Es wird entschieden keine der Spalten zu löschen, da diese für eine spätere Analyse möglicherweise wichtig sind. Vielmehr sollen für die Durchführung des LDAs lediglich englische Lieder im Datensatz behalten werden.

In [ ]:
```
# Importe
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

import re
from wordcloud import WordCloud

import gensim
from gensim.utils import simple_preprocess
import nltk
from nltk.corpus import stopwords
import gensim.corpora as corpora
```

Data Understanding mit allen Daten

In [ ]:
```
understanding_data = pd.read_csv('C:\\Users\sdo\Studienarbeit\DIE_DATEN\datensatz.c

understanding_data
```

Out[ ]:

	Artist	Song	Genre	Language	Lyrics
0	12 stones	world so cold	Rock	en	It starts with pain, followed by hate\nFueled ...
1	12 stones	broken	Rock	en	Freedom!\nAlone again again alone\nPatiently w...
2	12 stones	3 leaf loser	Rock	en	Biting the hand that feeds you, lying to the v...
3	12 stones	anthem for the underdog	Rock	en	You say you know just who I am\nBut you can't ...
4	12 stones	adrenaline	Rock	en	My heart is beating faster can't control these...
...	...	...	...	...	...
290178	bobby womack	i wish he didn t trust me so much	R&B	en	I'm the best friend he's got I'd give him the ...
290179	bad boys blue	i totally miss you	Pop	en	Bad Boys Blue "I Totally Miss You" I did you w...
290180	celine dion	sorry for love	Pop	en	Forgive me for the things That I never said to...
290181	dan bern	cure for aids	Indie	en	The day they found a cure for AIDS The day the...
290182	crawdad republic	iceberg meadows	Pop	en	Fourth of July has come, it's custom that we g...

290183 rows × 5 columns

Untersuchung Anzahl unterschiedlicher Sprachen

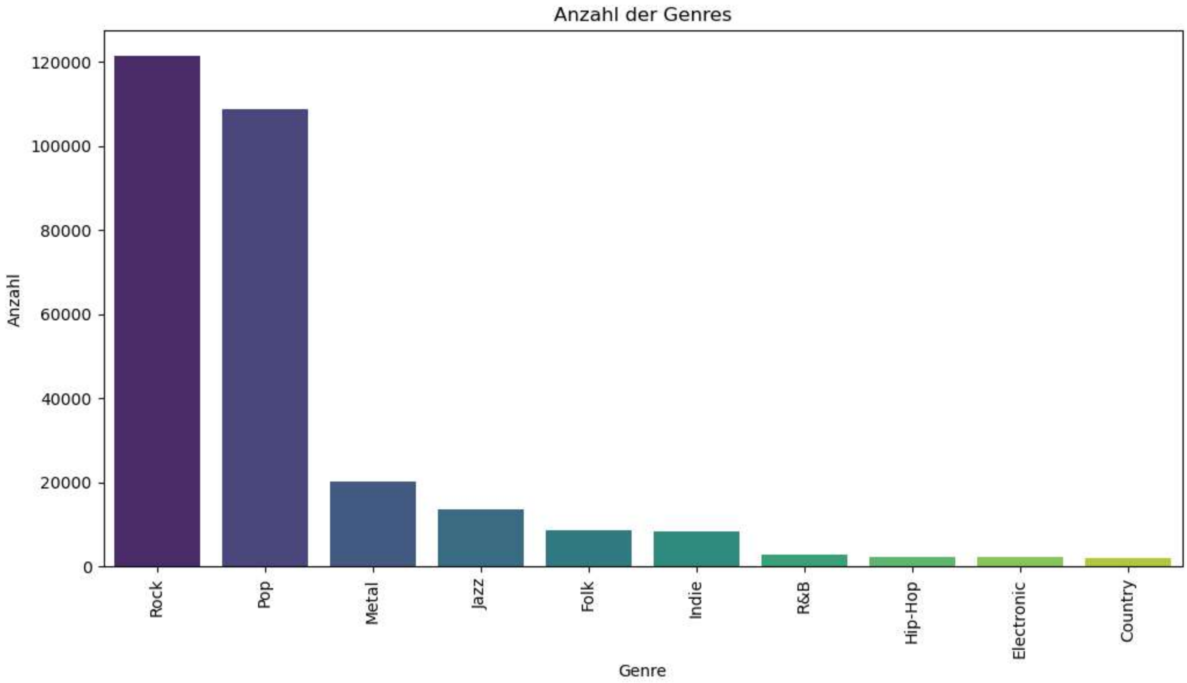
In [ ]:
```
understanding_data['Language'].value_counts()
```

Out[ ]:
```
Language
en    250197
pt     30102
es      3892
ro      1184
it       808
id       737
fr       644
de       478
sw       304
tl       241
so       229
cy       226
ca       137
tr       116
nl       116
sk        98
hr        97
no        93
sl        77
af        77
da        71
sv        61
et        58
fi        54
pl        24
cs        17
sq        15
hu        10
vi         7
ru         4
lt         2
lv         2
ko         1
Name: count, dtype: int64
```

