income inequality due to the lack of standardization. A standardized approach (the previous PCA with standardization) is more appropriate for this analysis.

3. t-SNE on the non-standardized data with two different random seeds for each of three different values of the "perplexity" parameter:

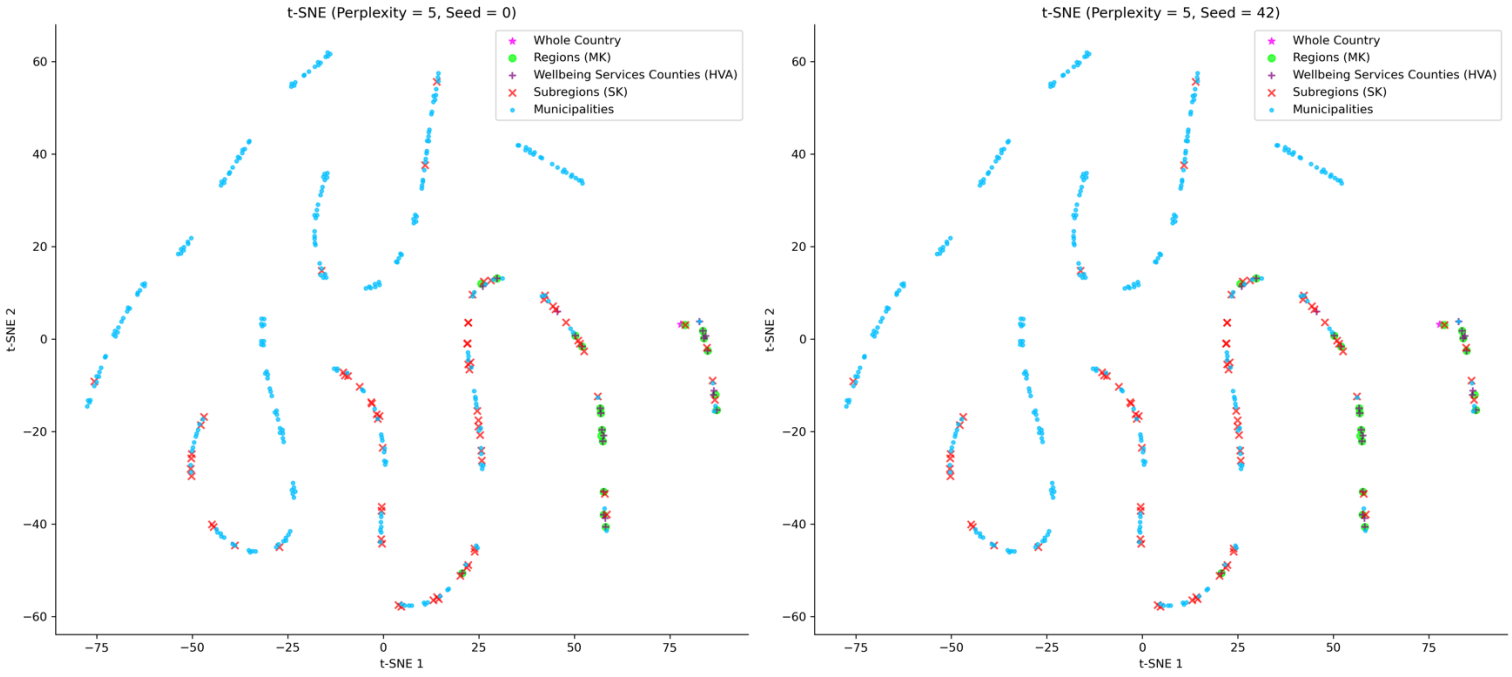


Figure 4: *t-SNE with perplexity 5 on seed 0 and 42*

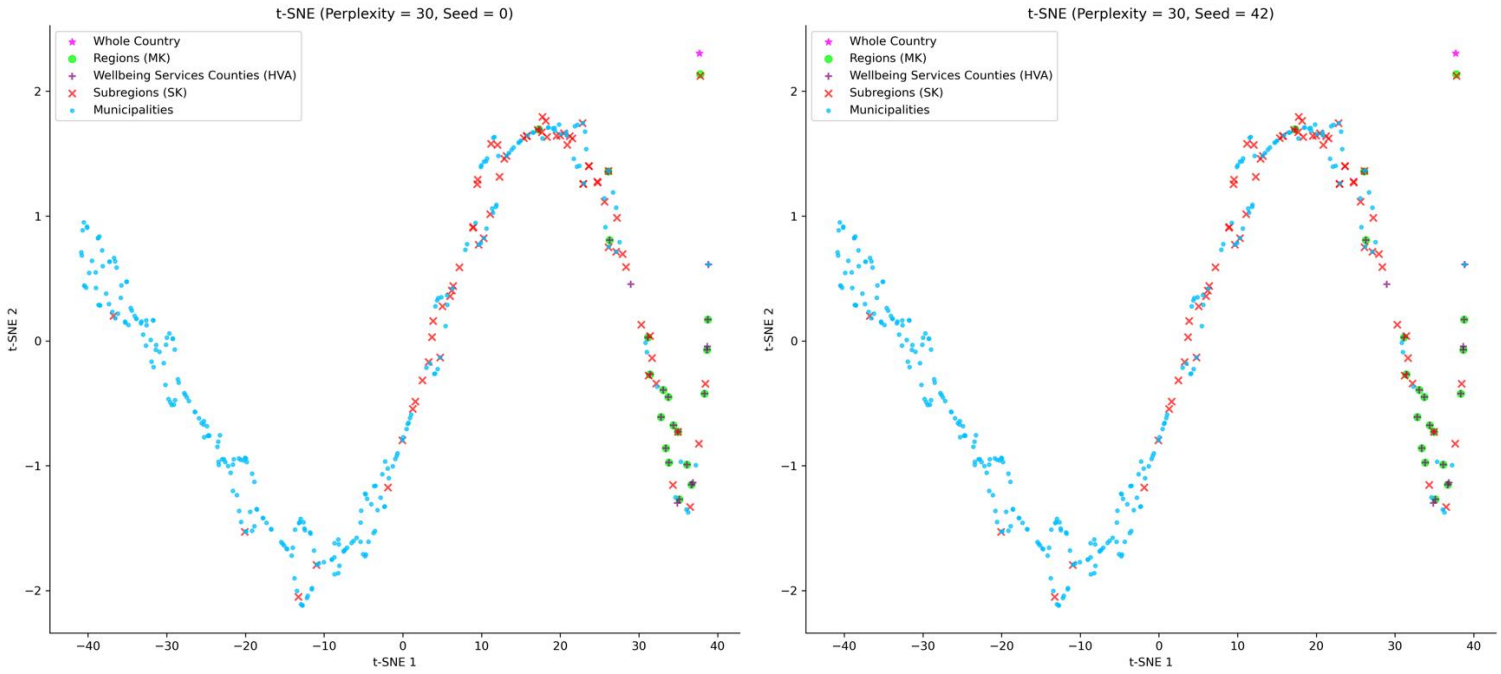


Figure 5: *t-SNE with perplexity 30 on seed 0 and 42*

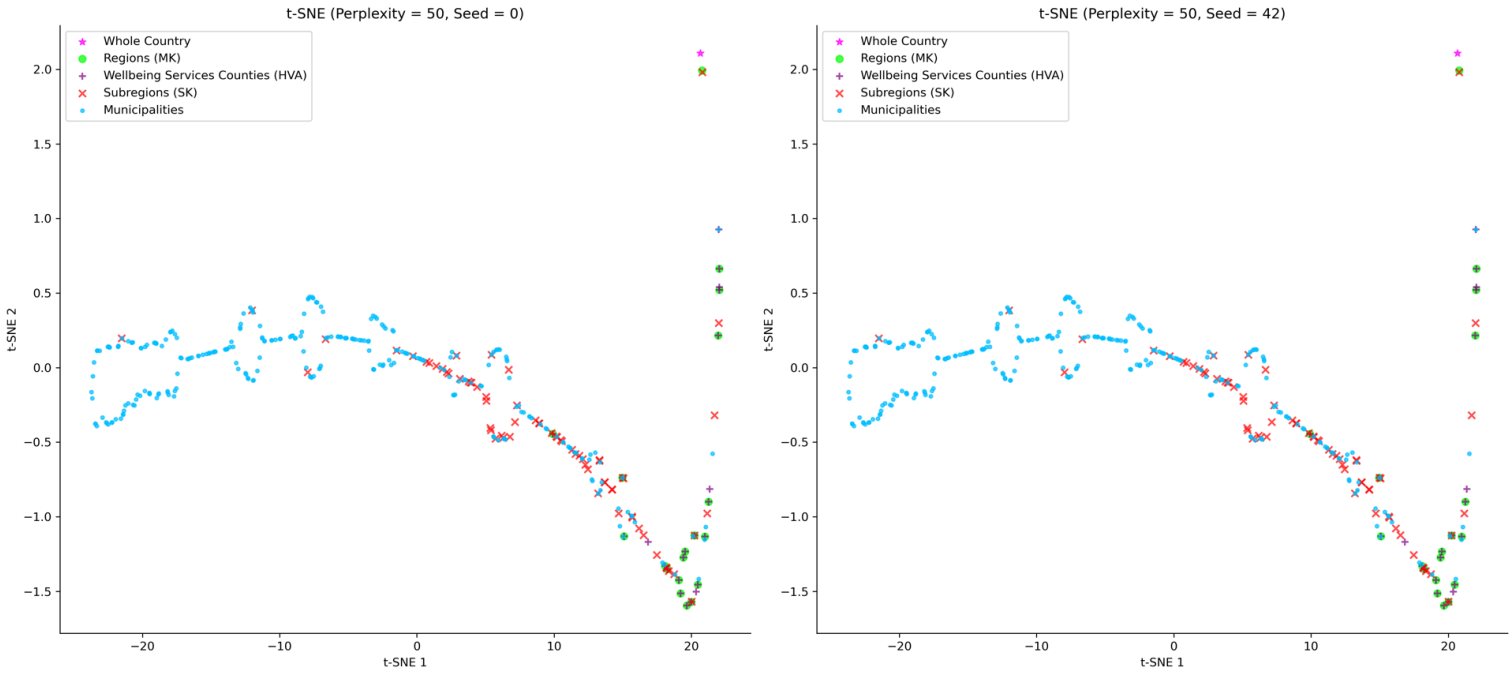


Figure 6: *t-SNE with perplexity 50 on seed 0 and 42*

Configurations:

- ***Figure 4***: Raw, non-standardized data (X)
- Left plot: *TSNE(n_components=2, perplexity=5, random_state=0, init='pca')*
- Right plot: *TSNE(n_components=2, perplexity=5, random_state=42, init='pca')*

Both plots apply the same marker styles and colors for categories, alpha, and spines as ***Figure 1***. Use figure size *plt.subplots(3, 2, figsize=(18, 24))* (shared across all subplots).

- ***Figure 5***: Raw, non-standardized data (X)
- Left plot: *TSNE(n_components=2, perplexity=30, random_state=0, init='pca')*
- Right plot: *TSNE(n_components=2, perplexity=30, random_state=42, init='pca')*

Both plots apply the same marker styles and colors for categories, alpha, and spines as ***Figure 1***. Use figure size *plt.subplots(3, 2, figsize=(18, 24))* (shared across all subplots).

