In [31]:
```
# Import all necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

#To Set seaborn style for better-looking plots
sns.set(style="whitegrid")


# Meaning of libraries and its function
# - pandas (pd) is for reading and analyzing the Excel file.
# - matplotlib.pyplot (plt) and seaborn (sns) are for creating charts.
# - numpy (np) helps with numerical calculations.
# - sns.set() makes plots look cleaner.
```

In [32]:
```
#To read the Excel file
df = pd.read_excel(r"C:\Users\Madina\Desktop\Quntative  data analysis\Nutrition_Ass

# To display the first 5 rows to understand the data
df.head()
```

Out[32]:

	What is your profession?	How many years of experience do you have in child nutrition assessment?	What is your highest level of education?	How often do you assess the nutritional status of children under five?	Which methods/tools do you commonly use for nutritional assessment?	Point out the of traditional n assessing
0	Clinical officer	4–6 years	Diploma	Daily	Anthropometric_measurements	Time
1	Nurse	Less than 1 year	Bachelor's degree	Occasionally	Observation	Lack_of_standa
2	Nurse	4–6 years	Certificate	Daily	Observation	Time
3	Nurse	Less than 1 year	Bachelor's degree	Weekly	Anthropometric_measurements	Limited_access_t
4	Nurse	Less than 1 year	Bachelor's degree	Weekly	Anthropometric_measurements	Limited_access_t

In [33]:
```
# To display basic information about the dataset shows the number of rows, columns,
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34 entries, 0 to 33
Data columns (total 10 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   What is your profession?
34 non-null     object
 1   How many years of experience do you have in child nutrition assessment?
34 non-null     object
 2   What is your highest level of education?
34 non-null     object
 3   How often do you assess the nutritional status of children under five?
34 non-null     object
 4   Which methods/tools do you commonly use for nutritional assessment?
34 non-null     object
 5   Point out the challenges of traditional methods for assessing nutritional sta
tus                                  34 non-null     object
 6   Do you feel that traditional methods are sufficient for accurately assessing
the nutritional status of children?  34 non-null     object
 7   Have you used or heard of AI-based methods in healthcare?
34 non-null     object
 8   What potential advantages do you see in using AI for nutritional assessments?
34 non-null     object
 9   What concerns or barriers do you foresee in implementing AI-based solutions?
34 non-null     object
dtypes: object(10)
memory usage: 2.8+ KB
```

In [34]:
```
# To get summary statistics for all columns
df.describe(include='all')
```

Out[34]:

	What is your profession?	How many years of experience do you have in child nutrition assessment?	What is your highest level of education?	How often do you assess the nutritional status of children under five?	Which methods/tools do you commonly use for nutritional assessment?	Point ou challeng tradit metho asse nutrit s
count	34	34	34	34	34	
unique	6	4	4	3	7	
top	Nurse	More than 6 years	Bachelor's degree	Daily	Anthropometric_measurements	Time_consu
freq	15	12	18	27	10	

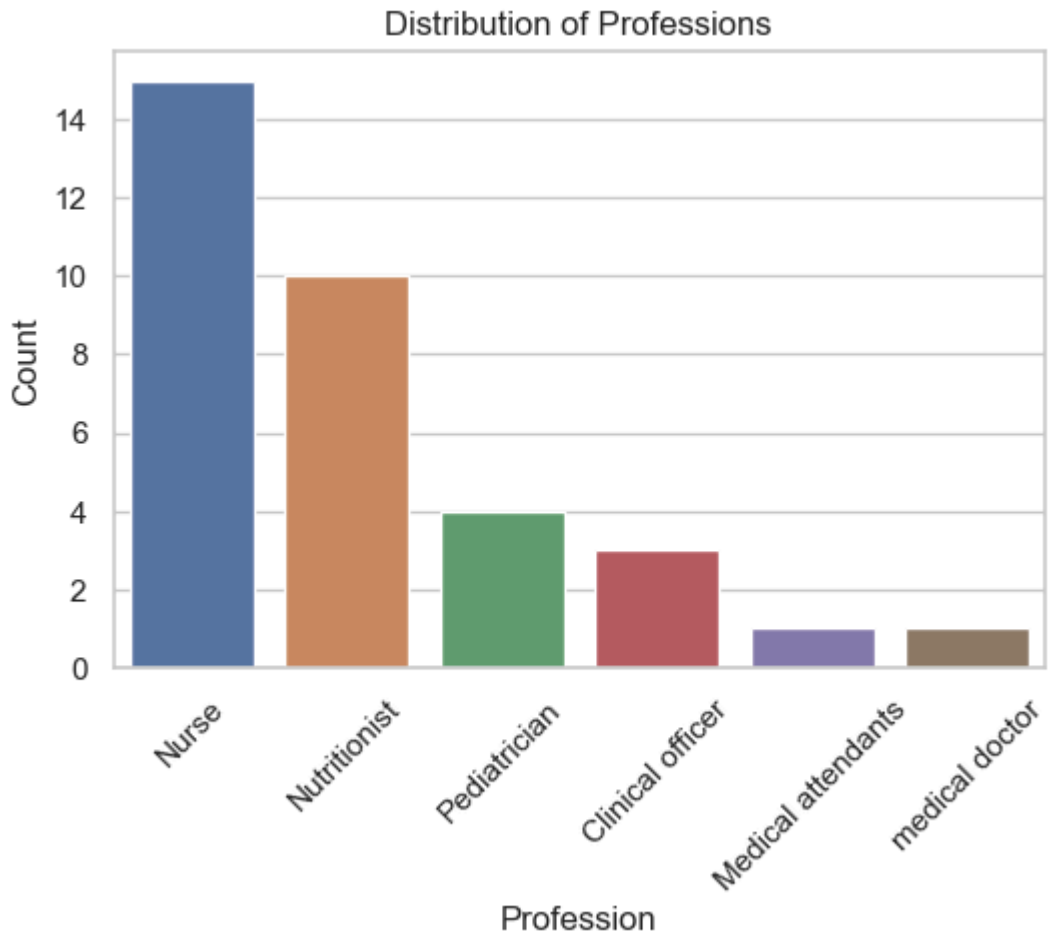
In [35]:
```
#To count unique values in the "What is your profession?" column
profession_counts = df['What is your profession?'].value_counts()

#To Display the counts
print(profession_counts)
```

