In [20]:
```
#1BM22AI037
#LAB 3
#handling missing data transformation write a python program to handle missing data using
#multiple techniques such as mean,median,mode,forward and backward fill.
#apply data transformation methods such as min-max scaling and z score normalization
#analyze how different techniques affect the data set
import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler, StandardScaler
data={
    'A':[1,2,np.nan,4,5,np.nan,7,8,9,10,11,np.nan,13,14,15],
    'B':[np.nan,1,2,3,np.nan,5,6,7,8,np.nan,10,11,12,np.nan,14],
    'C':[1,2,3,4,5,6,np.nan,8,9,10,np.nan,12,13,14,15],
    'D':[1,2,3,4,5,6,7,8,np.nan,np.nan,np.nan,np.nan,np.nan,np.nan,np.nan],
    'G':[np.nan,np.nan,np.nan,np.nan,np.nan,np.nan,np.nan,np.nan,np.nan,np.nan,np.nan,np.nan,np.nan,np.nan,np.nan]
}
df=pd.DataFrame(data)
mean_filled=df.fillna(df.mean())
median_filled=df.fillna(df.median())
mode_filled=df.fillna(df.mode().iloc[0])
forward_filled=df.fillna(method='ffill')
backward_filled=df.fillna(method='bfill')
scaler_minmax=MinMaxScaler()
scaler_zscore=StandardScaler()
mean_filled_scaled=pd.DataFrame(scaler_minmax.fit_transform(mean_filled),columns=mean_filled.columns)
median_filled_scaled=pd.DataFrame(scaler_minmax.fit_transform(median_filled),columns=median_filled.columns)
mode_filled_scaled=pd.DataFrame(scaler_minmax.fit_transform(mode_filled),columns=mode_filled.columns)
forward_filled_scaled=pd.DataFrame(scaler_minmax.fit_transform(forward_filled),columns=forward_filled.columns)
backward_filled_scaled=pd.DataFrame(scaler_minmax.fit_transform(backward_filled),columns=backward_filled.columns)
mean_filled_zscore=pd.DataFrame(scaler_zscore.fit_transform(mean_filled),columns=mean_filled.columns)
median_filled_zscore=pd.DataFrame(scaler_zscore.fit_transform(median_filled),columns=median_filled.columns)
mode_filled_zscore=pd.DataFrame(scaler_zscore.fit_transform(mode_filled),columns=mode_filled.columns)
forward_filled_zscore=pd.DataFrame(scaler_zscore.fit_transform(forward_filled),columns=forward_filled.columns)
backward_filled_zscore=pd.DataFrame(scaler_zscore.fit_transform(backward_filled),columns=backward_filled.columns)
results={
    'Mean Filled':mean_filled_scaled,
    'Median Filled':median_filled_scaled,
    'Mode Filled':mode_filled_scaled,
    'Forward Filled':forward_filled_scaled,
    'Backward Filled':backward_filled_scaled,
    'Mean Z-Score':mean_filled_zscore,
    'Median Z-Score':median_filled_zscore,
```

```
    'Mode Z-Score':mode_filled_zscore,
    'Forward Z-Score':forward_filled_zscore,
    'Backward Z-Score':backward_filled_zscore
}
for name,result in results.items():
    print(f"{name}:\n{result}\n")
```

```
Mean Filled:
           A         B         C         D   G
0   0.000000  0.475524  0.000000  0.000000 NaN
1   0.071429  0.000000  0.071429  0.142857 NaN
2   0.517857  0.076923  0.142857  0.285714 NaN
3   0.214286  0.153846  0.214286  0.428571 NaN
4   0.285714  0.475524  0.285714  0.571429 NaN
5   0.517857  0.307692  0.357143  0.714286 NaN
6   0.428571  0.384615  0.489011  0.857143 NaN
7   0.500000  0.461538  0.500000  1.000000 NaN
8   0.571429  0.538462  0.571429  0.500000 NaN
9   0.642857  0.475524  0.642857  0.500000 NaN
10  0.714286  0.692308  0.489011  0.500000 NaN
11  0.517857  0.769231  0.785714  0.500000 NaN
12  0.857143  0.846154  0.857143  0.500000 NaN
13  0.928571  0.475524  0.928571  0.500000 NaN
14  1.000000  1.000000  1.000000  0.500000 NaN

Median Filled:
           A         B         C         D   G
0   0.000000  0.461538  0.000000  0.000000 NaN
1   0.071429  0.000000  0.071429  0.142857 NaN
2   0.535714  0.076923  0.142857  0.285714 NaN
3   0.214286  0.153846  0.214286  0.428571 NaN