- ***Figure 6***: Raw, non-standardized data (X)
- Left plot: *TSNE(n_components=2, perplexity=50, random_state=0, init='pca')*
- Right plot: *TSNE(n_components=2, perplexity=50, random_state=42, init='pca')*

Both plots apply the same marker styles and colors for categories, alpha, and spines as ***Figure 1***. Use figure size *plt.subplots(3, 2, figsize=(18, 24))* (shared across all subplots).

Method: ***t-Stochastic Neighborhood Embedding (t-SNE)*** is a non-parametric dimensionality reduction technique that is used to preserve local structure in the data, making it a good choice for visualizing clusters. The intuition of this method is that it converts high-dimensional distances into probabilities and minimizes the deviation between these similarity scores (KL divergence) in the lower-dimensional space. Unlike PCA (linear) or MDS (preserving distance), t-SNE is good at visualizing clusters but sacrifices global structure interpretability. Distances between clusters in t-SNE plots are not meaningful and the method is sensitive to hyperparameters "perplexity". [2]

Data preprocessing: We see that the data is not standardized, meaning variables like "*Dwelling population, persons*" dominate the distance calculations over other variables like "*At risk of poverty rate*" and "*Gini coefficient, gross income*". This causes t-SNE to prioritize population differences, potentially grouping regions by population size rather than socioeconomic factors. For example, municipalities with small populations (most < 20,000) form tight clusters, while larger regions ("Whole Country" with 5,478,794 persons) are often separated, even if their poverty or Gini values are similar.

Initialization: Applying *init='pca'* initializes the t-SNE embedding with a PCA projection, which provides a more stable starting point compared to random initialization. This reduces variability between runs with the same random seed and often leads to better convergence. However, the

non-standardized data still means that the initial PCA projection is dominated by population, influencing the final t-SNE layout.

Hyperparameters:

- ***Perplexity***: This parameter balances attention between local and global aspects of dataset. The parameter corresponds to a guess about the number of close neighbors each point has.

Perplexity = 5: A low perplexity focuses on local structure, leading to many small, tight clusters (***Figures 4***). This can over-emphasize small differences, dividing the data into numerous groups.

Perplexity = 30: A moderate perplexity balances local and global structure, often producing more cohesive clusters (***Figures 5***).

