```
# Install required packages
!pip install nltk regex

# Import necessary libraries
import nltk
import re
import string
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

# Download NLTK data files
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('punkt_tab') # Download punkt_tab as well

# Define the preprocessing function
def preprocess_text(text):
    print("Original Text:")
    print(text)
    print("\n--- Preprocessing Steps ---")

    # 1. Tokenization
    tokens = word_tokenize(text)
    print("\n1. Tokens:")
    print(tokens)

    # 2. Filtration (remove punctuation and stopwords)
    stop_words = set(stopwords.words('english'))
    filtered_tokens = [
        word for word in tokens
        if word.lower() not in stop_words and word not in string.punctuation
    ]
    print("\n2. After Filtration (no stopwords/punctuation):")
    print(filtered_tokens)

    # 3. Script Validation: Keep only Latin-script words (basic ASCII check)
    latin_tokens = [word for word in filtered_tokens if re.match(r'^[\x00-\x7F]+$', word)]
    print("\n3. After Script Validation (Latin only):")
    print(latin_tokens)

    return latin_tokens

# Test the function
sample_text = "Hello! This is a test 😊. Let's remove non-Latin字符 and stopwords."
final_tokens = preprocess_text(sample_text)
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.11/dist-packages (3.9.1)
Requirement already satisfied: regex in /usr/local/lib/python3.11/dist-packages (2024.11.6)
Requirement already satisfied: click in /usr/local/lib/python3.11/dist-packages (from nltk) (8.2.1)
Requirement already satisfied: joblib in /usr/local/lib/python3.11/dist-packages (from nltk) (1.5.1)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from nltk) (4.67.1)
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
Original Text:
Hello! This is a test 😊. Let's remove non-Latin字符 and stopwords.

--- Preprocessing Steps ---

1. Tokens:
['Hello', '!', 'This', 'is', 'a', 'test', '😊', '.', 'Let', "'s", 'remove', 'non-Latin字符', 'and', 'stopwords', '.']

2. After Filtration (no stopwords/punctuation):
['Hello', 'test', '😊', 'Let', "'s", 'remove', 'non-Latin字符', 'stopwords']

3. After Script Validation (Latin only):
['Hello', 'test', 'Let', "'s", 'remove', 'stopwords']
```

```
# Install required packages
!pip install nltk regex

import nltk
import re
import string
import pandas as pd
```

```
import pandas as pd
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
import zipfile
import urllib.request

# Download NLTK resources
nltk.download('punkt')
nltk.download('stopwords')

# Download and extract dataset
dataset_url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/00228/smsspamcollection.zip'
filename = 'smsspamcollection.zip'
urllib.request.urlretrieve(dataset_url, filename)

with zipfile.ZipFile(filename, 'r') as zip_ref:
    zip_ref.extractall('sms_spam_data')

# Load dataset
data_path = 'sms_spam_data/SMSSpamCollection'

# Read dataset (tab-separated)
df = pd.read_csv(data_path, sep='\t', header=None, names=['label', 'message'])

# Show first 5 rows
print(df.head())

# Preprocessing function
def preprocess_text(text):
    # Tokenize text into words
    tokens = word_tokenize(text)

    # Remove stopwords and punctuation
    stop_words = set(stopwords.words('english'))
    filtered_tokens = [word for word in tokens if word.lower() not in stop_words and word not in string.punctuation]

    # Keep only ASCII characters (Latin script)
    latin_tokens = [word for word in filtered_tokens if re.match(r'^[\x00-\x7F]+$', word)]

    return ' '.join(latin_tokens)

# Apply preprocessing to the messages
df['cleaned_message'] = df['message'].apply(preprocess_text)

# Display cleaned messages
df[['message', 'cleaned_message']].head()
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.11/dist-packages (3.9.1)
Requirement already satisfied: regex in /usr/local/lib/python3.11/dist-packages (2024.11.6)
Requirement already satisfied: click in /usr/local/lib/python3.11/dist-packages (from nltk) (8.2.1)
Requirement already satisfied: joblib in /usr/local/lib/python3.11/dist-packages (from nltk) (1.5.1)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from nltk) (4.67.1)
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
  label                                            message
0   ham  Go until jurong point, crazy.. Available only ...
1   ham                      Ok lar... Joking wif u oni...
2  spam  Free entry in 2 a wkly comp to win FA Cup fina...
3   ham  U dun say so early hor... U c already then say...
4   ham  Nah I don't think he goes to usf, he lives aro...
```

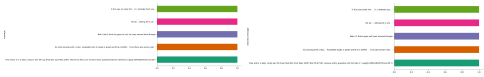
1 to 5 of 5 entries Filter

index	message	cleaned_message
0	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...	Go jurong point crazy .. Available bugis n great world la e buffet ... Cine got amore wat ...
1	Ok lar... Joking wif u oni...	Ok lar ... Joking wif u oni ...
2	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's	Free entry 2 wkly comp win FA Cup final tkts 21st May 2005 Text FA 87121 receive entry question std txt rate C 's apply 08452810075over18 's
3	U dun say so early hor... U c already then say...	U dun say early hor ... U c already say ...
4	Nah I don't think he goes to usf, he lives around here though	Nah n't think goes usf lives around though

Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

Categorical distributions



2-d categorical distributions