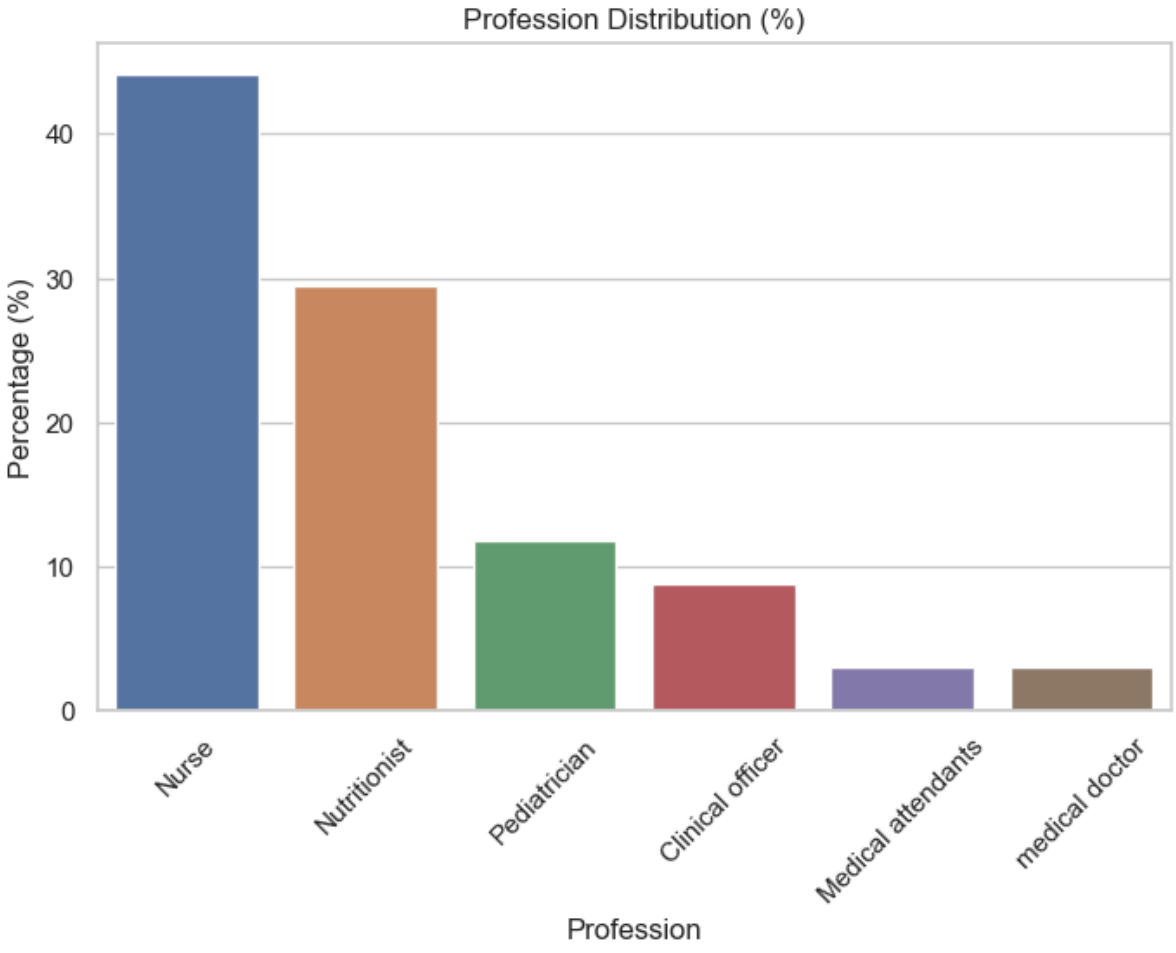
```
# Count professions and calculate percentages
profession_counts = df['What is your profession?'].value_counts()
profession_percent = df['What is your profession?'].value_counts(normalize=True) *

# Combine counts and percentages into a DataFrame
profession_summary = pd.DataFrame({
    'Count': profession_counts,
    'Percentage (%)': profession_percent.round(2)
})

# Display the summary
print(profession_summary)
```

```
Nurse                15
Nutritionist         10
Pediatrician          4
Clinical officer      3
Medical attendants    1
medical doctor        1
Name: What is your profession?, dtype: int64
                    Count  Percentage (%)
Nurse                  15           44.12
Nutritionist           10           29.41
Pediatrician            4           11.76
Clinical officer        3            8.82
Medical attendants      1            2.94
medical doctor          1            2.94
```

In [36]:

```
#To create a bar chart for profession counts
plt.figure(figsize=(6, 4))
sns.countplot(data=df, x='What is your profession?', order=df['What is your profess
plt.title('Distribution of Professions')
plt.xlabel('Profession')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()


# Create a bar chart for professions
plt.figure(figsize=(8, 5))
sns.barplot(x=profession_counts.index, y=profession_percent)
plt.title('Profession Distribution (%)')
plt.xlabel('Profession')
plt.ylabel('Percentage (%)')
plt.xticks(rotation=45)
plt.show()

# meaning
# - plt.figure(figsize=(8, 5)) sets the chart size.
# - sns.countplot() creates a bar chart for the profession column.
# - order=... sorts bars by frequency.
# - plt.title(), plt.xlabel(), plt.ylabel() add labels.
# - plt.xticks(rotation=45) rotates x-axis labels for readability.
```





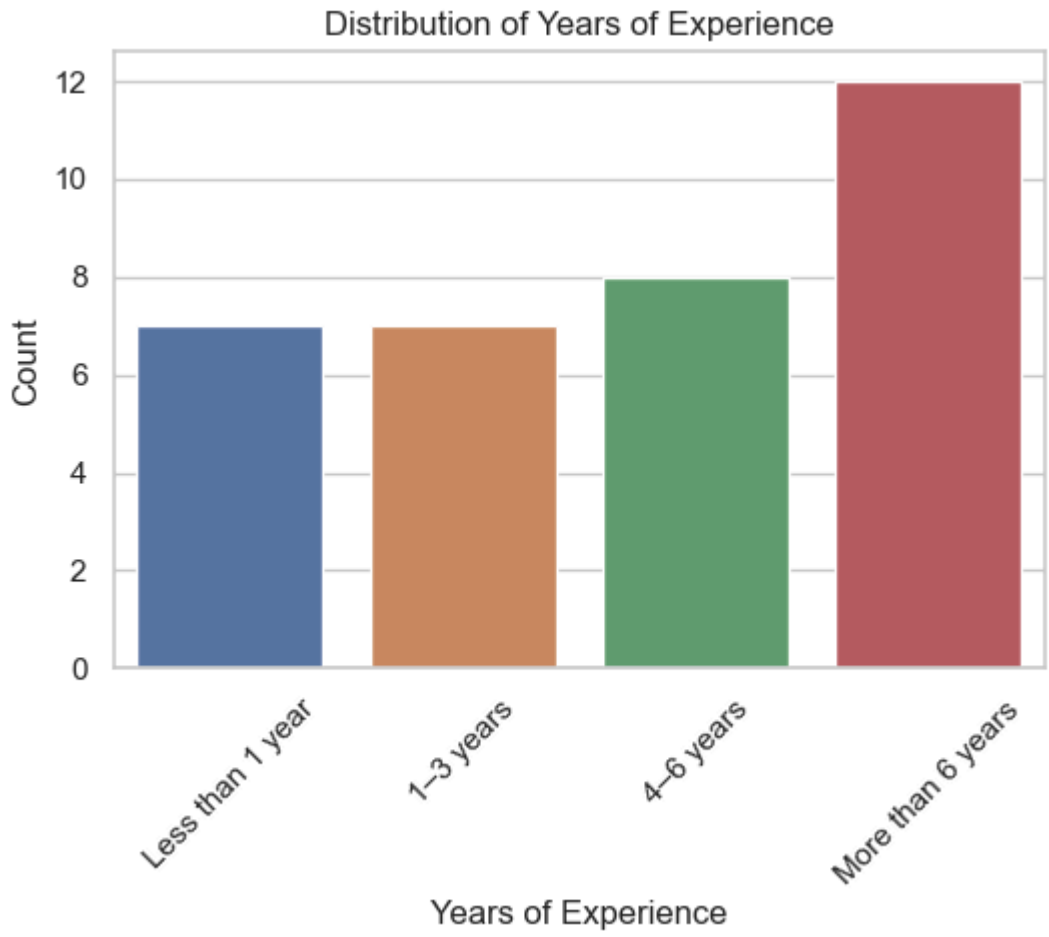
In [37]:
```
# To count unique values in the "How many years of experience..." column
experience_counts = df['How many years of experience do you have in child nutritior
```

