```
[22]: def stdNBgraph(dataset):
          #converted standard Normal Distribution
          import seaborn as sns
          import numpy as np
          mean=dataset.mean()
          std=dataset.std()
          values=[i for i in dataset]
          z_score=[((j-mean)/std) for j in values]
          sns.displot(z_score,kde=True)

          sum(z_score)/len(z_score)
          #z_score.std()

      # Compute the mean and standard deviation of the z-scores
          z_score_mean = np.mean(z_score)
          z_score_std = np.std(z_score)

          print(f"Mean of z-scores: {z_score_mean}")
          print(f"Standard deviation of z-scores: {z_score_std}")
```

Here's a line-by-line explanation of the stdNBgraph function, along with a summary at the end:

Code Explanation:

1. **Import Libraries:**

```
python

import seaborn as sns

import numpy as np
```

   - Imports the Seaborn library (sns) for creating statistical graphics, and NumPy (np) for numerical operations.

2. **Calculate Mean and Standard Deviation:**

```
python

mean = dataset.mean()

std = dataset.std()
```

   - Computes the mean (average) and standard deviation (a measure of dispersion) of the dataset. These are used to standardize the data.

3. **Create List of Values:**

```
python

values = [i for i in dataset]
```

- Converts the dataset into a list of values. This step is not strictly necessary because you can perform operations directly on a Pandas Series, but it ensures that values is a list.

4. **Compute Z-scores:**

```
python

z_score = [((j - mean) / std) for j in values]
```

- Computes the z-score for each value in the dataset. The z-score is calculated as (value - mean) / standard deviation, which standardizes the data by subtracting the mean and dividing by the standard deviation.

5. **Plot Z-scores Distribution:**

```
python

sns.displot(z_score, kde=True)
```

- Uses Seaborn's displot function to create a histogram of the z-scores. The kde=True argument adds a Kernel Density Estimate (KDE) curve, which smooths out the histogram to show the distribution of z-scores.

6. **Compute Mean of Z-scores:**

```
python

sum(z_score) / len(z_score)
```

- Calculates the mean of the z-scores. This line computes the average of the z-scores but does not store or use the result. It's more of a placeholder in this context.

7. **Compute and Print Mean and Standard Deviation of Z-scores:**

```
python

z_score_mean = np.mean(z_score)

z_score_std = np.std(z_score)


print(f"Mean of z-scores: {z_score_mean}")

print(f"Standard deviation of z-scores:
{z_score_std}")
```

- np.mean(z_score): Calculates the mean of the z-scores using NumPy.
- np.std(z_score): Calculates the standard deviation of the z-scores using NumPy.
- The print statements output these values. The mean should be close to 0, and the standard deviation should be close to 1 if the z-scores are correctly computed.

Summary:

The stdNBgraph function standardizes a dataset by calculating z-scores, which are then used to plot the distribution using Seaborn. It computes and prints the mean and standard deviation of these z-scores. The mean should be around 0, and the standard deviation should be around 1, verifying the standardization process.