Perplexity = 50: A high perplexity emphasizes global structure, smoothing out smaller clusters and creating overall structures (***Figures 6***). This can obscure local patterns but better visualizes broader trends.

- ***Random seeds*** *(0 and 42)*: The random seed affects the initial placement of points. Different seeds lead to different layouts, but the clustering tendencies should be similar.

Description and Interpretation of Visualizations:

Figure 4: The left plot (seed = 0) shows several distinct, elongated clusters of municipalities (blue dots), with t-SNE 1 ranging from -75 to 75 and t-SNE 2 from -60 to 60. Subregions (SK) (red crosses) and Wellbeing Services Counties (HVA) (purple pluses) are interspersed within these clusters. Regions (MK) (green circles) are scattered across the plot, often near the edges of clusters. "Whole Country" (pink star) is at (t-SNE 1 ≈ 75, t-SNE2 ≈ 0), near the edges of clusters. The right plot (seed = 42) has the similar layout where the clustering pattern remains consistent, with municipalities forming tight, elongated groups. This visualization is not very informative for SDG 10 because the clustering obscures patterns in poverty and inequality. The fragmented clusters make it difficult to assess broader trends in inequality across regions.

Figure 5: Both plots shows fewer, more cohesive clusters compared to perplexity = 5, with t-SNE 1 ranging from -40 to 40 and t-SNE 2 from -2 to 2. Municipalities form elongated, curved structures, with Subregions (SK) and Wellbeing Services Counties (HVA) interspersed. Regions (MK) are more spread out, crowded around t-SNE ≈ 30 to t-SNE ≈ 40. This visualization provides slightly more insight than perplexity = 5, since the larger clusters might reflect some socioeconomic similarities within population-similar groups. However, the lack of standardization still limits its usefulness for assessing inequality gap.

Figure 6: Both plots shows fewer, more continuous structures, with t-SNE 1 ranging from -20 to 20 and t-SNE 2 from -1.5 to 2. Municipalities form a few large, curved clusters, with Subregions (SK) and Wellbeing Services Counties (HVA) interspersed. Regions (MK) are spread out at the local minimum of the visualization. High-Gini regions are not clearly separated from low-Gini regions within the clusters because population differences overshadow socioeconomic factors. This visualization provides the most global view of the three perplexity settings, but it is not good enough to highlight inequality patterns due to the lack of standardization. The continuous structures might reflect some broader trends, but they are primarily driven by population size.

These t-SNE visualizations are not informative for assessing progress toward SDG 10. The lack of standardization obscures patterns in poverty and income inequality, making it impossible to identify regions with high inequalities or track progress in reducing inequality. A standardized approach (PCA with standardization) is more appropriate for this analysis, since it would allow socioeconomic factors to drive the layout and reveal meaningful clusters or outliers related to inequality.

<u>Task 2</u>: Relational Data

This is my edge list:

	node1	node2
1	1	2
2	1	3
3	1	4
4	1	6
5	1	8
6	1	10
7	1	11
8	2	3
9	2	8
10	2	10
11	3	4
12	3	6
13	3	9
14	3	11
15	4	5
16	4	8
17	4	10
18	4	16
19	5	8
20	5	10
21	6	7
22	6	9
23	6	13
24	6	14
25	7	9
26	7	12
27	7	13
28	8	9
29	8	10
30	9	11
31	9	13
32	10	11
33	10	16
34	10	17
35	11	13
36	11	15
37	12	13
38	12	14
39	12	15
40	13	14
41	13	15
42	14	15
43	15	16
44	16	17

The network dataset is modeled by representing the 17 SDGs as the nodes in an undirected graph and deriving the edges based on personally felt interconnections between their objectives which results in 44 edges as an example of the above edge list. The first node (1 to 17) is allocated to each SDG, and in this way, their connections are modelled while ensuring that the connection exists in at least one other node: Moreover, such aspects as the SDG1 (No Poverty) interconnecting to the SDG 2 (Zero Hunger) are important because the poverty issue has an impact on the food which is the reason for hunger and to the SDG 8 (Decent Work and Economic Growth) as well, which in turn empowers the poor through economic opportunities. Also, the emission of the SDG 13 (Climate Action) leads to the SDG 14 (Life Below Water) and the SDG 15 (Life on Land) due to common environmental targets, while the education of the SDG 4 (Quality Education) links with the SDG 5 (Gender Equality) so that well-defined education can be used for the empowerment of gender equity. In addition, my selected SDG 10 (Reduced Inequalities) correlates with SDG 16 (Peace, Justice, and Strong Institutions) because inequality reduction is the core issue for peaceful and sustainable societies and, at the same time, needed for the promotion of peace and the functioning of institutions while peaceful societies with robust, inclusive institutions provide the framework to address and reduce inequalities effectively. The edges, in this study, are not given any weights so that the focus remains on the fact of whether the connections are there or not, which in turn makes sure that each SDG has one link at least while SDG 17 has to be connected to SDG 10 and 16 to show the feature of sustainable development. Following this rule, the procedure is indeed coherent and allows simultaneous coverage of all the SDGs.