4   0.285714  0.461538  0.285714  0.571429 NaN
5   0.535714  0.307692  0.357143  0.714286 NaN
6   0.428571  0.384615  0.500000  0.857143 NaN
7   0.500000  0.461538  0.500000  1.000000 NaN
8   0.571429  0.538462  0.571429  0.500000 NaN
9   0.642857  0.461538  0.642857  0.500000 NaN
10  0.714286  0.692308  0.500000  0.500000 NaN
11  0.535714  0.769231  0.785714  0.500000 NaN
12  0.857143  0.846154  0.857143  0.500000 NaN
13  0.928571  0.461538  0.928571  0.500000 NaN
14  1.000000  1.000000  1.000000  0.500000 NaN

Mode Filled:
           A         B         C         D   G
0   0.000000  0.000000  0.000000  0.000000 NaN
1   0.071429  0.000000  0.071429  0.142857 NaN
2   0.000000  0.076923  0.142857  0.285714 NaN
3   0.214286  0.153846  0.214286  0.428571 NaN
4   0.285714  0.000000  0.285714  0.571429 NaN
5   0.000000  0.307692  0.357143  0.714286 NaN
```

```
6   0.428571  0.384615  0.000000  0.857143 NaN
7   0.500000  0.461538  0.500000  1.000000 NaN
8   0.571429  0.538462  0.571429  0.000000 NaN
9   0.642857  0.000000  0.642857  0.000000 NaN
10  0.714286  0.692308  0.000000  0.000000 NaN
11  0.000000  0.769231  0.785714  0.000000 NaN
12  0.857143  0.846154  0.857143  0.000000 NaN
13  0.928571  0.000000  0.928571  0.000000 NaN
14  1.000000  1.000000  1.000000  0.000000 NaN

Forward Filled:
          A         B         C         D   G
0   0.000000       NaN  0.000000  0.000000 NaN
1   0.071429  0.000000  0.071429  0.142857 NaN
2   0.071429  0.076923  0.142857  0.285714 NaN
3   0.214286  0.153846  0.214286  0.428571 NaN
4   0.285714  0.153846  0.285714  0.571429 NaN
5   0.285714  0.307692  0.357143  0.714286 NaN
6   0.428571  0.384615  0.357143  0.857143 NaN
7   0.500000  0.461538  0.500000  1.000000 NaN
8   0.571429  0.538462  0.571429  1.000000 NaN
9   0.642857  0.538462  0.642857  1.000000 NaN
10  0.714286  0.692308  0.642857  1.000000 NaN
11  0.714286  0.769231  0.785714  1.000000 NaN
12  0.857143  0.846154  0.857143  1.000000 NaN
13  0.928571  0.846154  0.928571  1.000000 NaN
14  1.000000  1.000000  1.000000  1.000000 NaN

Backward Filled:
          A         B         C         D   G
0   0.000000  0.000000  0.000000  0.000000 NaN
1   0.071429  0.000000  0.071429  0.142857 NaN
2   0.214286  0.076923  0.142857  0.285714 NaN
3   0.214286  0.153846  0.214286  0.428571 NaN
4   0.285714  0.307692  0.285714  0.571429 NaN
5   0.428571  0.307692  0.357143  0.714286 NaN
6   0.428571  0.384615  0.500000  0.857143 NaN
7   0.500000  0.461538  0.500000  1.000000 NaN
8   0.571429  0.538462  0.571429       NaN NaN
9   0.642857  0.692308  0.642857       NaN NaN
10  0.714286  0.692308  0.785714       NaN NaN
11  0.857143  0.769231  0.785714       NaN NaN
12  0.857143  0.846154  0.857143       NaN NaN
13  0.928571  1.000000  0.928571       NaN NaN
```

```
14  1.000000  1.000000  1.000000       NaN NaN
```

```
Mean Z-Score:
           A         B         C         D   G
0  -1.834610  0.000000 -1.614574 -2.091650 NaN
1  -1.581561 -1.776481 -1.378738 -1.494036 NaN
2   0.000000 -1.489109 -1.142901 -0.896421 NaN
3  -1.075461 -1.201737 -0.907064 -0.298807 NaN
4  -0.822412  0.000000 -0.671227  0.298807 NaN
5   0.000000 -0.626993 -0.435391  0.896421 NaN
6  -0.316312 -0.339621  0.000000  1.494036 NaN
7  -0.063262 -0.052249  0.036283  2.091650 NaN
8   0.189787  0.235122  0.272119  0.000000 NaN
9   0.442837  0.000000  0.507956  0.000000 NaN
10  0.695887  0.809866  0.000000  0.000000 NaN
11  0.000000  1.097238  0.979629  0.000000 NaN
12  1.201986  1.384610  1.215466  0.000000 NaN
13  1.455036  0.000000  1.451303  0.000000 NaN
14  1.708085  1.959354  1.687139  0.000000 NaN
```

```
Median Z-Score:
           A         B         C         D   G