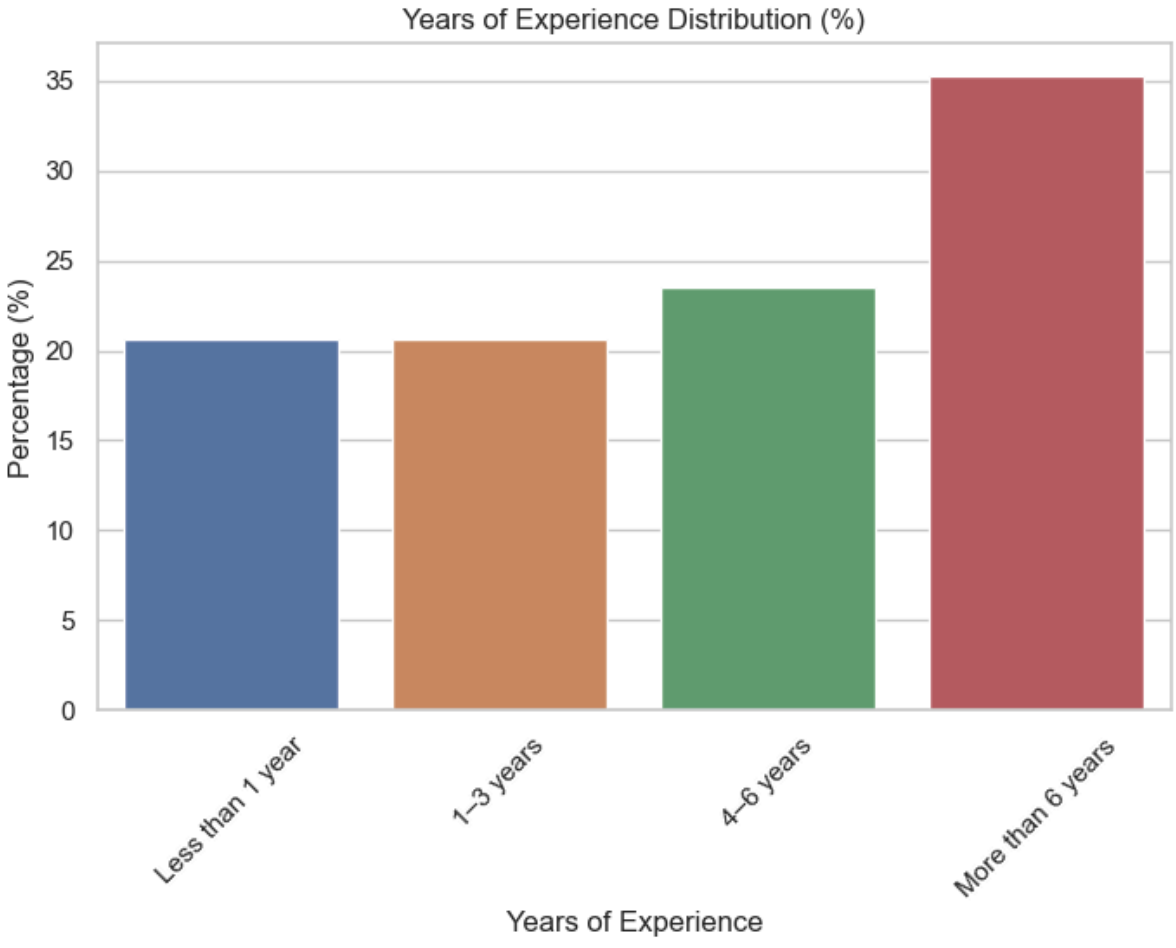
```
#To display the counts
print(experience_counts)


# Count experience levels and calculate percentages
experience_counts = df['How many years of experience do you have in child nutritior
experience_percent = df['How many years of experience do you have in child nutritic

# Combine into a DataFrame
experience_summary = pd.DataFrame({
    'Count': experience_counts,
    'Percentage (%)': experience_percent.round(2)
})

# Display the summary
print(experience_summary)
```

```
More than 6 years    12
4-6 years             8
Less than 1 year      7
1-3 years             7
Name: How many years of experience do you have in child nutrition assessment?, dty
pe: int64
                   Count  Percentage (%)
More than 6 years     12           35.29
4-6 years              8           23.53
Less than 1 year       7           20.59
1-3 years              7           20.59
```

In [38]:

```
#To  create a bar chart for experience counts
plt.figure(figsize=(6, 4))
sns.countplot(data=df, x='How many years of experience do you have in child nutriti
plt.title('Distribution of Years of Experience')
plt.xlabel('Years of Experience')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()

# Create a bar chart for experience
plt.figure(figsize=(8, 5))
sns.barplot(x=experience_counts.index, y=experience_percent, order=['Less than 1 ye
plt.title('Years of Experience Distribution (%)')
plt.xlabel('Years of Experience')
plt.ylabel('Percentage (%)')
plt.xticks(rotation=45)
plt.show()
```