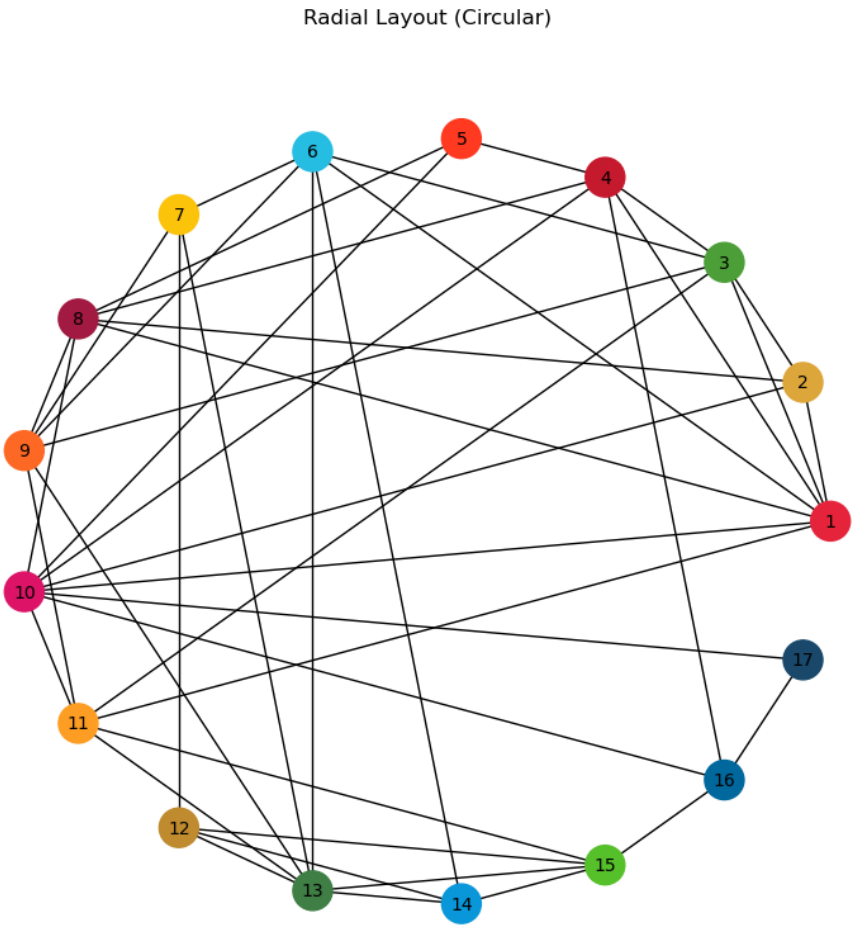
Visualization:

Shared Configurations:

- Undirected graph: *G = nx.Graph()*
- Node size: *500*
- Edge width: *1.0 (unweighted)*
- Labels: *SDG numbers from 1 to 17*
- Font size: *10*
- Figure size: *8×8*
- Node color: *Each unique color for each SDG*

```python
# Define SDG colors
sdg_colors = {
    1: "#E5243B",   # No Poverty
    2: "#DDA63A",   # Zero Hunger
    3: "#4C9F38",   # Good Health and Well-being
    4: "#C5192D",   # Quality Education
    5: "#FF3A21",   # Gender Equality
    6: "#26BDE2",   # Clean Water and Sanitation
    7: "#FCC30B",   # Affordable and Clean Energy
    8: "#A21942",   # Decent Work and Economic Growth
    9: "#FD6925",   # Industry, Innovation, and Infrastructure
    10: "#DD1367",  # Reduced Inequalities
    11: "#FD9D24",  # Sustainable Cities and Communities
    12: "#BF8B2E",  # Responsible Consumption and Production
    13: "#3F7E44",  # Climate Action
    14: "#0A97D9",  # Life Below Water
    15: "#56C02B",  # Life on Land
    16: "#00689D",  # Peace, Justice, and Strong Institutions
    17: "#19486A"   # Partnerships for the Goals
}
```

1. **A radial layout with numerical node ordering**.

