240970107

Vikith B Kotian

In [10]:
```
import pandas as pd
import numpy as np

# Load the Excel file
df = pd.read_excel("Cereals1.xls")
```

In [11]:
```
#1) Create a table with the 5-number summary of all the numeric attributes.
numcols = df.select_dtypes(include='number').columns
summary = df[numcols].describe()

print(summary.loc[['min', '25%', '50%', '75%', 'max']])
```

```
     calories  protein  fat  sodium  fiber  carbo  sugars  potass  vitamin
s  \
min      50.0      1.0  0.0     0.0   0.00   -1.0    -1.0    -1.0       0.
0
25%     100.0      2.0  0.0   132.5   0.75   12.0     3.0    40.0      25.
0
50%     110.0      2.5  1.0   180.0   1.75   14.5     7.0    90.0      25.
0
75%     110.0      3.0  2.0   212.5   3.00   17.0    11.0   120.0      25.
0
max     160.0      6.0  5.0   320.0  14.00   23.0    15.0   330.0     100.
0

     shelf  weight  cups     rating
min    1.0     0.5  0.25  18.042851
25%    1.0     1.0  0.67  32.932466
50%    2.0     1.0  0.75  40.253086
75%    3.0     1.0  1.00  50.780847
max    3.0     1.5  1.50  93.704912
```

In [12]:
```
#2. For each of the numeric attributes (proteins up to vitamins), identify a
numeric_cols = df.loc[:, 'protein':'vitamins'].columns
df[numeric_cols] = df[numeric_cols].replace(-1, np.nan)

means = df[numeric_cols].mean()

df[numeric_cols] = df[numeric_cols].fillna(means)
print(df)
```

```
                         name mfr type  calories  protein  fat  sodium  fi
ber  \
0           100%_Natural_Bran   Q    C       120        3    5      15
2.0
1                    All-Bran   K    C        70        4    1     260
9.0
2   All-Bran_with_Extra_Fiber   K    C        50        4    0     140   1
4.0
3              Almond_Delight   R    C       110        2    2     200
1.0
4     Apple_Cinnamon_Cheerios   G    C       110        2    2     180
1.5
..                        ...  ..  ...       ...      ...  ...     ...
...
71                    Triples   G    C       110        2    1     250
0.0
72                       Trix   G    C       110        1    1     140
0.0
73                 Wheat_Chex   R    C       100        3    1     230
3.0
74                    Wheaties   G    C       100        3    1     200
3.0
75         Wheaties_Honey_Gold   G    C       110        2    1     200
1.0

    carbo  sugars      potass  vitamins  shelf  weight  cups     rating
0     8.0     8.0  135.000000         0      3     1.0  1.00  33.983679
1     7.0     5.0  320.000000        25      3     1.0  0.33  59.425505
2     8.0     0.0  330.000000        25      3     1.0  0.50  93.704912
3    14.0     8.0   96.216216        25      3     1.0  0.75  34.384843
4    10.5    10.0   70.000000        25      1     1.0  0.75  29.509541
..    ...     ...         ...       ...    ...     ...   ...        ...
71   21.0     3.0   60.000000        25      3     1.0  0.75  39.106174
72   13.0    12.0   25.000000        25      2     1.0  1.00  27.753301
73   17.0     3.0  115.000000        25      1     1.0  0.67  49.787445
74   17.0     3.0  110.000000        25      1     1.0  1.00  51.592193
75   16.0     8.0   60.000000        25      1     1.0  0.75  36.187559

[76 rows x 16 columns]
```