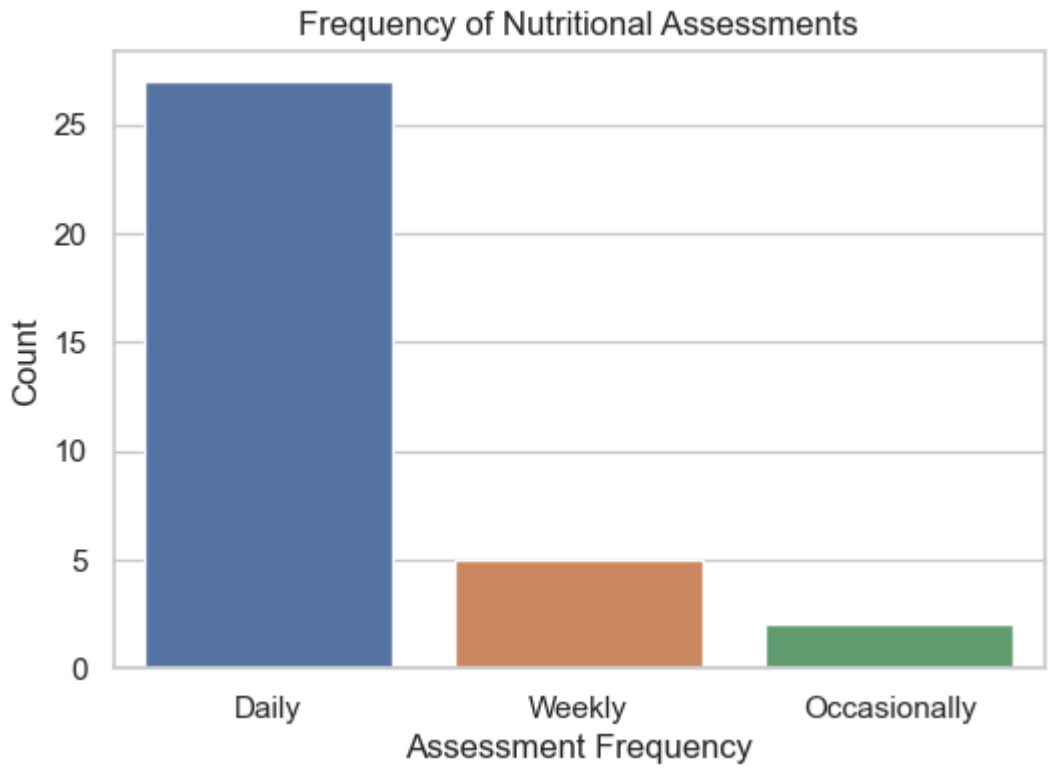
In [39]:
```
# To Count unique values in the "How often do you assess..." column
assessment_freq = df['How often do you assess the nutritional status of children ur
```

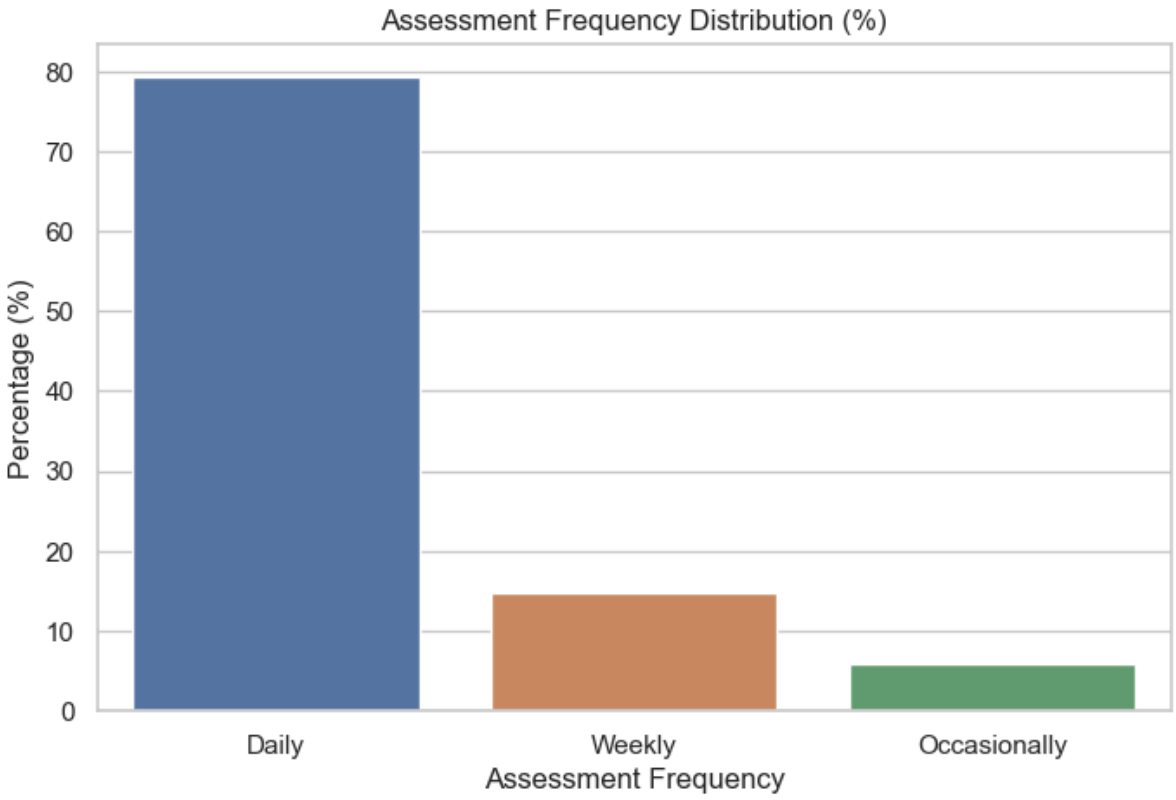
```
# To Display the counts
print(assessment_freq)


# Count assessment frequency and calculate percentages
assessment_counts = df['How often do you assess the nutritional status of children
assessment_percent = df['How often do you assess the nutritional status of childre

# Combine into a DataFrame
assessment_summary = pd.DataFrame({
    'Count': assessment_counts,
    'Percentage (%)': assessment_percent.round(2)
})

# Display the summary
print(assessment_summary)
```

```
Daily           27
Weekly           5
Occasionally     2
Name: How often do you assess the nutritional status of children under five?, dtyp
e: int64
              Count  Percentage (%)
Daily            27           79.41
Weekly            5           14.71
Occasionally      2            5.88
```

In [40]:

```
#To Create a bar chart for assessment frequency
plt.figure(figsize=(6, 4))
sns.countplot(data=df, x='How often do you assess the nutritional status of childre
plt.title('Frequency of Nutritional Assessments')
plt.xlabel('Assessment Frequency')
plt.ylabel('Count')
plt.show()


# Create a bar chart for assessment frequency
plt.figure(figsize=(8, 5))
sns.barplot(x=assessment_counts.index, y=assessment_percent, order=['Daily', 'Weekl
plt.title('Assessment Frequency Distribution (%)')
plt.xlabel('Assessment Frequency')
plt.ylabel('Percentage (%)')
plt.show()
```