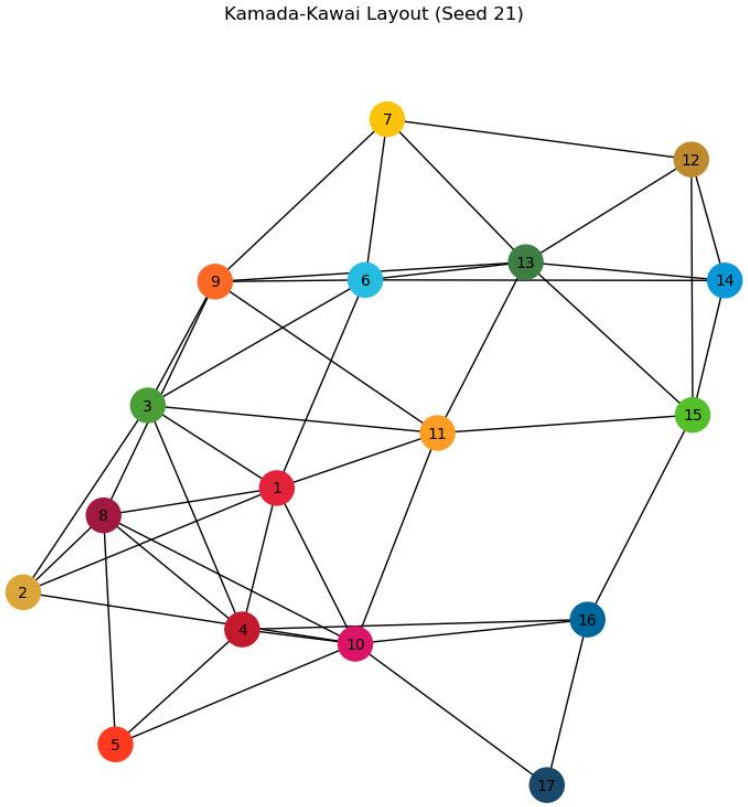


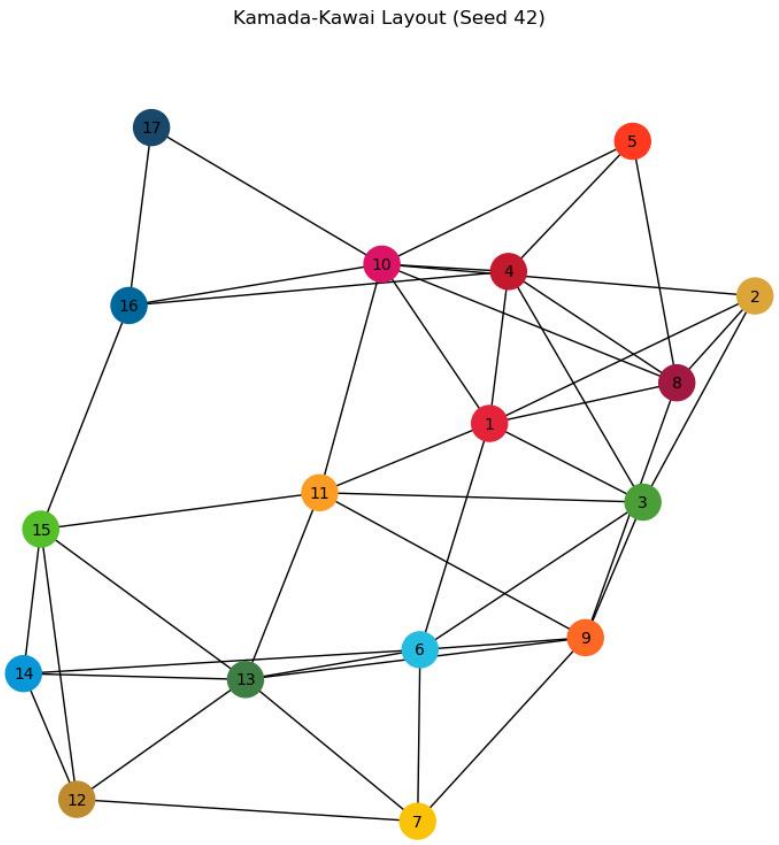
Configuration:

- Layout: *nx.circular_layout(G)*
- Title: *plt.title("Radial Layout (Circular)", pad=20)*

The radial layout graph is representing the interconnection of the 17 SDGs through a case of the 17 nodes, each colored differently corresponding to a particular SDG (red for SDG 1 (No Poverty), orange for SDG 5 (Gender Equality), or blue for SDG 14 (Life Below Water)). It is observed from the visualisation that node 10 (Reduced Inequalities) is the one that is mostly connected to other edges. Some connections are so dense that the nodes are no longer distinguishable and the overlapping is highlighting the complex interconnectedness of the goals. The majority of the nodes are strongly positively correlated with other nodes in the context of sustainable development, but there are a few that are not so intertwined with the rest. It is easy to identify the different parts, which are supposed to represent the progress achieved by different development component . Thus, the graph, though not directly representing the color mapping, is still very effective in that it makes it vividly clear that the accomplishment of the SDGs calls for a united effort across all dimensions so as to discuss their mutual dependencies and to secure balanced progress.