0  -1.846672 -0.038306 -1.619289 -2.091650 NaN
1  -1.593703 -1.762077 -1.383470 -1.494036 NaN
2   0.050594 -1.474782 -1.147651 -0.896421 NaN
3  -1.087766 -1.187487 -0.911832 -0.298807 NaN
4  -0.834797 -0.038306 -0.676014  0.298807 NaN
5   0.050594 -0.612896 -0.440195  0.896421 NaN
6  -0.328859 -0.325601  0.031442  1.494036 NaN
7  -0.075891 -0.038306  0.031442  2.091650 NaN
8   0.177078  0.248989  0.267261  0.000000 NaN
9   0.430047 -0.038306  0.503080  0.000000 NaN
10  0.683016  0.823580  0.031442  0.000000 NaN
11  0.050594  1.110875  0.974717  0.000000 NaN
12  1.188953  1.398170  1.210536  0.000000 NaN
13  1.441922 -0.038306  1.446355  0.000000 NaN
14  1.694890  1.972760  1.682174  0.000000 NaN
```

```
Mode Z-Score:
           A         B         C         D   G
0  -1.183263 -1.024440 -1.226682 -0.771845 NaN
1  -0.979252 -1.024440 -1.019938 -0.358357 NaN
2  -1.183263 -0.798461 -0.813194  0.055132 NaN
3  -0.571230 -0.572481 -0.606450  0.468620 NaN
```

```
4  -0.367220 -1.024440 -0.399705  0.882109 NaN
5  -1.183263 -0.120522 -0.192961  1.295597 NaN
6   0.040802  0.105457 -1.226682  1.709085 NaN
7   0.244813  0.331437  0.220527  2.122574 NaN
8   0.448824  0.557416  0.427271 -0.771845 NaN
9   0.652835 -1.024440  0.634016 -0.771845 NaN
10  0.856846  1.009375 -1.226682 -0.771845 NaN
11 -1.183263  1.235355  1.047504 -0.771845 NaN
12  1.264867  1.461334  1.254248 -0.771845 NaN
13  1.468878 -1.024440  1.460992 -0.771845 NaN
14  1.672889  1.913293  1.667736 -0.771845 NaN

Forward Z-Score:
           A         B         C         D   G
0  -1.550804       NaN -1.595797 -2.122574 NaN
1  -1.322745 -1.575453 -1.363399 -1.709085 NaN
2  -1.322745 -1.324813 -1.131002 -1.295597 NaN
3  -0.866626 -1.074172 -0.898604 -0.882109 NaN
4  -0.638566 -1.074172 -0.666206 -0.468620 NaN
5  -0.638566 -0.572892 -0.433809 -0.055132 NaN
6  -0.182448 -0.322252 -0.433809  0.358357 NaN
7   0.045612 -0.071611  0.030986  0.771845 NaN
8   0.273671  0.179029  0.263384  0.771845 NaN
9   0.501731  0.179029  0.495782  0.771845 NaN
10  0.729790  0.680309  0.495782  0.771845 NaN
11  0.729790  0.930949  0.960577  0.771845 NaN
12  1.185909  1.181590  1.192974  0.771845 NaN
13  1.413968  1.181590  1.425372  0.771845 NaN
14  1.642028  1.682870  1.657769  0.771845 NaN

Backward Z-Score:
           A         B         C         D   G
0  -1.677294 -1.453222 -1.634398 -1.527525 NaN
1  -1.444336 -1.453222 -1.405276 -1.091089 NaN
2  -0.978421 -1.221325 -1.176155 -0.654654 NaN
3  -0.978421 -0.989428 -0.947034 -0.218218 NaN
4  -0.745464 -0.525634 -0.717913  0.218218 NaN
5  -0.279549 -0.525634 -0.488792  0.654654 NaN
6  -0.279549 -0.293736 -0.030549  1.091089 NaN
7  -0.046591 -0.061839 -0.030549  1.527525 NaN
8   0.186366  0.170058  0.198572       NaN NaN
9   0.419323  0.633852  0.427693       NaN NaN
10  0.652281  0.633852  0.885935       NaN NaN
11  1.118196  0.865749  0.885935       NaN NaN
```

```
12  1.118196  1.097647  1.115056       NaN NaN
13  1.351153  1.561441  1.344177       NaN NaN
14  1.584111  1.561441  1.573299       NaN NaN
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\preprocessing\_data.py:461: RuntimeWarning: All-NaN slice encountered
  data_min = np.nanmin(X, axis=0)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\preprocessing\_data.py:462: RuntimeWarning: All-NaN slice encountered
  data_max = np.nanmax(X, axis=0)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\preprocessing\_data.py:461: RuntimeWarning: All-NaN slice encountered
  data_min = np.nanmin(X, axis=0)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\preprocessing\_data.py:462: RuntimeWarning: All-NaN slice encountered