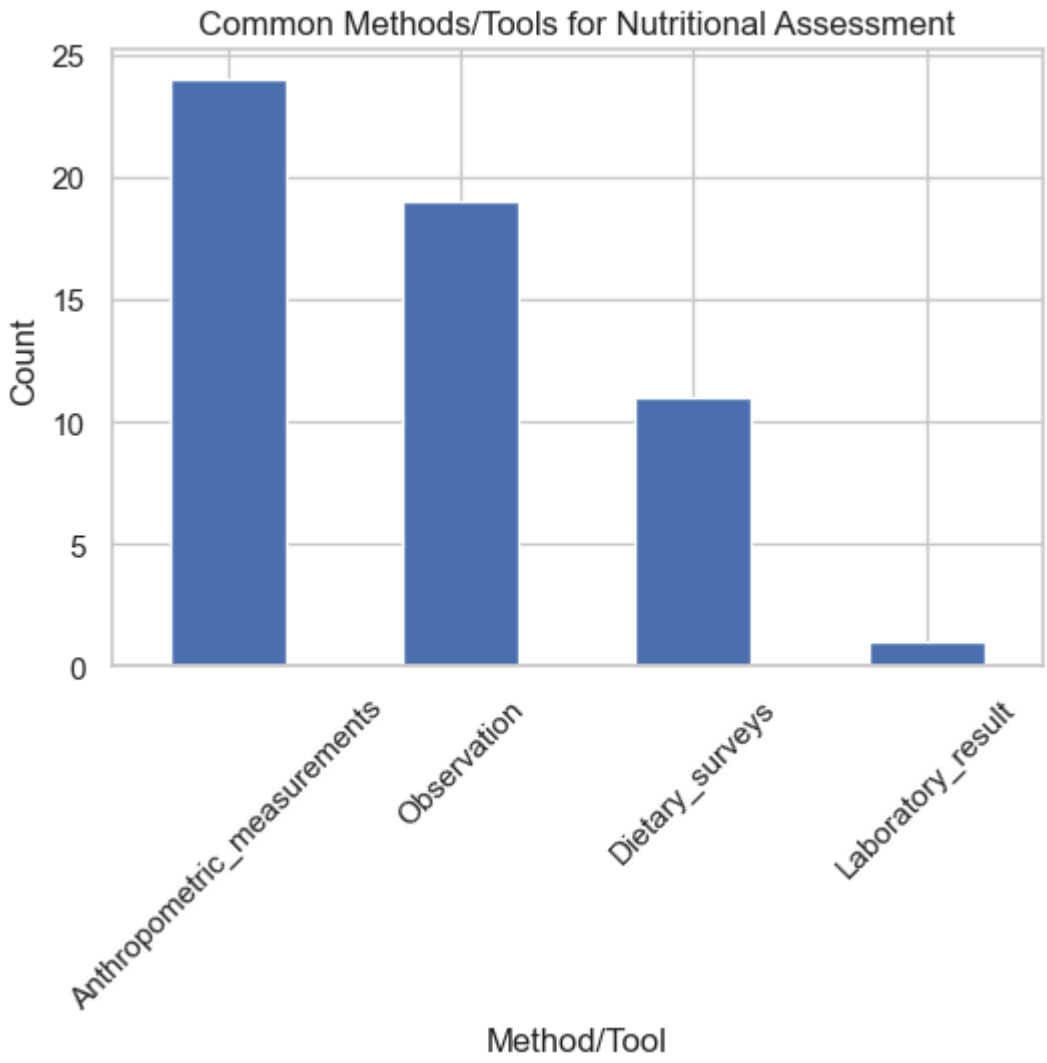


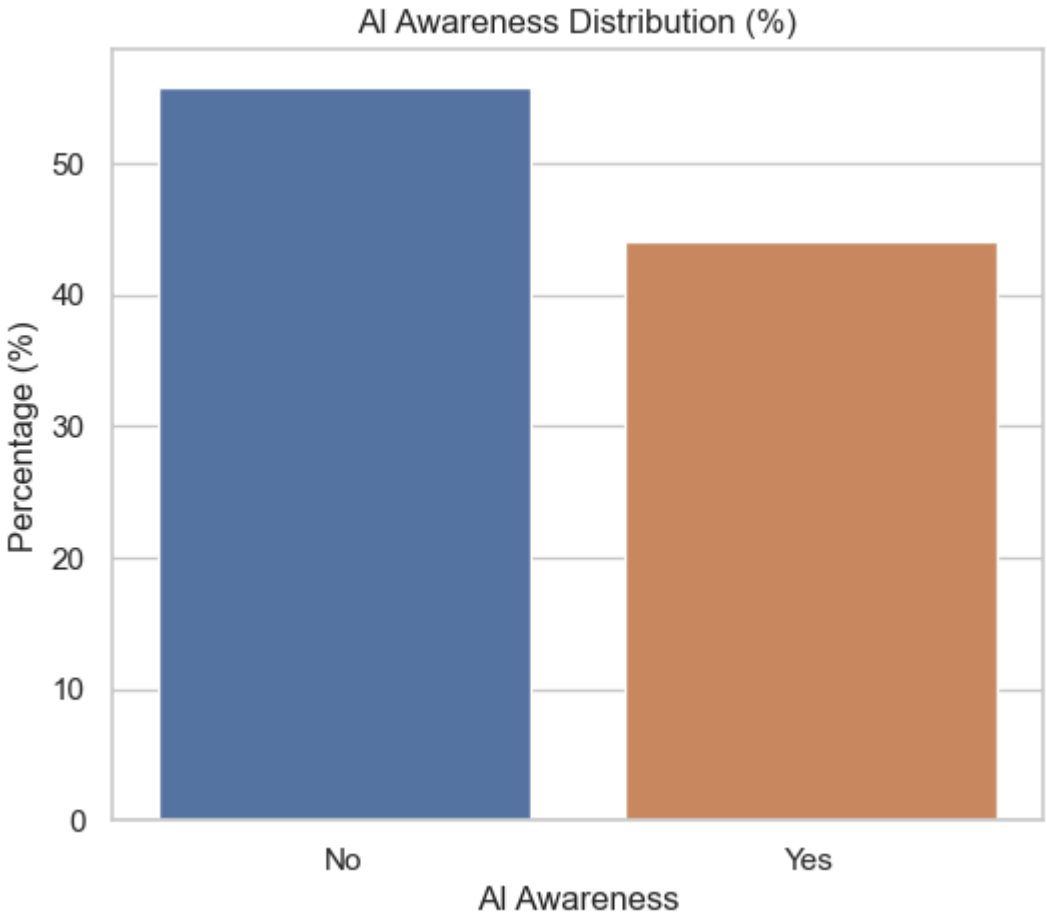
In [41]:
```
# To Split and count methods/tools used
methods = df['Which methods/tools do you commonly use for nutritional assessment?']

#To Display the counts
print(methods)


# Count AI awareness and calculate percentages
ai_awareness_counts = df['Have you used or heard of AI-based methods in healthcare?
ai_awareness_percent = df['Have you used or heard of AI-based methods in healthcare

# Combine into a DataFrame
ai_awareness_summary = pd.DataFrame({
```

```
    'Count': ai_awareness_counts,
    'Percentage (%)': ai_awareness_percent.round(2)
})

# Display the summary
print(ai_awareness_summary)
```

```
Anthropometric_measurements    24
Observation                    19
Dietary_surveys                11
Laboratory_result               1
dtype: int64
     Count  Percentage (%)
No      19           55.88
Yes     15           44.12
```

In [42]:
```
# To Create a bar chart for methods/tools
plt.figure(figsize=(6,4))
methods.plot(kind='bar')
plt.title('Common Methods/Tools for Nutritional Assessment')
plt.xlabel('Method/Tool')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()


# Create a bar chart for AI awareness
plt.figure(figsize=(6, 5))
sns.barplot(x=ai_awareness_counts.index, y=ai_awareness_percent)
plt.title('AI Awareness Distribution (%)')
plt.xlabel('AI Awareness')
plt.ylabel('Percentage (%)')
plt.show()
```