2. A Kamada-Kawai layout with two different random seeds.



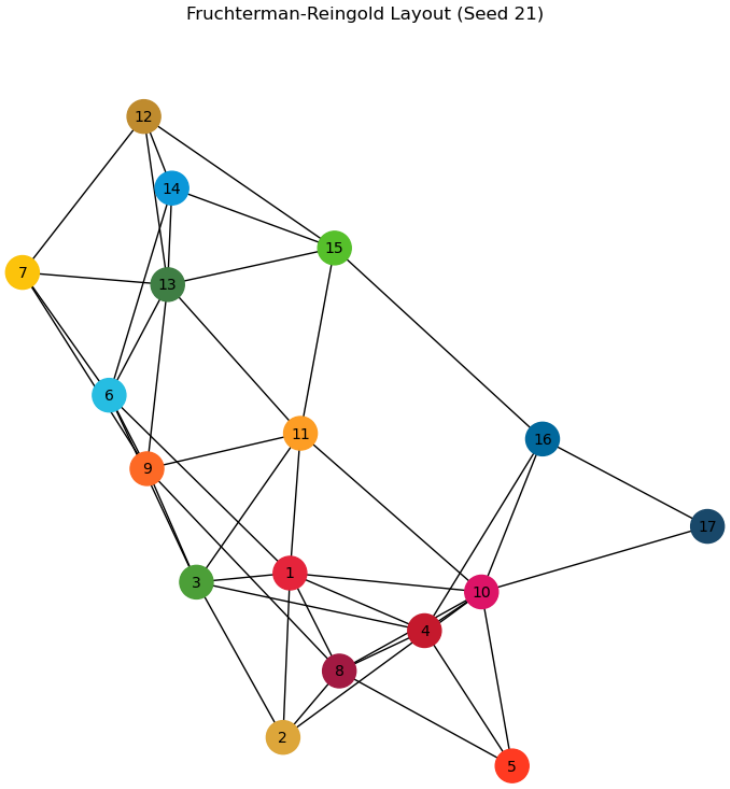


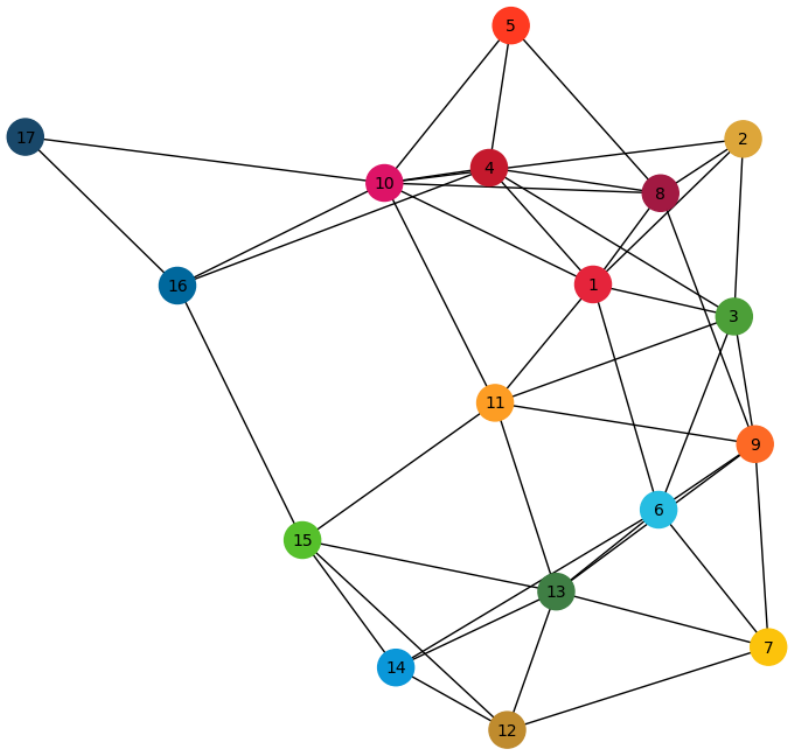
Configuration:

- Layout: *nx.kamada_kawai_layout(G)*
- Random seed: *np.random.seed(seed)* where seed = 21 and seed = 42
- Title: *plt.title(f"Kamada-Kawai Layout (Seed {seed})", pad=20)*

The Kamada-Kawai layout with different seeds gives us different visualizations. The relationship structure between the nodes is the same in both visualizations. However, the orientation is different: for example, in the visualization with seed 21, the nodes 2, 5, 17 are at the bottom of the layout, but they are at the top of the layout with seed 42. This type of layout will approximate graph-theoretic distance through geodesic distance. From both visualizations, we see that the nodes with the most interconnected nodes will be located in the centre of the layout while the nodes with the least interconnected ones will be located far away from the centre. It is easy to identify the different parts, which are supposed to represent the progress achieved by different development component. In general, The Kamada-Kawai layout with different seeds gives us a good overview of the close relationship between SDGs.