  data_max = np.nanmax(X, axis=0)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\preprocessing\_data.py:461: RuntimeWarning: All-NaN slice encountered
  data_min = np.nanmin(X, axis=0)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\preprocessing\_data.py:462: RuntimeWarning: All-NaN slice encountered
  data_max = np.nanmax(X, axis=0)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\preprocessing\_data.py:461: RuntimeWarning: All-NaN slice encountered
  data_min = np.nanmin(X, axis=0)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\preprocessing\_data.py:462: RuntimeWarning: All-NaN slice encountered
  data_max = np.nanmax(X, axis=0)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\preprocessing\_data.py:461: RuntimeWarning: All-NaN slice encountered
  data_min = np.nanmin(X, axis=0)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\preprocessing\_data.py:462: RuntimeWarning: All-NaN slice encountered
  data_max = np.nanmax(X, axis=0)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\extmath.py:985: RuntimeWarning: invalid value encountered in divide
  updated_mean = (last_sum + new_sum) / updated_sample_count
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\extmath.py:990: RuntimeWarning: invalid value encountered in divide
  T = new_sum / new_sample_count
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\extmath.py:1020: RuntimeWarning: invalid value encountered in divide
  new_unnormalized_variance -= correction ** 2 / new_sample_count
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\extmath.py:985: RuntimeWarning: invalid value encountered in divide
  updated_mean = (last_sum + new_sum) / updated_sample_count
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\extmath.py:990: RuntimeWarning: invalid value encountered in divide
  T = new_sum / new_sample_count
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\extmath.py:1020: RuntimeWarning: invalid value encountered in divide
  new_unnormalized_variance -= correction ** 2 / new_sample_count
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\extmath.py:985: RuntimeWarning: invalid value encountered in divide
  updated_mean = (last_sum + new_sum) / updated_sample_count
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\extmath.py:990: RuntimeWarning: invalid value encountered in divide
  T = new_sum / new_sample_count
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\extmath.py:1020: RuntimeWarning: invalid value encountered in divide
  new_unnormalized_variance -= correction ** 2 / new_sample_count
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\extmath.py:985: RuntimeWarning: invalid value encountered in divide
  updated_mean = (last_sum + new_sum) / updated_sample_count
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\extmath.py:990: RuntimeWarning: invalid value encountered in divide
  T = new_sum / new_sample_count
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\extmath.py:1020: RuntimeWarning: invalid value encountered in divide
```

```
  new_unnormalized_variance -= correction ** 2 / new_sample_count
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\extmath.py:985: RuntimeWarning: invalid value encountered in divide
  updated_mean = (last_sum + new_sum) / updated_sample_count
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\extmath.py:990: RuntimeWarning: invalid value encountered in divide
  T = new_sum / new_sample_count
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\extmath.py:1020: RuntimeWarning: invalid value encountered in divide
  new_unnormalized_variance -= correction ** 2 / new_sample_count
```

In [23]:

```
#Write a python program to clean and manipulate unstructured text data using string manipulation techniques.