Untersuchung Anzahl unterschiedlicher Genres

In [ ]:
```
type1_counts = understanding_data['Genre'].value_counts().reset_index()
type1_counts.columns = ['Genre', 'Count']

plt.figure(figsize=(12, 6))
sns.barplot(data=type1_counts, x='Genre', y='Count', palette='viridis')
plt.xlabel('Genre')
plt.ylabel('Anzahl')
plt.title('Anzahl der Genres')
plt.xticks(rotation=90)  # Für die bessere Lesbarkeit der Typen

plt.show()
```

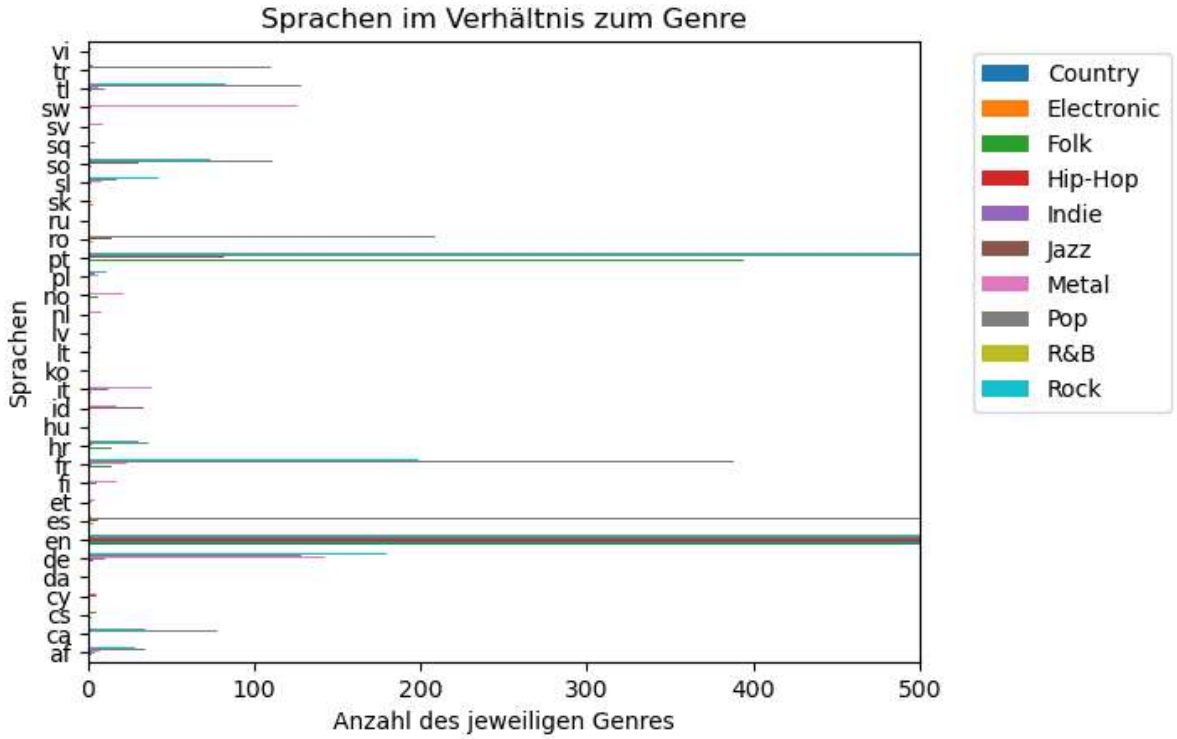
```
c:\Users\sdo\AppData\Local\anaconda3\envs\DataScience\Lib\site-packages\seaborn\_o
ldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be remo
ved in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
c:\Users\sdo\AppData\Local\anaconda3\envs\DataScience\Lib\site-packages\seaborn\_o
ldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be remo
ved in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
c:\Users\sdo\AppData\Local\anaconda3\envs\DataScience\Lib\site-packages\seaborn\_o
ldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be remo
ved in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```



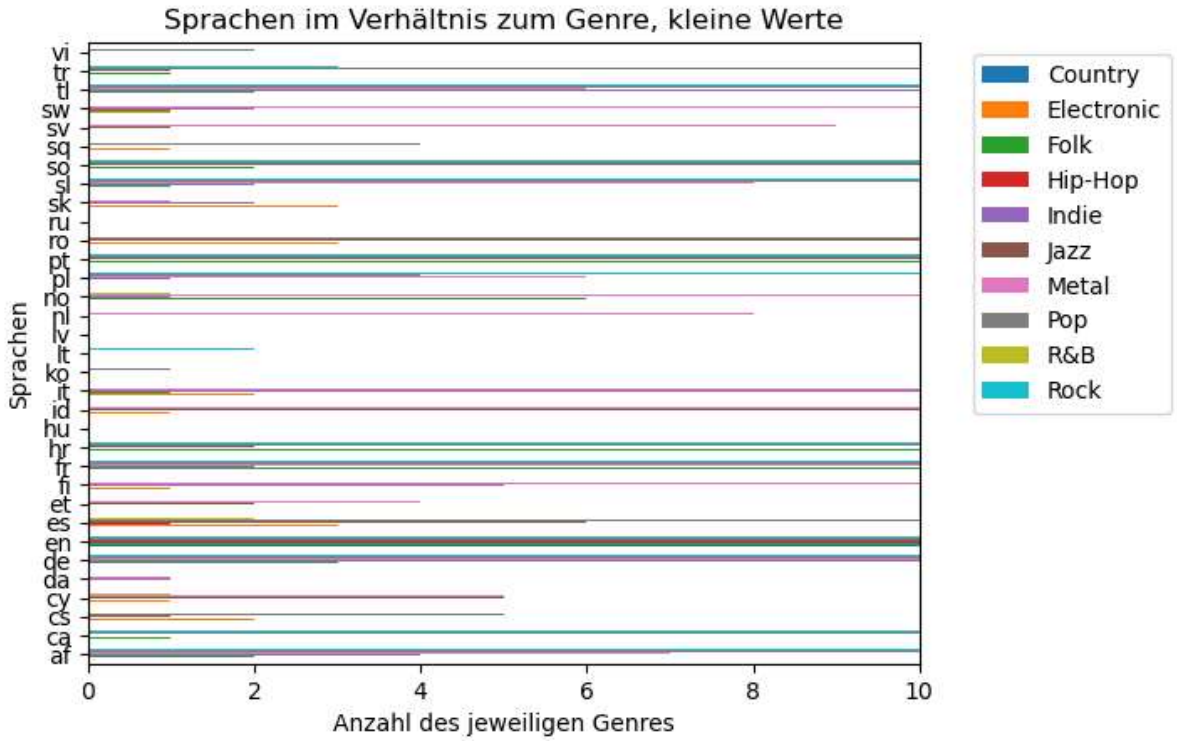
Sprachen im Verhältnis zum Genre

In [ ]:

```
crosstab = pd.crosstab(understanding_data['Language'], understanding_data['Genre'])
crosstab.plot(kind='barh').legend(bbox_to_anchor=(1.05, 1), loc='upper left')
plt.xlabel('Anzahl des jeweiligen Genres')
plt.ylabel('Sprachen')
plt.xlim(0,500)
plt.title('Sprachen im Verhältnis zum Genre')
plt.show()
```



In [ ]:
```
crosstab = pd.crosstab(understanding_data['Language'], understanding_data['Genre'])
crosstab.plot(kind='barh').legend(bbox_to_anchor=(1.05, 1), loc='upper left')
plt.xlabel('Anzahl des jeweiligen Genres')
plt.ylabel('Sprachen')
plt.xlim(0,10)
plt.title('Sprachen im Verhältnis zum Genre, kleine Werte')
plt.show()
```



Anzahl der Genres für die englische Sprache

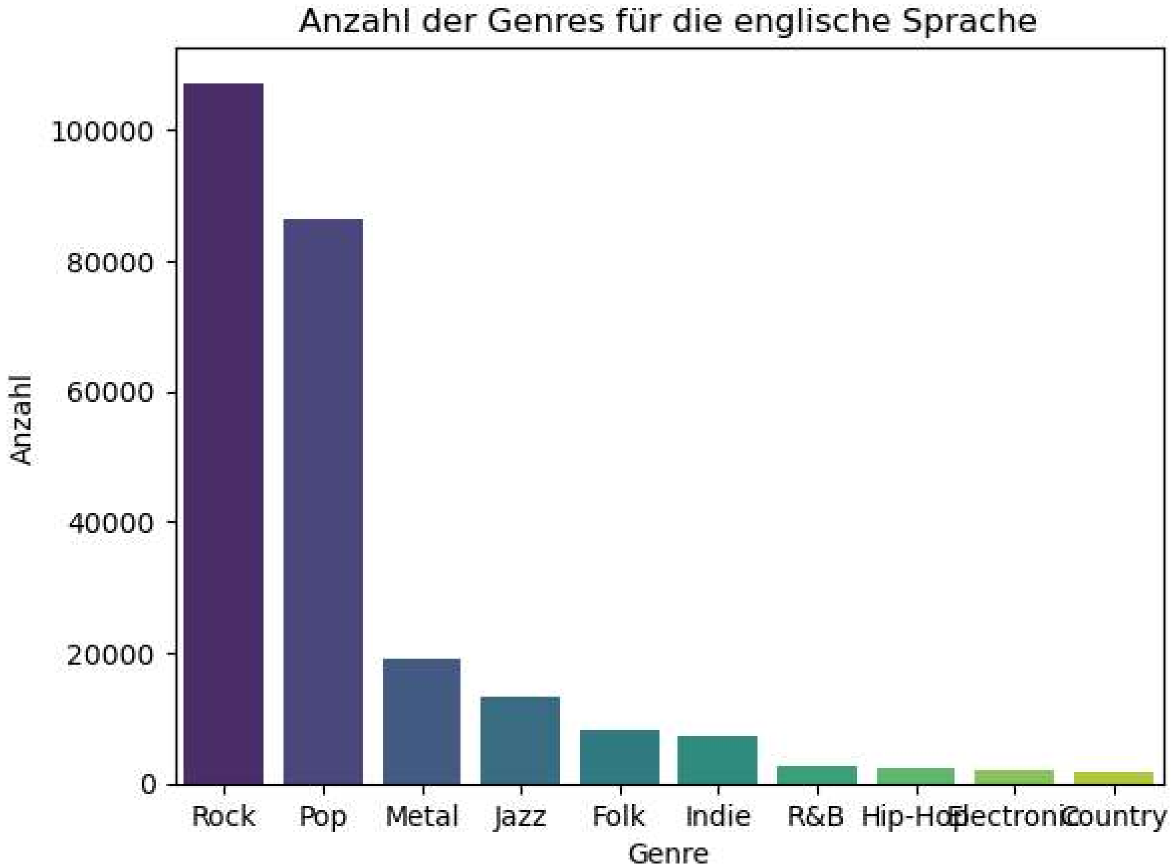
In [ ]:
```
en_Genre = understanding_data[understanding_data['Language'] == 'en']['Genre'].valu
en_Genre.columns = ['Language', 'Anzahl Genre']
sns.barplot(data=en_Genre, x='Language', y='Anzahl Genre', palette='viridis')
plt.xlabel('Genre')
```

```python
plt.ylabel('Anzahl')
plt.title('Anzahl der Genres für die englische Sprache')
plt.xticks(rotation=0)
```

```
c:\Users\sdo\AppData\Local\anaconda3\envs\DataScience\Lib\site-packages\seaborn\_o
ldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be remo
ved in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
c:\Users\sdo\AppData\Local\anaconda3\envs\DataScience\Lib\site-packages\seaborn\_o
ldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be remo
ved in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
c:\Users\sdo\AppData\Local\anaconda3\envs\DataScience\Lib\site-packages\seaborn\_o
ldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be remo
ved in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```

Out[ ]:

```
(array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
 [Text(0, 0, 'Rock'),
  Text(1, 0, 'Pop'),
  Text(2, 0, 'Metal'),
  Text(3, 0, 'Jazz'),
  Text(4, 0, 'Folk'),
  Text(5, 0, 'Indie'),
  Text(6, 0, 'R&B'),
  Text(7, 0, 'Hip-Hop'),
  Text(8, 0, 'Electronic'),
  Text(9, 0, 'Country')])
```



In [ ]:

```python
understanding_data[understanding_data['Language'] == 'en']['Genre'].value_counts()
```

Out[ ]:
```
Genre
Rock          107145
Pop            86298
Metal          19133
Jazz           13314
Folk            8169
Indie           7240
R&B             2765
Hip-Hop         2238
Electronic      2005
Country         1890
Name: count, dtype: int64
```

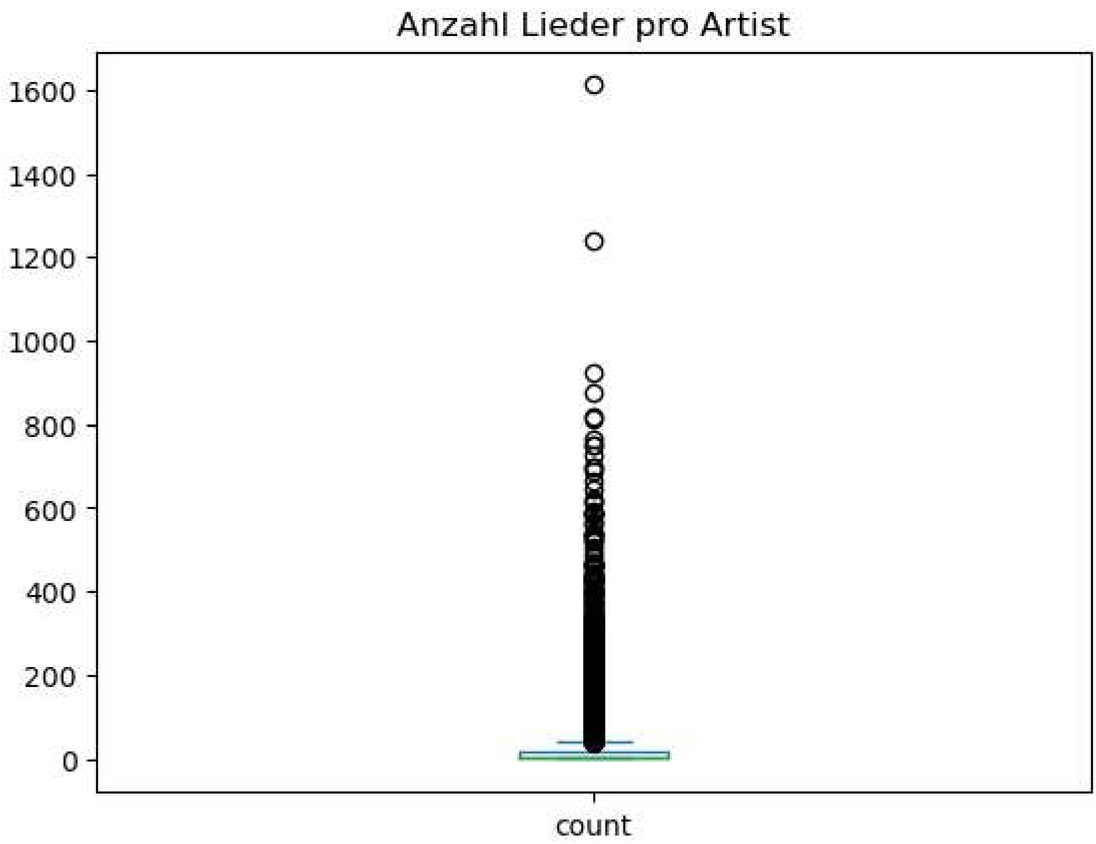
Untersuchung der Artisten

In [ ]:
```
artist_counts = understanding_data['Artist'].value_counts()
more_than_100_count = (artist_counts > 100).sum()
print(f'Number of artists with more than 100 occurrences: {more_than_100_count}')

understanding_data['Artist'].value_counts()
```

```
Number of artists with more than 100 occurrences: 802
```

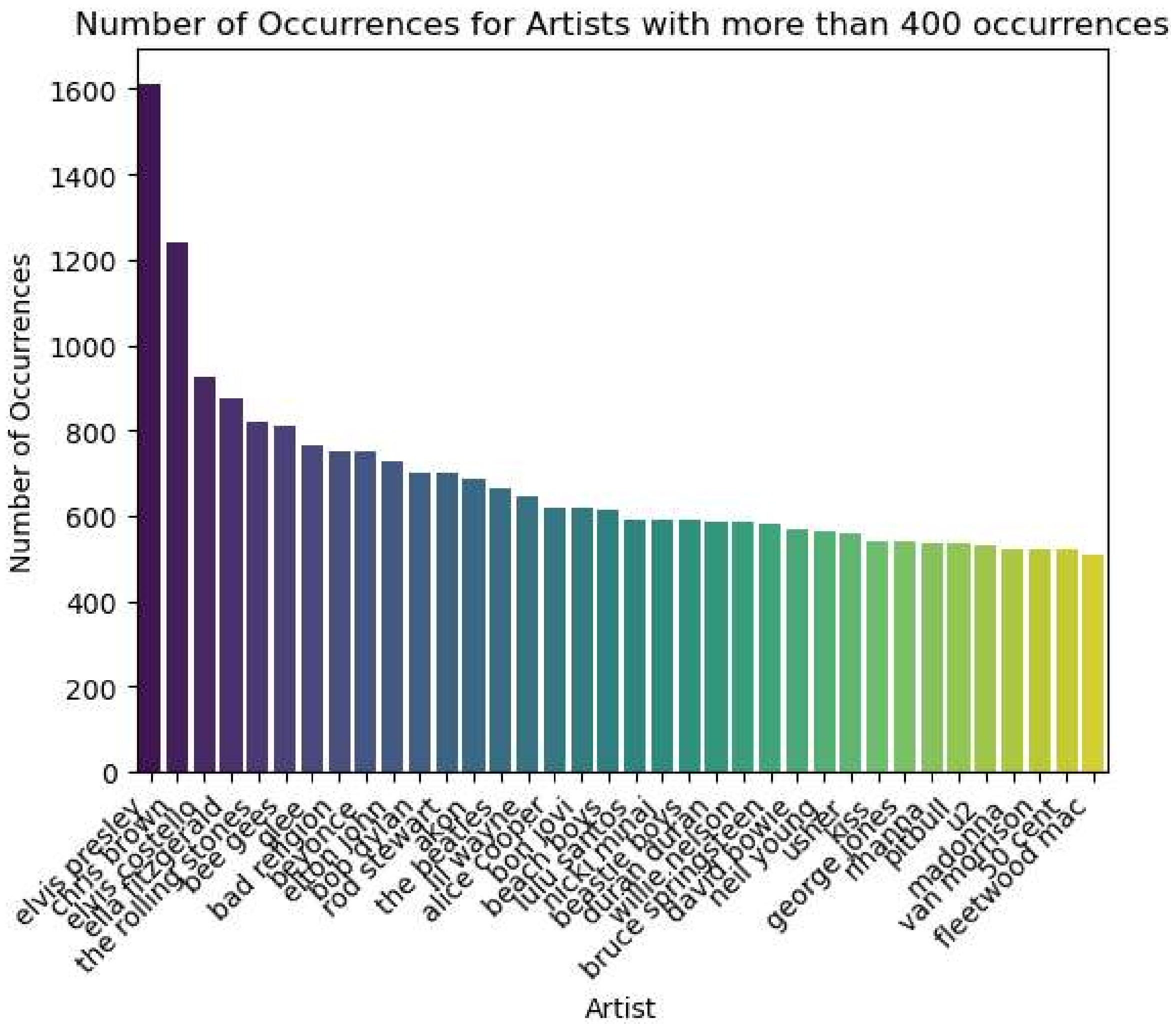
Out[ ]:
```
Artist
elvis presley           1611
chris brown             1239
elvis costello           923
ella fitzgerald          874
the rolling stones       820
                        ...
the gregory brothers       1
flamingokvintetten         1
fjeld                      1
danceplaycreate            1
crawdad republic           1
Name: count, Length: 11152, dtype: int64
```

In [ ]:
```
artist_counts = understanding_data['Artist'].value_counts()
artist_counts.plot(kind = 'box')
plt.title('Anzahl Lieder pro Artist')
plt.show()
```



In [ ]:
```
# Assuming 'Artist' is the index after using value_counts
artist_counts = understanding_data['Artist'].value_counts()

# Select artists with more than 400 occurrences
more_than_600_count = artist_counts[artist_counts > 500]

# Create a bar plot
sns.barplot(x=more_than_600_count.index, y=more_than_600_count.values, palette='vir

# Set plot labels and title
plt.xlabel('Artist')
plt.ylabel('Number of Occurrences')
plt.title('Number of Occurrences for Artists with more than 400 occurrences')
plt.xticks(rotation=45, ha='right')  # Rotate x-axis labels for better visibility

# Show the plot
plt.show()
```

```
c:\Users\sdo\AppData\Local\anaconda3\envs\DataScience\Lib\site-packages\seaborn\_o
ldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be remo
ved in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
c:\Users\sdo\AppData\Local\anaconda3\envs\DataScience\Lib\site-packages\seaborn\_o
ldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be remo
ved in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
c:\Users\sdo\AppData\Local\anaconda3\envs\DataScience\Lib\site-packages\seaborn\_o
ldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be remo
ved in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```



Strichprobentest für Genrezuordnung bei Künstlern

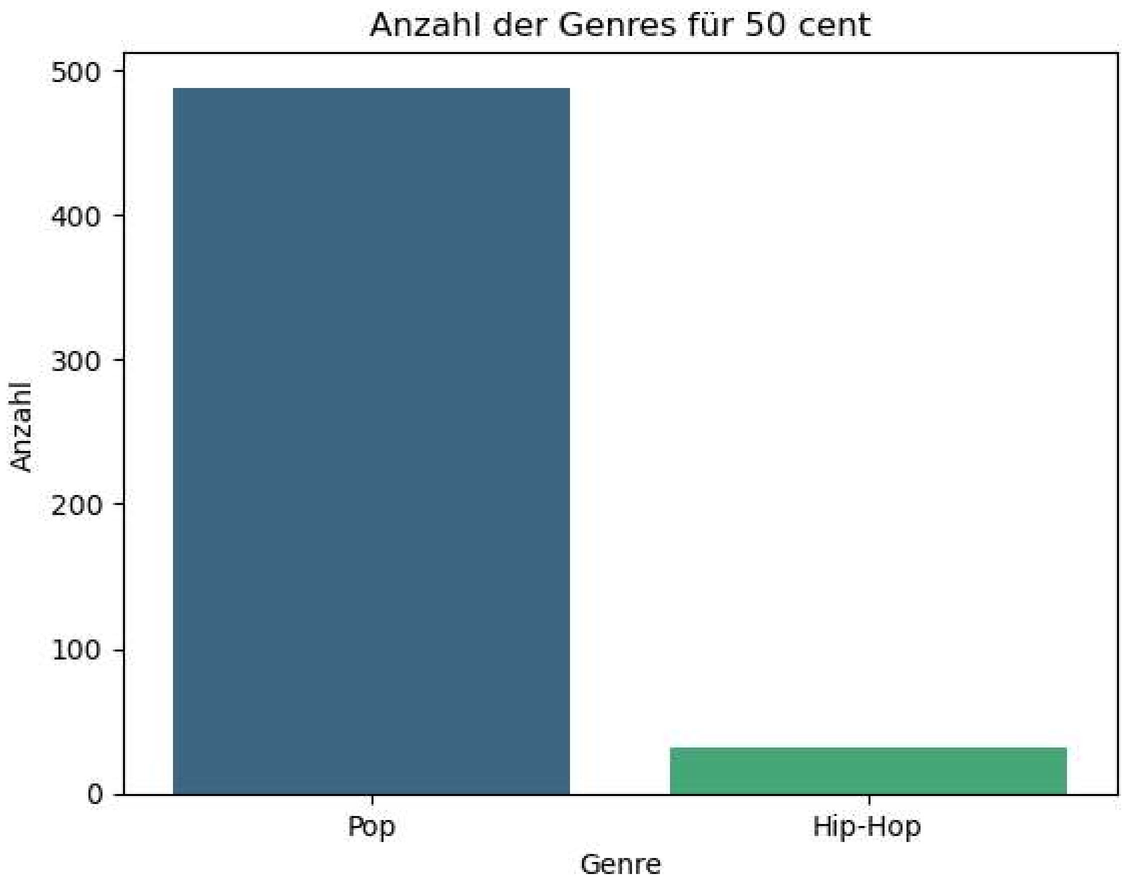
In [ ]:

```
en_Genre = understanding_data[understanding_data['Artist'] == '50 cent']['Genre'].v
en_Genre.columns = ['Artist', 'Anzahl Genre']
sns.barplot(data=en_Genre, x='Artist', y='Anzahl Genre', palette='viridis')
plt.xlabel('Genre')
plt.ylabel('Anzahl')
plt.title('Anzahl der Genres für 50 cent')
plt.xticks(rotation=0)
```

```
c:\Users\sdo\AppData\Local\anaconda3\envs\DataScience\Lib\site-packages\seaborn\_o
ldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be remo
ved in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
c:\Users\sdo\AppData\Local\anaconda3\envs\DataScience\Lib\site-packages\seaborn\_o
ldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be remo
ved in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
c:\Users\sdo\AppData\Local\anaconda3\envs\DataScience\Lib\site-packages\seaborn\_o
ldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be remo
ved in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```

Out[ ]:

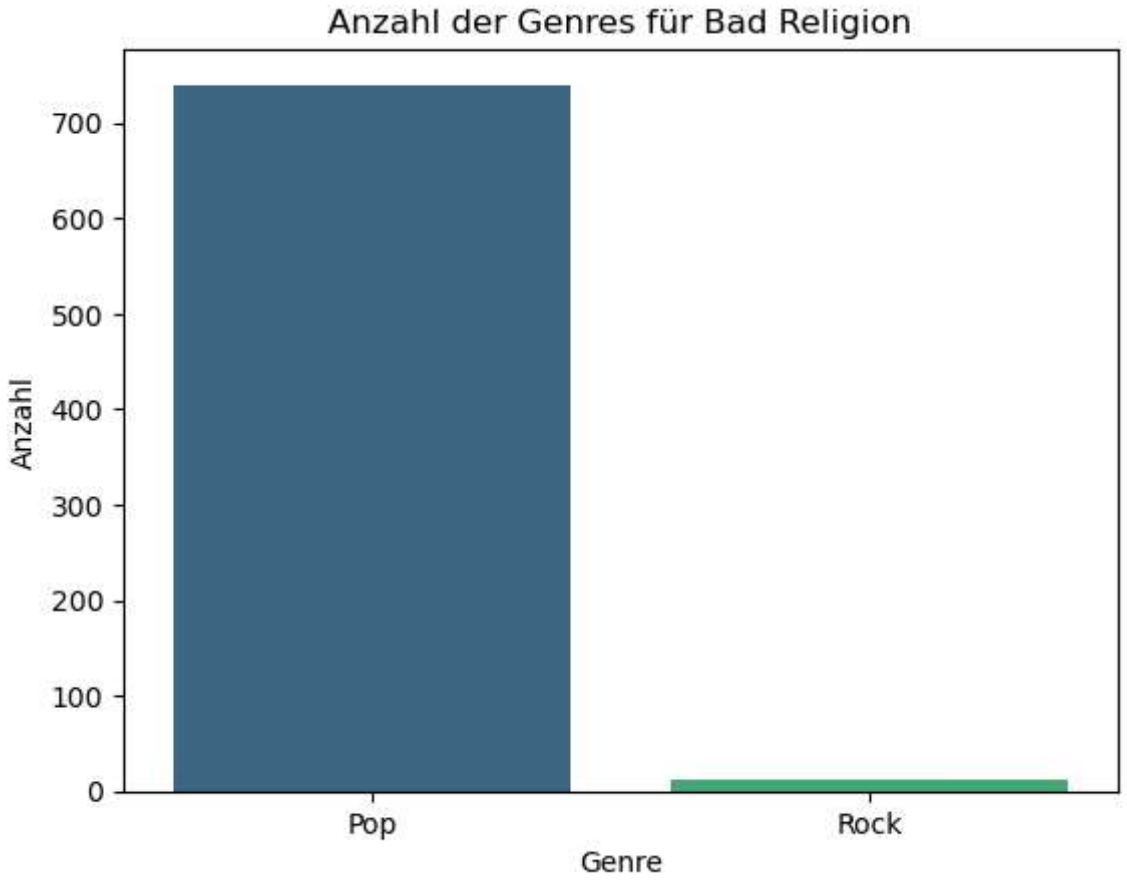
```
(array([0, 1]), [Text(0, 0, 'Pop'), Text(1, 0, 'Hip-Hop')])
```



In [ ]:
```
en_Genre = understanding_data[understanding_data['Artist'] == 'bad religion']['Genr
en_Genre.columns = ['Artist', 'Anzahl Genre']
sns.barplot(data=en_Genre, x='Artist', y='Anzahl Genre', palette='viridis')
plt.xlabel('Genre')
plt.ylabel('Anzahl')
plt.title('Anzahl der Genres für Bad Religion')
plt.xticks(rotation=0)
```

```
c:\Users\sdo\AppData\Local\anaconda3\envs\DataScience\Lib\site-packages\seaborn\_o
ldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be remo
ved in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
c:\Users\sdo\AppData\Local\anaconda3\envs\DataScience\Lib\site-packages\seaborn\_o
ldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be remo
ved in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
c:\Users\sdo\AppData\Local\anaconda3\envs\DataScience\Lib\site-packages\seaborn\_o
ldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be remo
ved in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```

Out[ ]:
```
(array([0, 1]), [Text(0, 0, 'Pop'), Text(1, 0, 'Rock')])
```



Untersuchung der Songs

In [ ]:
```
total_rows_count = len(understanding_data['Song'])
print(f'Total number of rows in the column: {total_rows_count}')
```

```
Total number of rows in the column: 290183
```

In [ ]:
```
len(understanding_data['Song'].unique())
```

Out[ ]:
```
164358
```

In [ ]:
```
duplicate_rows = understanding_data.duplicated()

# Count the number of duplicate rows
duplicate_rows_count = duplicate_rows.sum()

print(f'Number of duplicate rows: {duplicate_rows_count}')
```

```
Number of duplicate rows: 28477
```

Anmerkung zu den Daten

- Die meisten Lieder sind sowieso auf Englisch, die anderen können also problemlos rausgefiltert werden
- Die häufigsten Genres sind Rock und Pop, zu den anderen gibt es ein deutliches Ungleichgewicht
- Es gibt nur 10 verschiedene Genres, dabei werden auch Artisten, die anderen Genres zuzuordnen sind (siehe Beispiele von 50 cent (eigentlich Rap) und Bad Religion

(eigentlich Punk)) anderen Genres zugeordet, zu denen sie eigentlich gar nicht passen, das wird die Analyse wahrscheinlich verfälschen?

- Es gibt 11152 verschiedene Künstler, davon sind von vielen deutlich mehr als 100 Lieder im Datensatz, einige haben aber nur einen Song
- In dem Datensatz gibt es über 290 000 Lieder, davon sind aber nur 164 358 unique, exakte Duplikate sind aber nur 28 477 Zeilen, da muss herausgefunden werden, was mit den anderen Zeilen los ist

Data Cleaning

In [ ]:
```
# song_data = pd.read_csv('C:\\Users\sdo\Studienarbeit\DIE_DATEN\datensatz.csv')

# song_data = song_data[(song_data['Language'] == 'en')]

# song_data.to_csv("outData.csv", index=False)

song_data = pd.read_csv('C:\\Users\sdo\Studienarbeit\outData.csv')

song_data
```

Out[ ]:

	Artist	Song	Genre	Language	Lyrics
0	12 stones	world so cold	Rock	en	It starts with pain, followed by hate\nFueled ...
1	12 stones	broken	Rock	en	Freedom!\nAlone again again alone\nPatiently w...
2	12 stones	3 leaf loser	Rock	en	Biting the hand that feeds you, lying to the v...
3	12 stones	anthem for the underdog	Rock	en	You say you know just who I am\nBut you can't ...
4	12 stones	adrenaline	Rock	en	My heart is beating faster can't control these...
...	...	...	...	...	...
250192	bobby womack	i wish he didn t trust me so much	R&B	en	I'm the best friend he's got I'd give him the ...
250193	bad boys blue	i totally miss you	Pop	en	Bad Boys Blue "I Totally Miss You" I did you w...
250194	celine dion	sorry for love	Pop	en	Forgive me for the things That I never said to...
250195	dan bern	cure for aids	Indie	en	The day they found a cure for AIDS The day the...
250196	crawdad republic	iceberg meadows	Pop	en	Fourth of July has come, it's custom that we g...

250197 rows × 5 columns

In [ ]:
```
# Remove the columns
song_data = song_data[['Genre', 'Lyrics']]
# Print out the first rows of papers
song_data
```