In [1]:
```
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\user\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Out[1]:
```
True
```

In [2]:
```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\user\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[2]:
```
True
```

In [3]:
```
#pip install PyPDF2
```

In [4]:
```
import PyPDF2
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [30]:
```
# Step 1: Extract Text from PDF
def extract_text_from_pdf(pdf_path):
    text = ""
    with open(pdf_path, "rb") as file:
        reader = PyPDF2.PdfReader(file)
        for page_num in range(len(reader.pages)):
            page = reader.pages[page_num]
            text += page.extract_text()
    return text
```

In [31]:
```
# Step 2: Preprocess Text
def preprocess_text(text):
    tokens = word_tokenize(text.lower())
    stop_words = set(stopwords.words("english"))
    tokens = [word for word in tokens if word.isalnum() and word not in stop_words]
    return " ".join(tokens)
```

In [32]:
```
# Step 3: Calculate Importance Scores using TF-IDF
def calculate_tfidf(texts):
    tfidf_vectorizer = TfidfVectorizer()
    tfidf_matrix = tfidf_vectorizer.fit_transform(texts)
    feature_names = tfidf_vectorizer.get_feature_names_out()
    return tfidf_matrix, feature_names
```

In [33]:
```
# Step 4: Select Top Keywords
def select_top_keywords(tfidf_matrix, feature_names, top_n=10):
    top_keywords = []
    for doc in tfidf_matrix:
        doc = doc.toarray().flatten()
        indices = doc.argsort()[-top_n:][::-1]
        top_keywords.append([feature_names[i] for i in indices])
    return top_keywords
```

In [34]:
```
# Example usage
pdf_path = "C://Users//user//Downloads//Data-Visualization-Tools.pdf"
text = extract_text_from_pdf(pdf_path)
preprocessed_text = preprocess_text(text)
tfidf_matrix, feature_names = calculate_tfidf([preprocessed_text])
top_keywords = select_top_keywords(tfidf_matrix, feature_names)
```

In [35]:
```
print("Top Keywords:")
for keywords in top_keywords:
    print(keywords)
```

```
Top Keywords:
['data', 'chart', 'visualization', 'dashboard', 'fig', 'tools', 'map', 'tableau', 'step', 'google']
```

In [ ]:
```

```