```
import pandas as pd
import numpy as np
from scipy import stats

# Sample data
data = {
    'product_title': ['Pineapple slicer', 'Levis Jeans Pant', 'Wallet', 'Salwar'],
    'product_category': ['Apparel', 'Apparel', 'Apparel', 'Apparel'],
    'star_rating': [4, 5, 5, 5],
    'review_headline': ['Really good', 'Perfect Dress', 'Love it', 'Awesome'],
    'review_date': ['2013-01-14', '2014-04-22', '2015-07-28', '2015-06-12']
}

# Create DataFrame
df = pd.DataFrame(data)

# Filter by product category
category = 'Apparel'
df_category = df[df['product_category'] == category]

# Calculate mean rating
mean_rating = df_category['star_rating'].mean()

# Calculate standard deviation and sample size
std_dev = df_category['star_rating'].std()
n = len(df_category)

# Calculate confidence interval
confidence_level = 0.95
degrees_freedom = n - 1
confidence_interval = stats.t.interval(
    confidence_level,
    degrees_freedom,
    mean_rating,
    std_dev / np.sqrt(n)
)

# Display results
print(f"Mean rating for '{category}': {mean_rating:.2f}")
print(f"95% confidence interval for the mean rating: ({confidence_interval[0]:.2f}, {confidence_inte
```

```
Mean rating for 'Apparel': 4.75
95% confidence interval for the mean rating: (3.95, 5.55)
```

```
import pandas as pd

# Sample data
data = {
    'DISEASE_NAME': ['Common Cold', 'Diabetes', 'Bronchitis', 'Influenza', 'Kidney Stones'],
    'DIAGNOSED_PATIENTS': [320, 120, 100, 150, 60]
}

# Create DataFrame
df = pd.DataFrame(data)

# Calculate the frequency distribution (already provided by DIAGNOSED_PATIENTS)
# Find the most common disease
most_common_disease = df.loc[df['DIAGNOSED_PATIENTS'].idxmax()]

# Display the results
print("Frequency distribution of diseases:")
print(df)

print("\nMost common disease:")
print(f"Disease: {most_common_disease['DISEASE_NAME']}")
print(f"Number of diagnosed patients: {most_common_disease['DIAGNOSED_PATIENTS']}")
```

```
Frequency distribution of diseases:
    DISEASE_NAME  DIAGNOSED_PATIENTS
0    Common Cold                 320
1       Diabetes                 120
2     Bronchitis                 100
3      Influenza                 150
4  Kidney Stones                  60

Most common disease:
Disease: Common Cold
Number of diagnosed patients: 320
```

```
import pandas as pd

# Sample data
data = {
    'WEATHER_CONDITION': ['Sunny', 'Rainy', 'Cloudy', 'Snowy', 'Windy'],
    'OCCURRENCES': [120, 80, 60, 30, 50]
}

# Create DataFrame
df = pd.DataFrame(data)

# Calculate the frequency distribution (already provided by OCCURRENCES)
# Find the most common weather condition
most_common_weather = df.loc[df['OCCURRENCES'].idxmax()]

# Display the results
print("Frequency distribution of weather conditions:")
print(df)

print("\nMost common weather type:")
print(f"Weather Condition: {most_common_weather['WEATHER_CONDITION']}")
print(f"Number of occurrences: {most_common_weather['OCCURRENCES']}")
```

```
Frequency distribution of weather conditions:
  WEATHER_CONDITION  OCCURRENCES
0             Sunny          120
1             Rainy           80
2            Cloudy           60
3             Snowy           30
4             Windy           50

Most common weather type:
Weather Condition: Sunny
Number of occurrences: 120
```

```
import pandas as pd

# Sample data: DataFrame containing sales data with customer ages
data = {
    'customer_id': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    'purchase_amount': [100, 150, 200, 250, 300, 350, 400, 450, 500, 550],
    'age': [22, 35, 40, 29, 23, 45, 32, 38, 26, 41]
}

# Create DataFrame
df = pd.DataFrame(data)

# Calculate the frequency distribution of ages
age_frequency = df['age'].value_counts().sort_index()

# Display the frequency distribution
print("Frequency distribution of customer ages:")
print(age_frequency)
```

```
Frequency distribution of customer ages:
age
22    1
23    1
26    1
29    1
32    1
35    1
38    1
40    1
41    1
45    1
Name: count, dtype: int64
```

```
import pandas as pd

# Sample data: DataFrame containing post interaction data with number of likes
data = {
    'post_id': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    'likes': [120, 150, 150, 200, 220, 180, 150, 240, 120, 200]
}

# Create DataFrame
df = pd.DataFrame(data)

# Calculate the frequency distribution of likes
likes_frequency = df['likes'].value_counts().sort_index()

# Display the frequency distribution
print("Frequency distribution of likes:")
print(likes_frequency)
```

```
Frequency distribution of likes:
likes
120    2
150    3
180    1
200    2
220    1
240    1
Name: count, dtype: int64
```

```
import pandas as pd
from collections import Counter
import re

# Sample data: DataFrame containing customer reviews
data = {
    'review_id': [1, 2, 3, 4, 5],
    'review_text': [
        'The product is great, I love it!',
        'Excellent quality and very durable.',
        'Not worth the money, very disappointed.',
        'Great product, will buy again.',
        'The quality is okay, not as expected.'
```

```
        ]
    }

    # Create DataFrame
    df = pd.DataFrame(data)

    # Function to preprocess and tokenize text
    def preprocess_text(text):
        # Convert to lowercase
        text = text.lower()
        # Remove punctuation and numbers
        text = re.sub(r'[^\w\s]', '', text)
        # Tokenize by splitting on whitespace
        words = text.split()
        return words

    # Preprocess and tokenize all reviews
    all_words = []
    for review in df['review_text']:
        all_words.extend(preprocess_text(review))

    # Calculate the frequency distribution of words
    word_freq = Counter(all_words)

    # Display the frequency distribution
    print("Frequency distribution of words:")
    for word, freq in word_freq.items():
        print(f"{word}: {freq}")
```

```
Frequency distribution of words:
the: 3
product: 2
is: 2
great: 2
i: 1
love: 1
it: 1
excellent: 1
quality: 2
and: 1
very: 2
durable: 1
not: 2
worth: 1
money: 1
```