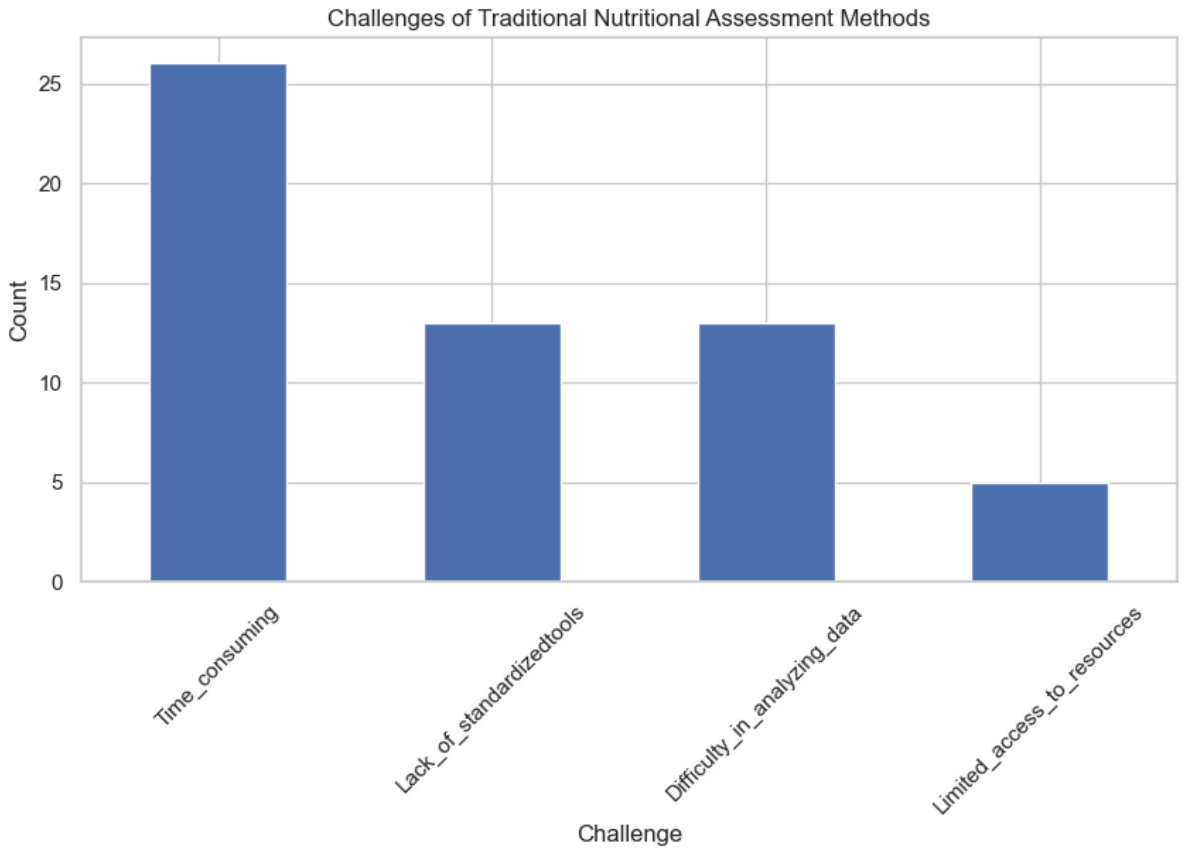

Common Methods/Tools for Nutritional Assessment
Count
25
20
15
10
5
0
Anthropometric_measurements
Observation
Dietary_surveys
Laboratory_result
Method/Tool


AI Awareness Distribution (%)
Percentage (%)
50
40
30
20
10
0
No
Yes
AI Awareness

In [53]:
```
# To split and count challenges correctly
challenges = df['Point out the challenges of traditional methods for assessing nutr
    .dropna() \
    .str.split(' ', expand=True) \
    .stack() \
    .str.strip() \
    .loc[lambda x: x != ''] \
    .value_counts()

# To display the counts
print(challenges)
```

```
Time_consuming                 26
Lack_of_standardizedtools      13
Difficulty_in_analyzing_data   13
Limited_access_to_resources     5
dtype: int64
```

In [58]:
```
# To create a bar chart for challenges
plt.figure(figsize=(10,5))
challenges.plot(kind='bar')
plt.title('Challenges of Traditional Nutritional Assessment Methods')
plt.xlabel('Challenge')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```

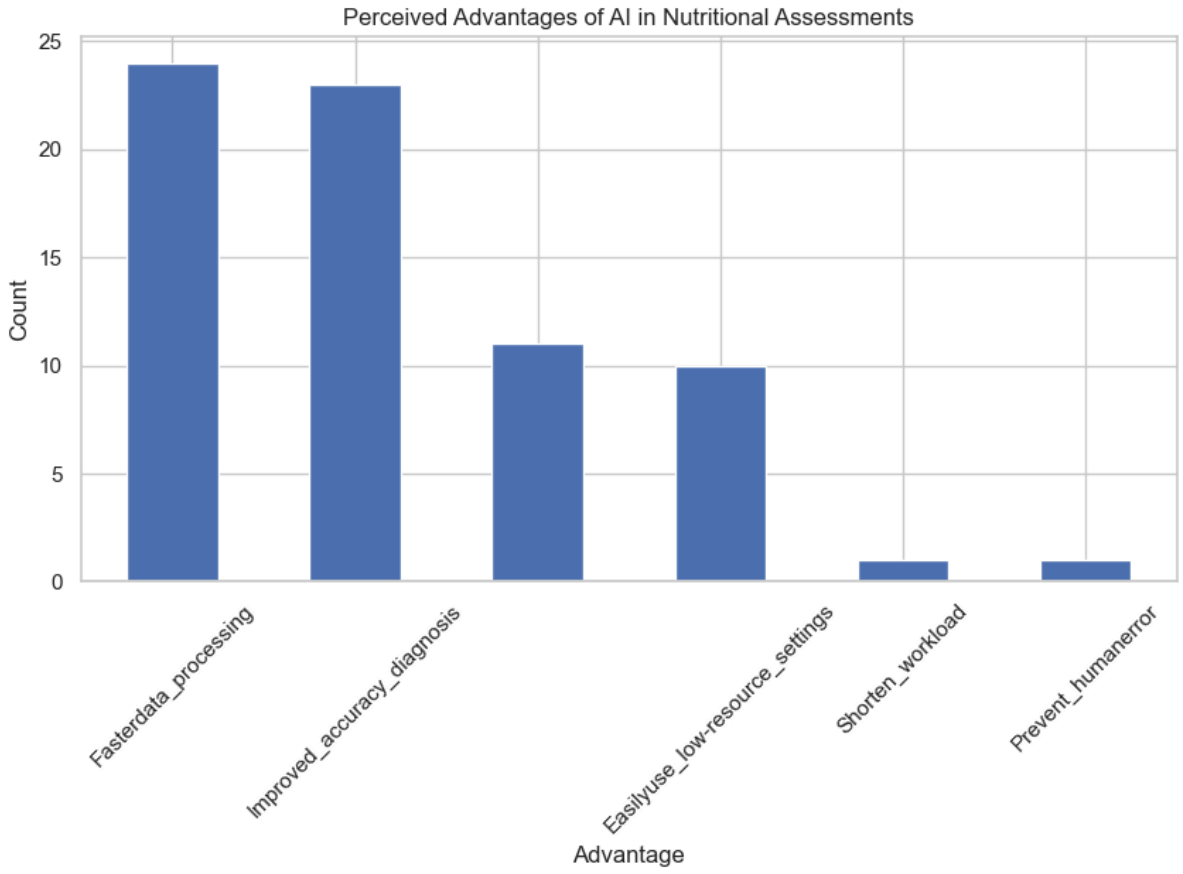


In [14]:
```
#To count advantages of AI
advantages = df['What potential advantages do you see in using AI for nutritional a
print("Advantages of AI:")
print(advantages)
```