3. A Fruchterman-Reingold layout with two different random seeds.





Configuration:

- Layout: *nx.fruchterman_reingold_layout(G)*
- Random seed: *np.random.seed(seed)* where seed = 21 and seed = 100
- Title: *plt.title(f"Fruchterman-Reingold Layout (Seed {seed})", pad=20)*

The Fruchterman-Reingold layout with two different random seeds gives us different graphs. The relationship structure between the nodes is the same in both visualizations. However, the orientation is different: for example, in the visualization with seed 21, the nodes 2, 5, 17 are at the bottom of the layout, but they are at the top of the layout with seed 100. This type of layout will apply the following characteristics: adjacent nodes attract one another, non-adjacent nodes repel one another. In both cases, the nodes with the most interconnected nodes will be located in the centre of the layout while the nodes with the least interconnected ones will be located far away from the centre. This feature indicates the importance of objectives of some specific SDGs (such as SDG 1 (No Poverty)) compared to another SDGs. It is easy to identify the different parts, which are supposed to represent the progress achieved by different development component. In general, The Fruchterman-Reingold layout with different seeds gives us a interesting overview of the close relationship between SDGs.

The impact of methodological choices:

The methodological choices are indeed the factors by which the viewers are influenced in seeing the relationships between the SDGs. The radial layout represents the equality among the goals while it hinders the structural relationships between SDGs. On the other hand, the Kamada-Kawai and Fruchterman-Reingold layouts are the best with respect to the network structure which are employing a force-directed layout. The level of connectedness of the related SDGs is better in these layouts. Moreover, changing random seeds can result in different clusters to a greater or lesser extent. Withdrawing weights from the edges and using different colors for the SDGs reduce the complexity of the plot thus making the graph more intuitive; however, this might make the weaker connections seems stronger than necessary and the differences in the relationship strength might not be visible. Hyperparameters such as node size and edge witdth are also of great importance as they influence the readability of the plot which can be achieved if the size of the nodes and the edges is greater as well as the edges are uniform and thereby giving the reader clear details. Therefore, all these decisions work together to make the interpretability and complexity even; however, new biases that the decisions bring are that people tend to overrate the central or visually prominent SDGs (SDG 1 in Fruchterman-Reingold) and ignore the less important (SDG 17) and some may not realize that the variability in the layouts and the lack of edge weights may lead to them missing some of the important connections.

The challenges for both Task 1 and Task 2:

Both two tasks of this assignment are very challenging but very interesting. Especially, task 1 is a bit complicated when I have to analyze each figure to identify the difference between each dimensionality reduction methods with different random seeds. Moreover, finding suitable dataset is a bit time-consuming because of the task's requirements. I have to find dataset which is lare enough to visualize PCA, MDS, and t-SNE because if the dataset is not large enough, the visualizations are not good and we cannot analyze the trends and big picture. However, fortunately, after searching for lots of sources, I have found a reliable dataset to visualize these dimensionality reduction techniques. After that I have to analyze the dataset to understand its feature, then categorize them into suitable groups to visualize. This process is quite demanding but the companion code base helps me a lot. Task 2 is easier because I do not have to find dataset but I have to create a CSV file manually. However, this process is very interesting because I can create my own dataset based on my perception without cleaning the dataset. Finding suitable colors for each node is a bit difficult but Youtube tutorial helps me a lot. Moreover, the companion code base is very helpful because it guides me how to create different networks based on different layouts. In general, this final assignment is very demanding but very interesting, which makes me learn lots of useful knowledge about visualizing relational data and dimensionality reduction methods.

References:

[1] Geeksforgeeks. (Jan 02, 2023) '**Multidimensional Scaling (MDS) using Scikit Learn**'. Available at: https://www.geeksforgeeks.org/multidimensional-scaling-mds-using-scikit-learn/

[2] Distill. (Oct. 18, 2016) '**How to Use t-SNE Effectively**'. Available at: https://distill.pub/2016/misread-tsne/