CODE EXPLANATION

This code is meant to analyze and visualize the degree distribution of a given network using the NetworkX library in Python. The degree distribution of a network provides insights into the distribution of node degrees (the number of connections each node has). The code also visualizes the complementary cumulative degree distribution (CCDF) of the network.

Here's a breakdown of the code step by step:

1. **Importing Necessary Libraries:**
   - `networkx` (imported as `nx`): A library used for creating, analyzing, and visualizing complex networks.
   - `os`: A module providing functions for interacting with the operating system.
   - `openpyxl`: A library for working with Excel files.
   - `get_column_letter`: A function from `openpyxl.utils` to convert column indices to letters in Excel notation.
   - `matplotlib.pyplot` (imported as `plt`): A plotting library for creating visualizations.
   - `numpy` (imported as `np`): A library for numerical computations.

2. **Loading Network Data:**
   The code contains commented lines to load network data from different files using the `read_pajek` function. You can choose one of them based on the network you want to analyze. A Pajek file is commonly used to store network data.

3. **Degree Sequence and Counting:**
   - `degree_sequence`: A sorted list of degrees of all nodes in the network.
   - `degree_count`: An array to store the count of nodes with each degree.
   - The `for` loop iterates through each degree in the sorted `degree_sequence` and increments the corresponding entry in `degree_count`. This creates a histogram-like representation of the degree distribution.

4. **Degree Probability Distribution Function (PDF) Plot:**
   - The code uses `matplotlib` to create a histogram of the degree distribution.
   - `plt.hist` creates a histogram with `degree_sequence` data, using bins from 0 to the maximum degree.
   - `density=True` normalizes the histogram to create a probability density function (PDF).
   - Axes labels and titles are added using `plt.xlabel`, `plt.ylabel`, and `plt.title`.
   - `plt.show()` displays the plot.

5. **Complementary Cumulative Degree Distribution (CCDF) Calculation:**
   - `degree_ccdf`: An array to store the complementary cumulative distribution function values.
   - The second `for` loop iterates through each degree in the sorted `degree_sequence` and increments the corresponding and all higher-degree entries in `degree_ccdf`.
   - After the loop, the `degree_ccdf` array is normalized.

6. **CCDF Log-log Plot:**
   - The code uses a log-log plot to visualize the CCDF.
   - `plt.loglog` plots the `degree_ccdf` data against the degrees using blue dots ('b.').
   - Axes labels and title are added similar to the previous plot.
   - `plt.show()` displays the plot.

In summary, this code takes a network (which needs to be uncommented from the provided paths or substituted with your own data), calculates the degree distribution and CCDF, and then creates two plots: one for the degree distribution as a PDF histogram and another for the CCDF on a log-log scale. These visualizations provide insights into the structural characteristics of the network.