```
Advantages of AI:
Fasterdata_processing              24
Improved_accuracy_diagnosis        23
                                   11
Easilyuse_low-resource_settings    10
Shorten_workload                    1
Prevent_humanerror                  1
dtype: int64
```

In [244…

```
plt.figure(figsize=(10, 5))
advantages.plot(kind='bar')
plt.title('Perceived Advantages of AI in Nutritional Assessments')
plt.xlabel('Advantage')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



In [245…

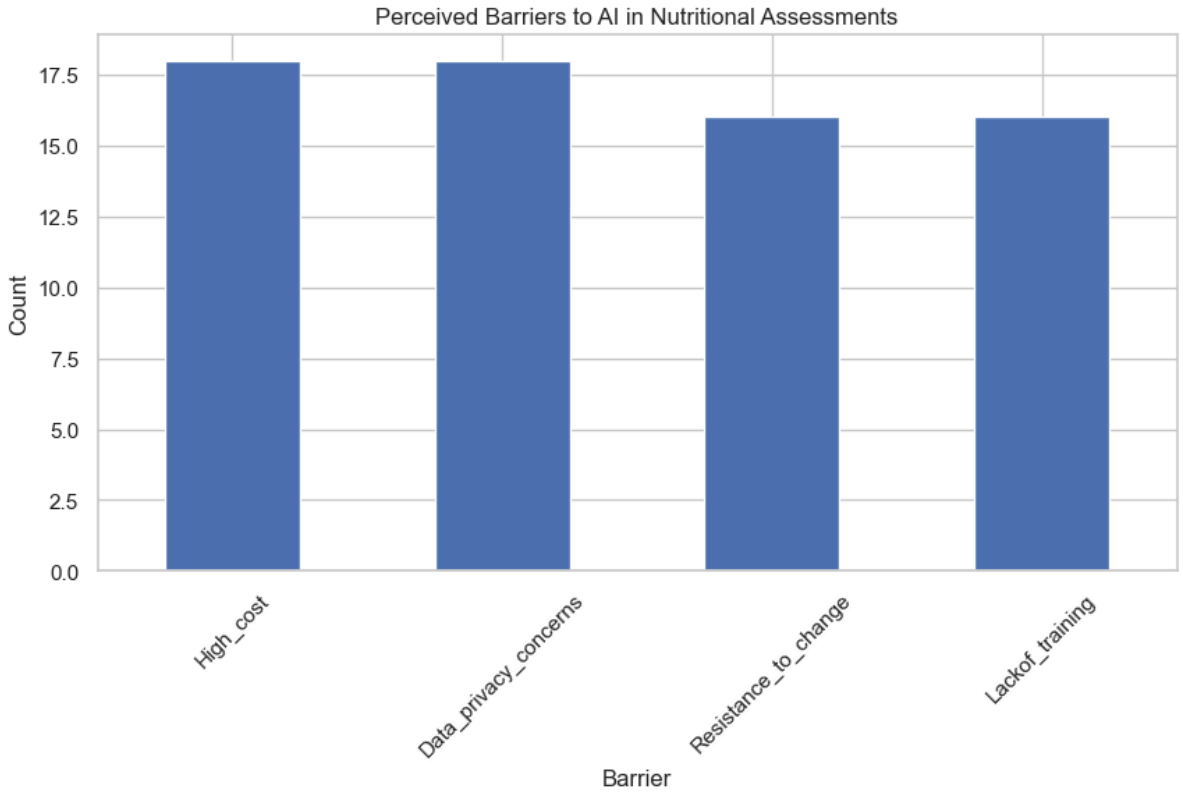
```
# To count barriers to AI
barriers = df['What concerns or barriers do you foresee in implementing AI-based sc
print("\nBarriers to AI:")
print(barriers)
```

```
Barriers to AI:
High_cost                18
Data_privacy_concerns    18
Resistance_to_change     16
Lackof_training          16
dtype: int64
```

In [246…

```
# Create bar chart for AI barriers
plt.figure(figsize=(10, 5))
barriers.plot(kind='bar')
plt.title('Perceived Barriers to AI in Nutritional Assessments')
plt.xlabel('Barrier')
plt.ylabel('Count')
```

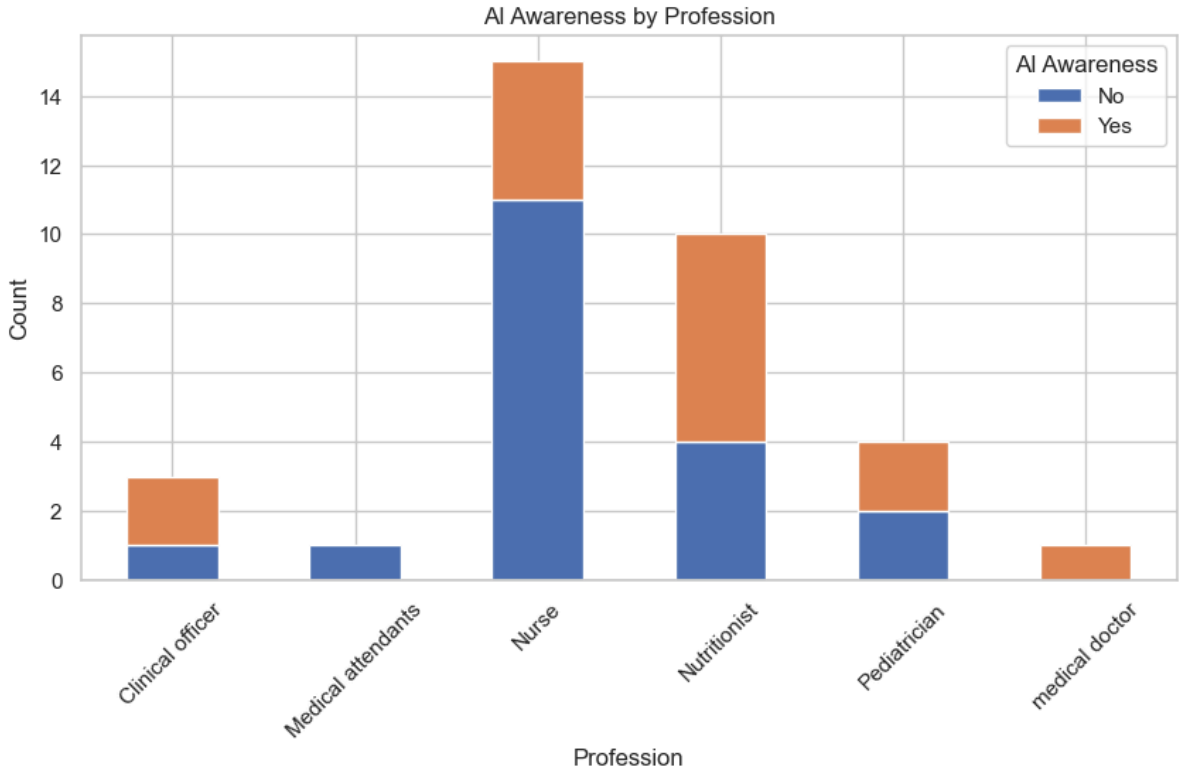
```
plt.xticks(rotation=45)
plt.show()
```