#Analyze how different data handling and cleaning techniques impacts the quality of the dataset.
#Steps to be performed:
#1. Webscrape the customer review of a product from any e commerce website
#2. Load the dataset containing unstructured textual data
#3. Clean the data by removing special characters, numbers and unnecessary spaces
#4. Extract key phrases like product name and ratings using regex and string methods
#5. Convert the extracted information into structured dataset with meaningful columns
import pandas as pd
import re
import random
def generate_random_reviews(num_reviews):
    products = ["Widget A", "Widget B", "Gadget C"]
    ratings = [1, 2, 3, 4, 5]
    reviews = []
    for _ in range(num_reviews):
        product = random.choice(products)
        rating = random.choice(ratings)
        review_text = f"This {product} is awesome! Rating: {rating}/5!!! Would buy again!!!"
        reviews.append(review_text)
    return reviews
num_reviews = 20
reviews_data = generate_random_reviews(num_reviews)
df = pd.DataFrame(reviews_data, columns=["Review"])
def clean_review(review):
    cleaned = re.sub(r'[^a-zA-Z\s]', '', review)
    cleaned = re.sub(r'\s+', ' ', cleaned).strip()
    return cleaned
df['Cleaned_Review'] = df['Review'].apply(clean_review)
def extract_product_and_rating(review):
    product_match = re.search(r'This (\w+ \w+|\w+)', review)
    product = product_match.group(1) if product_match else None
    rating_match = re.search(r'Rating:\s*(\d)', review)
    rating = int(rating_match.group(1)) if rating_match else None
    return product, rating
df[['Product', 'Rating']] = df['Review'].apply(extract_product_and_rating).apply(pd.Series)
```

```
print("Structured Dataset:")
print(df[['Product', 'Rating', 'Cleaned_Review']])
print("\nDataset Quality Analysis:")
print(f"Total Reviews: {len(df)}")
print(f"Unique Products: {df['Product'].nunique()}")
print(f"Average Rating: {df['Rating'].mean()}")
print(f"Missing Values:\n{df.isnull().sum()}")
```

```
Structured Dataset:
     Product  Rating                                    Cleaned_Review
0   Gadget C       4  This Gadget C is awesome Rating Would buy again
1   Widget B       4  This Widget B is awesome Rating Would buy again
2   Widget A       5  This Widget A is awesome Rating Would buy again
3   Widget A       4  This Widget A is awesome Rating Would buy again
4   Widget B       2  This Widget B is awesome Rating Would buy again
5   Gadget C       5  This Gadget C is awesome Rating Would buy again
6   Widget B       1  This Widget B is awesome Rating Would buy again
7   Widget B       1  This Widget B is awesome Rating Would buy again
8   Widget B       3  This Widget B is awesome Rating Would buy again
9   Gadget C       2  This Gadget C is awesome Rating Would buy again
10  Gadget C       2  This Gadget C is awesome Rating Would buy again
11  Widget A       1  This Widget A is awesome Rating Would buy again
12  Widget B       4  This Widget B is awesome Rating Would buy again
13  Widget A       1  This Widget A is awesome Rating Would buy again
14  Gadget C       4  This Gadget C is awesome Rating Would buy again
15  Gadget C       4  This Gadget C is awesome Rating Would buy again
16  Widget B       2  This Widget B is awesome Rating Would buy again
17  Widget B       1  This Widget B is awesome Rating Would buy again
18  Gadget C       3  This Gadget C is awesome Rating Would buy again
19  Widget A       5  This Widget A is awesome Rating Would buy again

Dataset Quality Analysis:
Total Reviews: 20
Unique Products: 3
Average Rating: 2.9
Missing Values:
Review            0
Cleaned_Review    0
Product           0
Rating            0
dtype: int64
```