In [13]:
```
#3) Create a table with the 5-number summary of all the numeric attributes a
res=df.describe().loc[['min','25%','50%','75%','max']]
print(res)
```

```
     calories  protein  fat  sodium  fiber      carbo  sugars  potass  \
min      50.0      1.0  0.0     0.0   0.00   7.000000     0.0   15.00
25%     100.0      2.0  0.0   132.5   0.75  12.000000     3.0   43.75
50%     110.0      2.5  1.0   180.0   1.75  14.966667     7.0   90.00
75%     110.0      3.0  2.0   212.5   3.00  17.000000    11.0  120.00
max     160.0      6.0  5.0   320.0  14.00  23.000000    15.0  330.00

     vitamins  shelf  weight  cups     rating
min       0.0    1.0     0.5  0.25  18.042851
25%      25.0    1.0     1.0  0.67  32.932466
50%      25.0    2.0     1.0  0.75  40.253086
75%      25.0    3.0     1.0  1.00  50.780847
max     100.0    3.0     1.5  1.50  93.704912
```

In [14]:
```
#4) For each of the numeric attributes (proteins up to vitamins), identify a
for col in numeric_cols:
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1

    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    median = df[col].median()
    median_int = int(round(median))

    df.loc[(df[col] < lower_bound) | (df[col] > upper_bound), col] = median_
print(df)
```

```
                        name mfr type  calories  protein  fat  sodium  fi
ber  \
0            100%_Natural_Bran   Q    C       120        3    5      15
2.0
1                     All-Bran   K    C        70        4    1     260
2.0
2   All-Bran_with_Extra_Fiber   K    C        50        4    0     140
2.0
3               Almond_Delight   R    C       110        2    2     200
1.0
4     Apple_Cinnamon_Cheerios   G    C       110        2    2     180
1.5
..                        ...  ..  ...       ...      ...  ...     ...
...
71                    Triples   G    C       110        2    1     250
0.0
72                       Trix   G    C       110        1    1     140
0.0
73                 Wheat_Chex   R    C       100        3    1     230
3.0
74                    Wheaties   G    C       100        3    1     200
3.0
75         Wheaties_Honey_Gold   G    C       110        2    1     200
1.0

    carbo  sugars      potass  vitamins  shelf  weight  cups     rating
0     8.0     8.0  135.000000        25      3     1.0  1.00  33.983679
1     7.0     5.0   90.000000        25      3     1.0  0.33  59.425505
2     8.0     0.0   90.000000        25      3     1.0  0.50  93.704912
3    14.0     8.0   96.216216        25      3     1.0  0.75  34.384843
4    10.5    10.0   70.000000        25      1     1.0  0.75  29.509541
..    ...     ...         ...       ...    ...     ...   ...        ...
71   21.0     3.0   60.000000        25      3     1.0  0.75  39.106174
72   13.0    12.0   25.000000        25      2     1.0  1.00  27.753301
73   17.0     3.0  115.000000        25      1     1.0  0.67  49.787445
74   17.0     3.0  110.000000        25      1     1.0  1.00  51.592193
75   16.0     8.0   60.000000        25      1     1.0  0.75  36.187559

[76 rows x 16 columns]
```

In [15]:
```
#5) Create a table with the 5-number summary of all the numeric attributes a
res=df.describe().loc[['min','25%','50%','75%','max']]
print(res)
```

```
     calories  protein  fat  sodium  fiber      carbo  sugars  potass  \
min      50.0      1.0  0.0    15.0   0.00   7.000000     0.0   15.00
25%     100.0      2.0  0.0   147.5   0.75  12.000000     3.0   43.75
50%     110.0      2.0  1.0   180.0   1.75  14.966667     7.0   90.00
75%     110.0      3.0  2.0   212.5   3.00  17.000000    11.0  110.00
max     160.0      4.0  5.0   320.0   6.00  23.000000    15.0  230.00

     vitamins  shelf  weight  cups     rating
min      25.0    1.0     0.5  0.25  18.042851
25%      25.0    1.0     1.0  0.67  32.932466
50%      25.0    2.0     1.0  0.75  40.253086
75%      25.0    3.0     1.0  1.00  50.780847
max      25.0    3.0     1.5  1.50  93.704912
```