In [247…

```
# Create a cross-tabulation of profession and AI awareness
cross_tab = pd.crosstab(df['What is your profession?'], df['Have you used or heard

# Display the cross-tabulation
print(cross_tab)

# Create a stacked bar chart for cross-tabulation
cross_tab.plot(kind='bar', stacked=True, figsize=(10, 5))
plt.title('AI Awareness by Profession')
plt.xlabel('Profession')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.legend(title='AI Awareness')
plt.show()
```

```
Have you used or heard of AI-based methods in healthcare?  No  Yes
What is your profession?
Clinical officer                                            1    2
Medical attendants                                          1    0
Nurse                                                      11    4
Nutritionist                                                4    6
Pediatrician                                                2    2
medical doctor                                              0    1
```

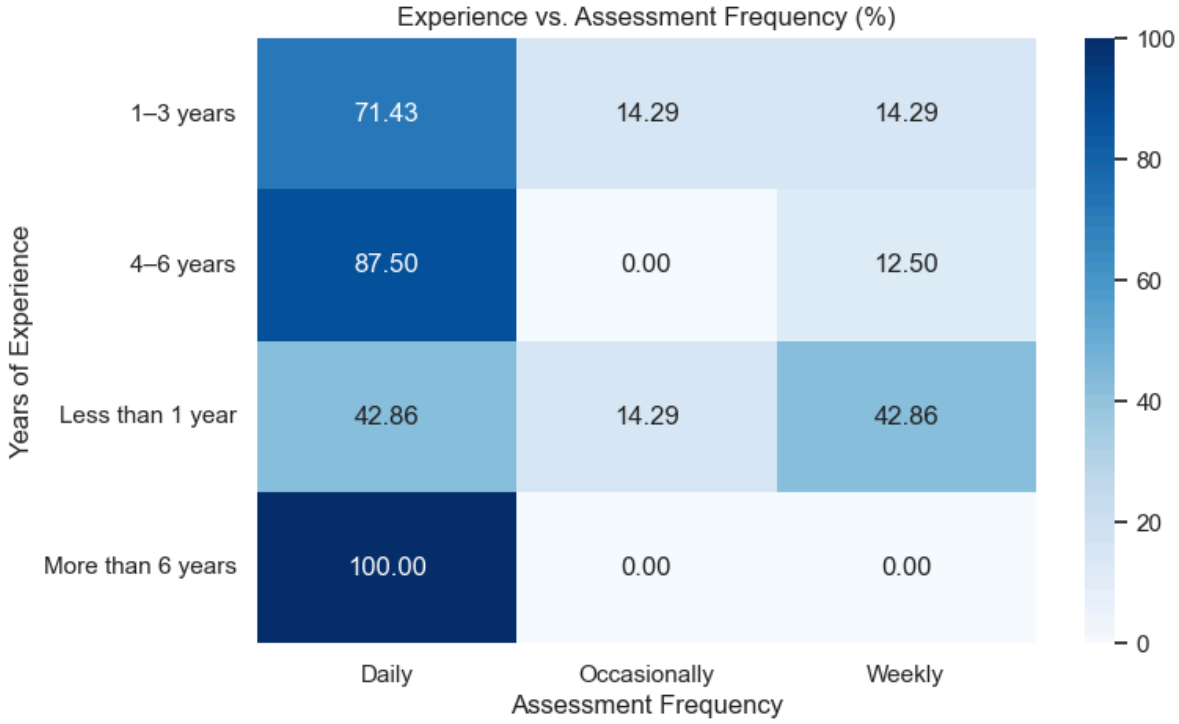


In [250…

```python
from scipy.stats import chi2_contingency  # <- Fixes the error

# Define Cramér's V function
def cramers_v(x, y):
    contingency_table = pd.crosstab(x, y)
    chi2 = chi2_contingency(contingency_table)[0]
    n = contingency_table.sum().sum()
    r, k = contingency_table.shape
    return np.sqrt(chi2 / (n * (min(r, k) - 1)))

# List of variables to check associations
columns = [
    'What is your profession?',
    'Have you used or heard of AI-based methods in healthcare?',
    'How many years of experience do you have in child nutrition assessment?',
    'How often do you assess the nutritional status of children under five?'
]

# Create empty DataFrame for results
cramers_results = pd.DataFrame(index=columns, columns=columns)

# Compute Cramér's V for each pair
for col1 in columns:
    for col2 in columns:
        if col1 == col2:
            cramers_results.loc[col1, col2] = 1.0  # perfect correlation with self
        else:
            cramers_results.loc[col1, col2] = cramers_v(df[col1], df[col2])

# Convert values to float
cramers_results = cramers_results.astype(float)

# Plot heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(cramers_results, annot=True, cmap="YlGnBu", fmt=".2f", square=True)
plt.title("Cramér's V Heatmap: Strength of Association")
plt.tight_layout()
plt.show()
```

In [251…

```
# Create cross-tabulation for Experience vs. Assessment Frequency
cross_tab = pd.crosstab(
    df['How many years of experience do you have in child nutrition assessment?'],
    df['How often do you assess the nutritional status of children under five?'],
    normalize='index'
) * 100  # Convert to percentages

# Create a heatmap
plt.figure(figsize=(8, 5))
sns.heatmap(cross_tab, annot=True, fmt='.2f', cmap='Blues')
plt.title('Experience vs. Assessment Frequency (%)')
plt.xlabel('Assessment Frequency')
plt.ylabel('Years of Experience')
plt.show()
```


Experience vs. Assessment Frequency (%)
Years of Experience
1–3 years
4–6 years
Less than 1 year
More than 6 years
71.43
14.29
14.29
87.50
0.00
12.50
42.86
14.29
42.86
100.00
0.00
0.00
Daily
Occasionally
Weekly
Assessment Frequency
100
80
60
40
20
0

In [ ]: