

**REPRESENTATION OF SPORTS
RELATED DOPING IN SWISS MEDIA
TRENDANALYSIS**

Aaron Steiner, Livia Köppel, Silvan Lachmuth

Keine Untersuchung der Wada

23 chinesische Top-Schwimmer 2021 positiv getestet

Samstag, 20.04.2024, 11:27 Uhr

Aktualisiert um 15:29 Uhr

↗ TEILEN

- 23 chinesische Top-Schwimmerinnen und -Schwimmer wurden vor Olympia 2021 positiv auf ein verbotenes Mittel getestet.
- Die Welt-Anti-Doping-Agentur (Wada) verzichtete laut Recherchen auf eine eigene Untersuchung.
- Travis Tygart, Chef der US-Anti-Doping-Agentur Usada, sprach gegenüber der ARD-Dopingredaktion von «schockierenden Enthüllungen».

Nach Crystal-Meth-Befund

Strafverfahren gegen Handball-Goalie Portner eingestellt

Beim Schweizer Handball-Goalie wurden in einer Urinprobe Spuren von Methamphetamine gefunden. Immerhin drohen ihm strafrechtlich keine weiteren Konsequenzen mehr.



RESEARCH QUESTION



How is sports-related doping represented in the Swiss media?

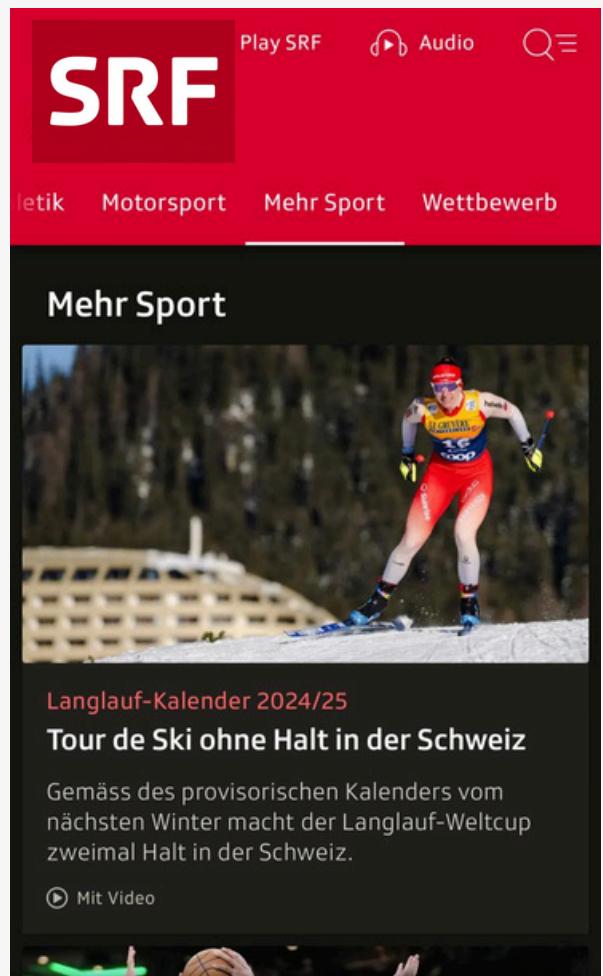
How often is sports-related doping mentioned in the Swiss media?

When do we discuss sports-related doping in the Swiss media?

DATA

Source: **Swissdox**

Online Journals:



Period: **May 2014 - May 2024**

Keywords:

 **Dopingkontrolle**
 **Dopingtest**
 **Blutdoping**
 **Anti-Doping**
 **Steroide**
 **Anabolika**
 **Testosteron**

COMPUTATIONAL METHODS

```
import pandas as pd # Importiere die Pandas-Bibliothek und verwende pd als Alias.  
from bs4 import BeautifulSoup # Importiere die BeautifulSoup-Bibliothek zum Analysieren von HTML und XML.  
import plotnine # Importiere die Plotnine-Bibliothek für die Datenvisualisierung.  
  
pfad ="/home/aaron/documents/KED2024/DATEN_final.tsv" # Pfad zur Datei mit den Daten.  
  
# Lese die Daten aus der angegebenen Datei unter Verwendung von Tabulatoren als Trennzeichen.  
df1 = pd.read_csv(pfad, sep="\t")  
  
# Liste von Spalten, die aus dem DataFrame entfernt werden sollen.  
drop = [  
    "id",  
    "#regional",  
    "doctype",  
    "doctype_description",  
    "dateline",  
    "language",  
    "subhead",  
    "content_id",  
    "medium_name",  
]  
  
# Erstelle ein neues DataFrame mit den ausgewählten Spalten entfernt.  
df_sub = df1.drop(columns=drop)
```

COMPUTATIONAL METHODS

```
# Funktion zum Entfernen von HTML-Tags aus einem Text.
def remove_html(text):
    soup = BeautifulSoup(text) # Erzeuge ein BeautifulSoup-Objekt für den Text.

    try:
        soup.au.decompose() # Entferne speziell das <au>-Tag, falls vorhanden.
    except AttributeError:
        pass

    text = soup.get_text(separator="\n", strip=True) # Erhalte nur den Text, ohne HTML-Tags.

    return text

# Wende die remove_html-Funktion auf die Spalte "content" an, um HTML-Tags zu entfernen.
df_sub["content"] = df_sub["content"].apply(lambda x: remove_html(x))

# Konvertiere die Spalte "pubtime" in das Datumsformat und setze die Zeitzone auf UTC.
df_sub["pubtime"] = pd.to_datetime(df_sub["pubtime"], utc=True)

# Füge eine neue Spalte "year" hinzu, die das Jahr aus der Spalte "pubtime" extrahiert.
df_sub['year'] = df_sub['pubtime'].dt.year

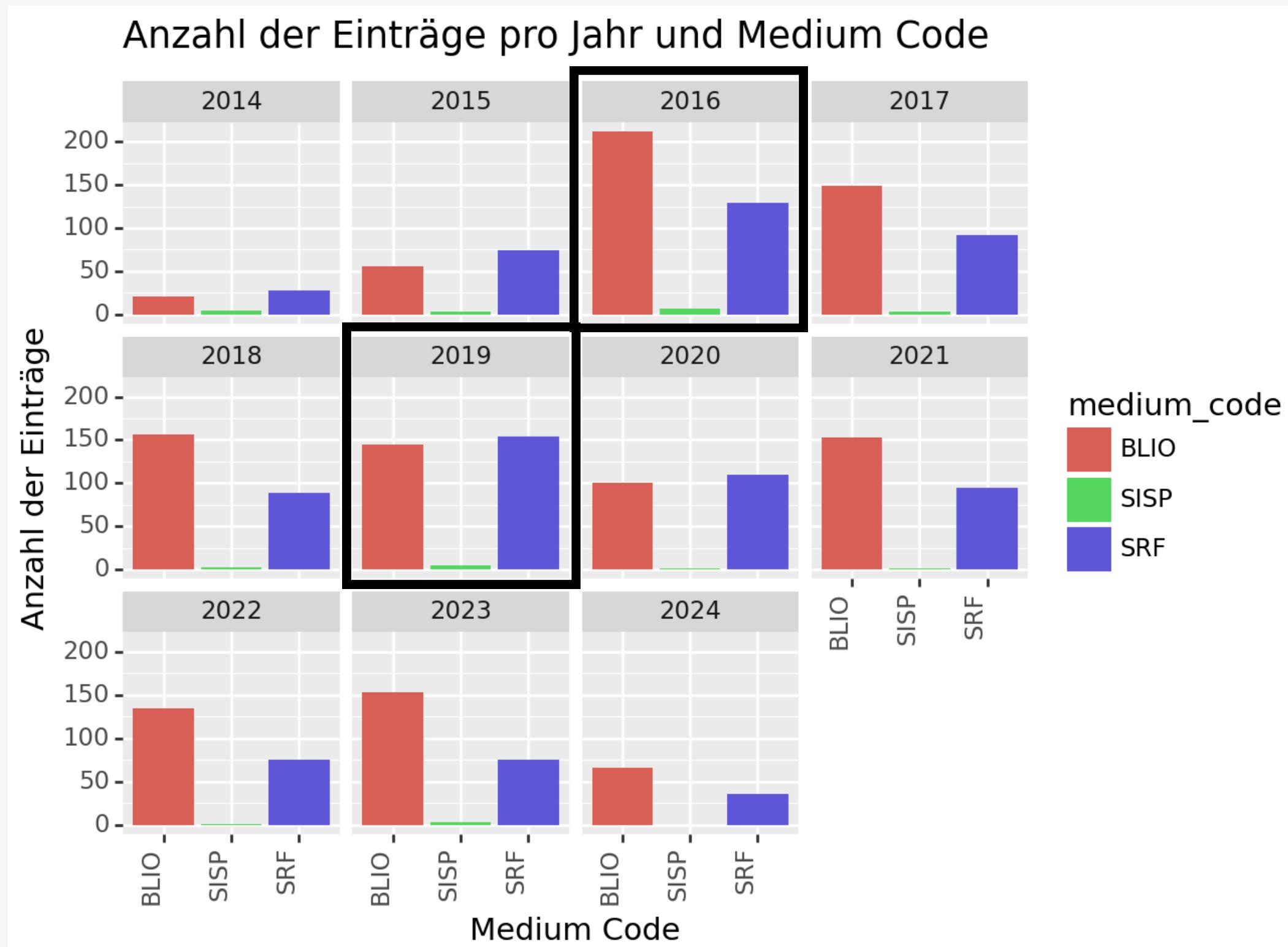
# Gruppieren die Daten nach Jahr und "medium_code" und zähle die Anzahl der Einträge.
count_data = df_sub.groupby(['year', 'medium_code']).size().reset_index(name='count')
```

COMPUTATIONAL METHODS

```
from plotnine import ggplot, aes, geom_bar, labs, facet_wrap, theme, element_text # Importiere benötigte Funktionen und Klassen aus plotnine.

# Erstelle ein Plot-Objekt mit ggplot und definiere Ästhetik (x, y, fill) sowie Datenquelle (count_data).
plot = (
    ggplot(count_data, aes(x='medium_code', y='count', fill='medium_code')) + # Definiere X, Y und Fill Ästhetik.
    geom_bar(stat='identity') + # Füge ein Balkendiagramm hinzu und behalte die Daten bei.
    facet_wrap('~year') + # Fasse die Grafiken nach Jahr zusammen.
    labs(title='Anzahl der Einträge pro Jahr und Medium Code', x='Medium Code', y='Anzahl der Einträge') + # Beschriftungen für den Plot.
    theme(axis_text_x=element_text(rotation=90, hjust=1)) # Dreht die X-Achsen-Beschriftungen für bessere Lesbarkeit
)
print(plot) # Drucke den Plot.
```

RESULTS



Medium with the
largest number
of entries:

Blick

Years with the
most entries:

2016 & 2019

COMPUTATIONAL METHODS

```
import pandas as pd
from plotnine import ggplot, aes, geom_col, labs, facet_wrap, theme, element_text

# Liste der zu zählenden Wörter
words_to_count = ["Steroide", "Anabolika", "Testosteron", "Blutdoping", "Anti-Doping", "Dopingkontrolle", "Dopingtest"]

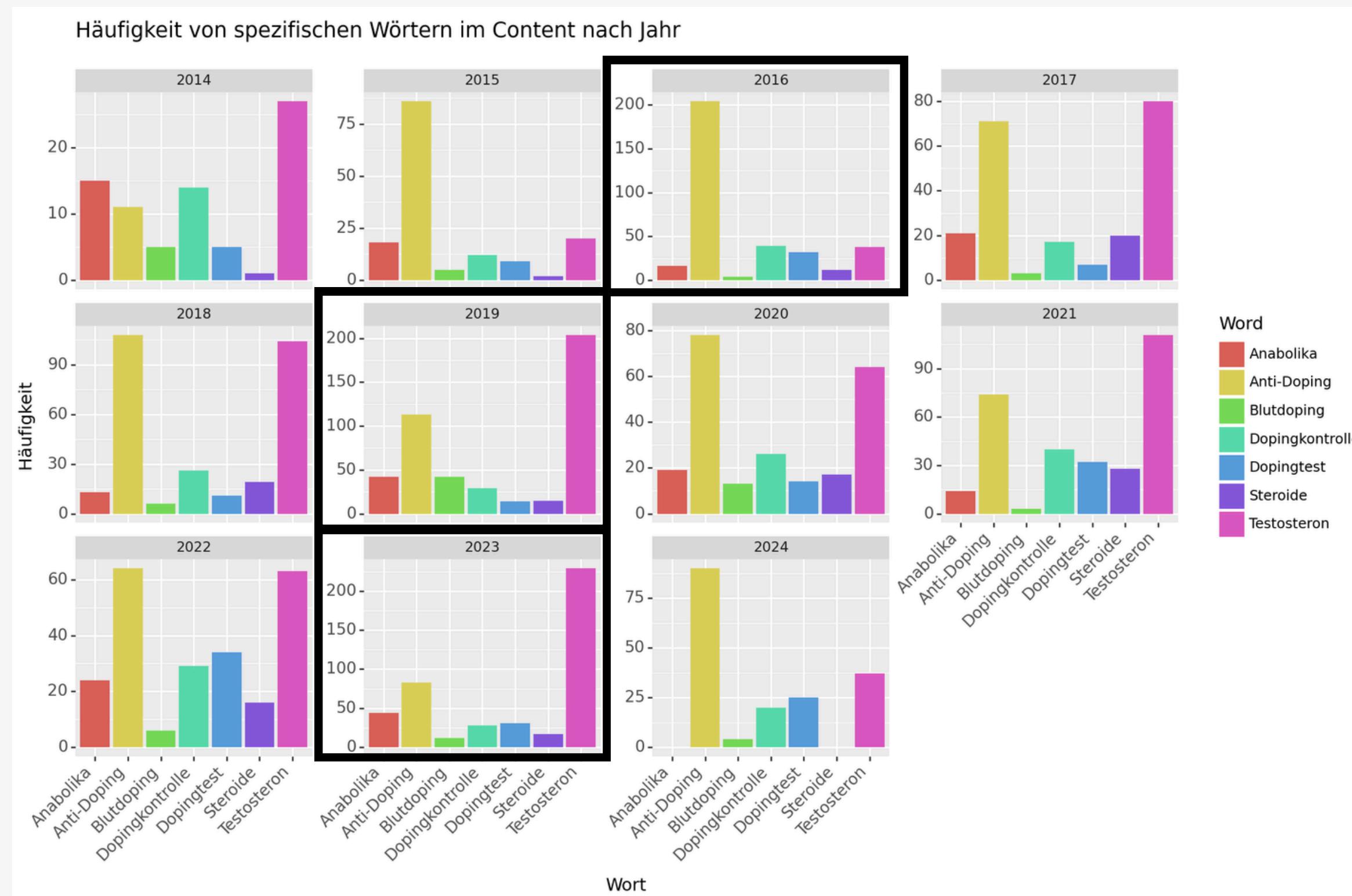
# Zählen der Häufigkeit jedes Wortes in der 'content' Spalte und Aggregation nach Jahr
for word in words_to_count:
    df_sub[word] = df_sub['content'].str.count(word)

# Gruppieren nach 'year' und Summieren der Häufigkeiten für jedes Wort
yearly_word_counts = df_sub.groupby('year')[words_to_count].sum().reset_index()

# Schmelzen der Daten für das Plotten
melted_data = pd.melt(yearly_word_counts, id_vars=['year'], value_vars=words_to_count, var_name='Word', value_name='Frequency')

# Visualisierung mit plotnine
plot = (
    ggplot(melted_data, aes(x='Word', y='Frequency', fill='Word')) +
    geom_col() +
    facet_wrap(~year, scales='free_y') + # Erlaubt unterschiedliche Y-Skalen für jede Facette
    labs(title='Häufigkeit von spezifischen Wörtern im Content nach Jahr', x='Wort', y='Häufigkeit') +
    theme(
        figure_size=(12, 8), # Größere Figurengröße
        axis_text_x=element_text(rotation=45, hjust=1, size=10), # Drehung und Größenanpassung der X-Achsen-Texte
        axis_text_y=element_text(size=10) # Größenanpassung der Y-Achsen-Texte
    )
)
print(plot)
```

RESULTS



Most frequent words:
Anti-Doping (200)
Testosteron (225)

Years with the
most frequent words:
2016, 2019 & 2023

COMPUTATIONAL METHODS

```
import pandas as pd
from plotnine import ggplot, aes, geom_line, labs, theme, element_text

# Liste der zu zählenden Wörter
words_to_count = ["Steroide", "Anabolika", "Testosteron", "Blutdoping", "Anti-Doping", "Dopingkontrolle", "Dopingtest"]

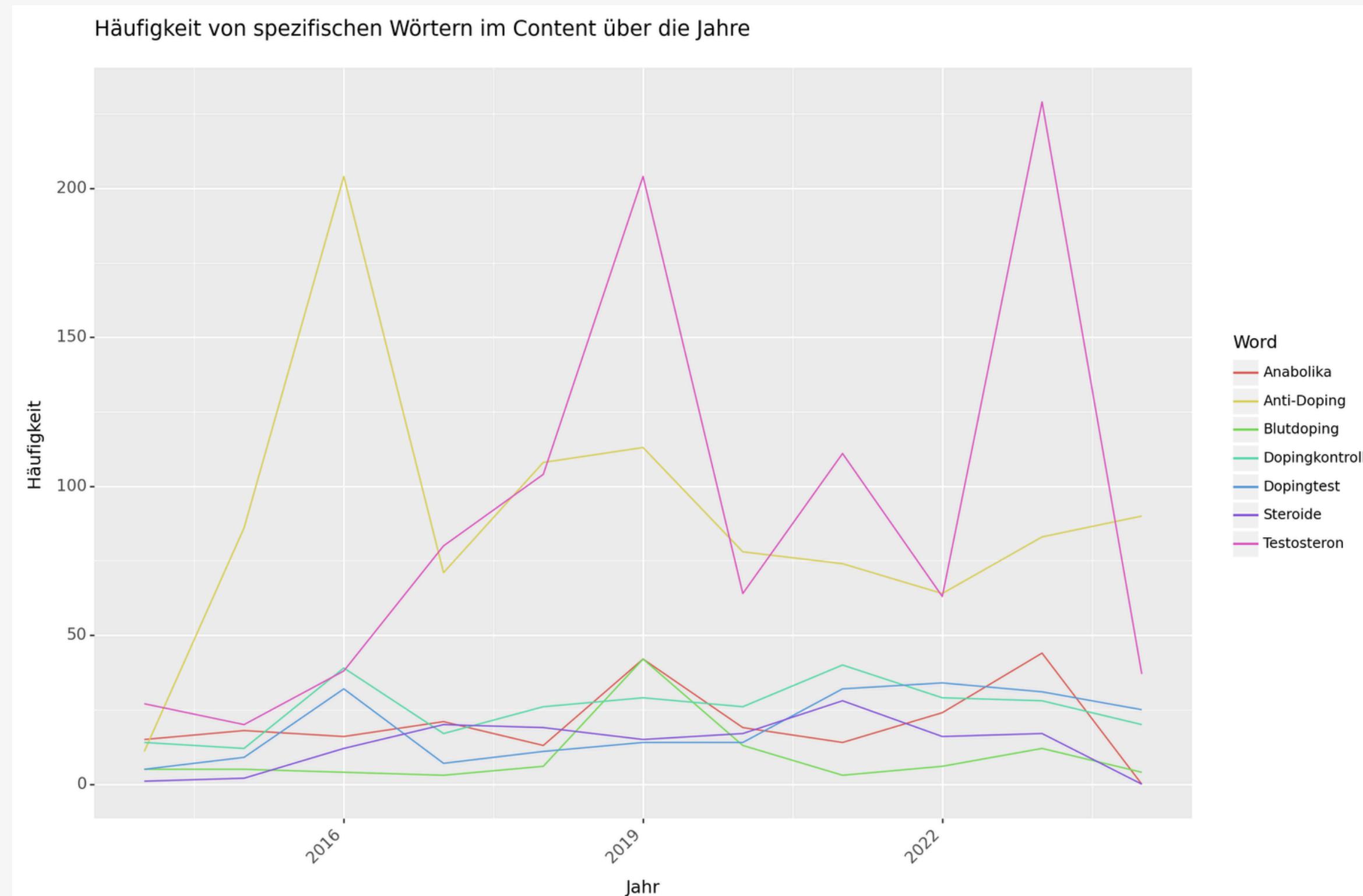
# Zählen der Häufigkeit jedes Wortes in der 'content' Spalte und Aggregation nach Jahr
for word in words_to_count:
    df_sub[word] = df_sub['content'].str.count(word)

# Gruppieren nach 'year' und Summieren der Häufigkeiten für jedes Wort
yearly_word_counts = df_sub.groupby('year')[words_to_count].sum().reset_index()

# Schmelzen der Daten für das Plotten
melted_data = pd.melt(yearly_word_counts, id_vars=['year'], value_vars=words_to_count, var_name='Word', value_name='Frequency')

# Visualisierung als Liniendiagramm
plot = (
    ggplot(melted_data, aes(x='year', y='Frequency', color='Word')) +
    geom_line() +
    labs(title='Häufigkeit von spezifischen Wörtern im Content über die Jahre', x='Jahr', y='Häufigkeit') +
    theme(
        figure_size=(12, 8),
        axis_text_x=element_text(rotation=45, hjust=1, size=10), # Drehung und Größenanpassung der X-Achsen-Texte
        axis_text_y=element_text(size=10) # Größenanpassung der Y-Achsen-Texte
    )
)
print(plot)
```

RESULTS



Most frequent words:

Anti-Doping (200)
Testosteron (225)

Years with the
most frequent words:

2016, 2019 & 2023

COMPUTATIONAL METHODS

```
import spacy

# Lade das deutsche NLP-Modell
nlp = spacy.load("de_core_news_sm")

# Funktion zum Extrahieren von Ländern und Entitäten
def extract_entities(text):
    doc = nlp(text)
    entities = {'LOC': [], 'ORG': [], 'PER': []}
    for ent in doc.ents:
        if ent.label_ == 'LOC': # Standorte und Länder
            entities['LOC'].append(ent.text)
        elif ent.label_ == 'ORG': # Organisationen
            entities['ORG'].append(ent.text)
        elif ent.label_ == 'PER': # Personen
            entities['PER'].append(ent.text)
    return entities

# Anwendung der Funktion auf jede Zeile in der Spalte 'content'
df_sub['entities'] = df_sub['content'].apply(extract_entities)

# Anzeigen der ersten paar Zeilen zur Überprüfung
print(df_sub['entities'].head())

from collections import Counter

# Erstellen eines Counters für alle Lokationen
location_counter = Counter()
df_sub['entities'].apply(lambda x: location_counter.update(x['LOC']))
```

```
0  {'LOC': ['Muskelkörper', 'Fitnesstraining', 'A...'}
1  {'LOC': ['Muskelkörper', 'Fitnesstraining', 'A...'}
2  {'LOC': ['Italiener', 'Italiener', 'Italien', ...}
3  {'LOC': ['Britin'], 'ORG': ['TAS', 'WM-Goldmed...'}
4  {'LOC': ['Allgäuerin', 'Ullrichs', 'Mallorca',...'}
Name: entities, dtype: object
```

COMPUTATIONAL METHODS

```
# Erstellen von Countern für Organisationen und Personen
organization_counter = Counter()
person_counter = Counter()

# Aktualisiere die Counter mit den extrahierten Daten
df_sub['entities'].apply(lambda x: organization_counter.update(x['ORG']))
df_sub['entities'].apply(lambda x: person_counter.update(x['PER']))

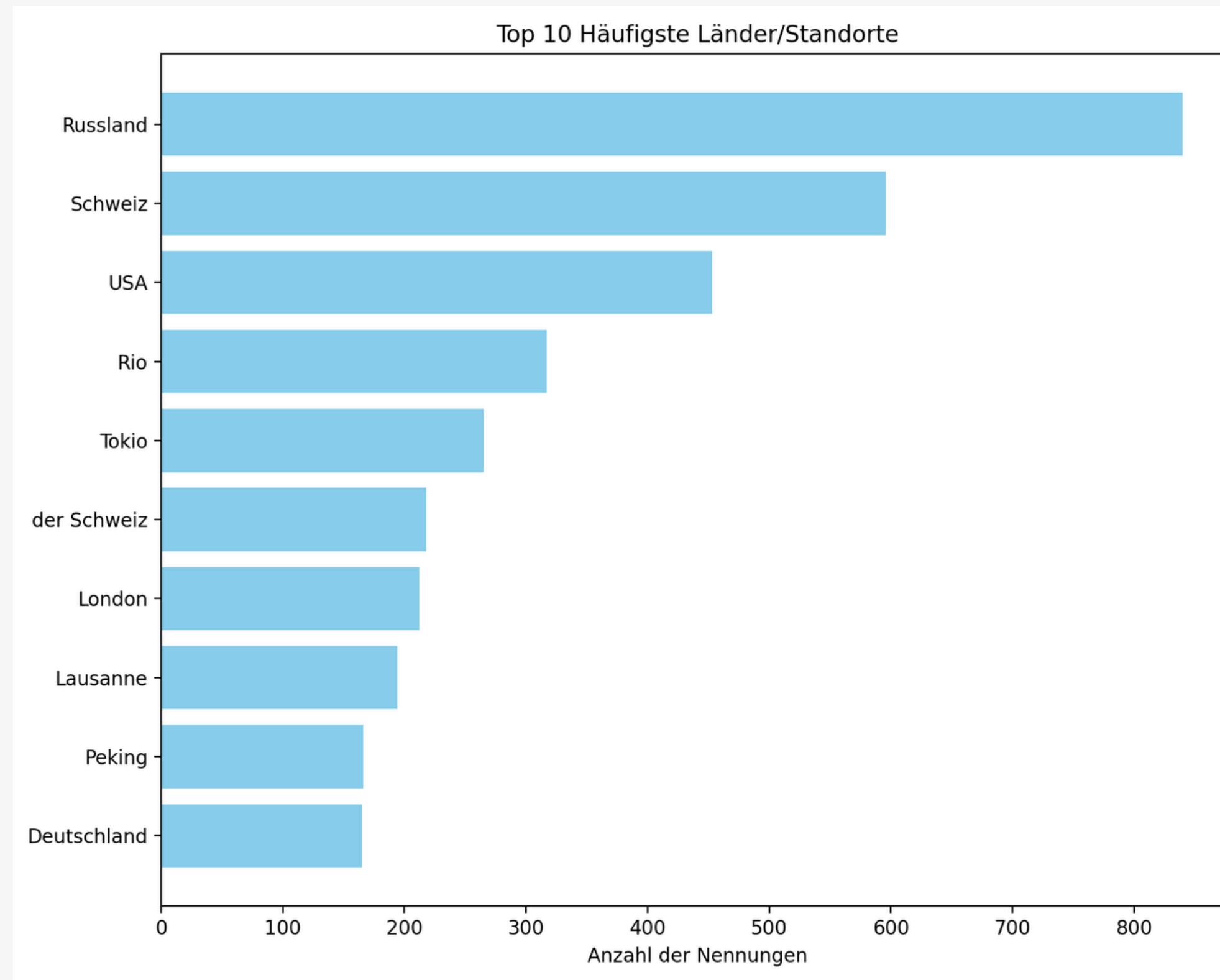
# Extrahiere die häufigsten Organisationen und Personen
top_organizations = organization_counter.most_common(10)
top_persons = person_counter.most_common(10)

import matplotlib.pyplot as plt

# Funktion zum Zeichnen von Balkendiagrammen
def plot_top_entities(entity_list, title):
    entities, counts = zip(*entity_list)
    plt.figure(figsize=(10, 8))
    plt.barh(entities, counts, color='skyblue')
    plt.xlabel('Anzahl der Nennungen')
    plt.title(title)
    plt.gca().invert_yaxis() # Um die Balken von oben nach unten anzuzeigen
    plt.show()

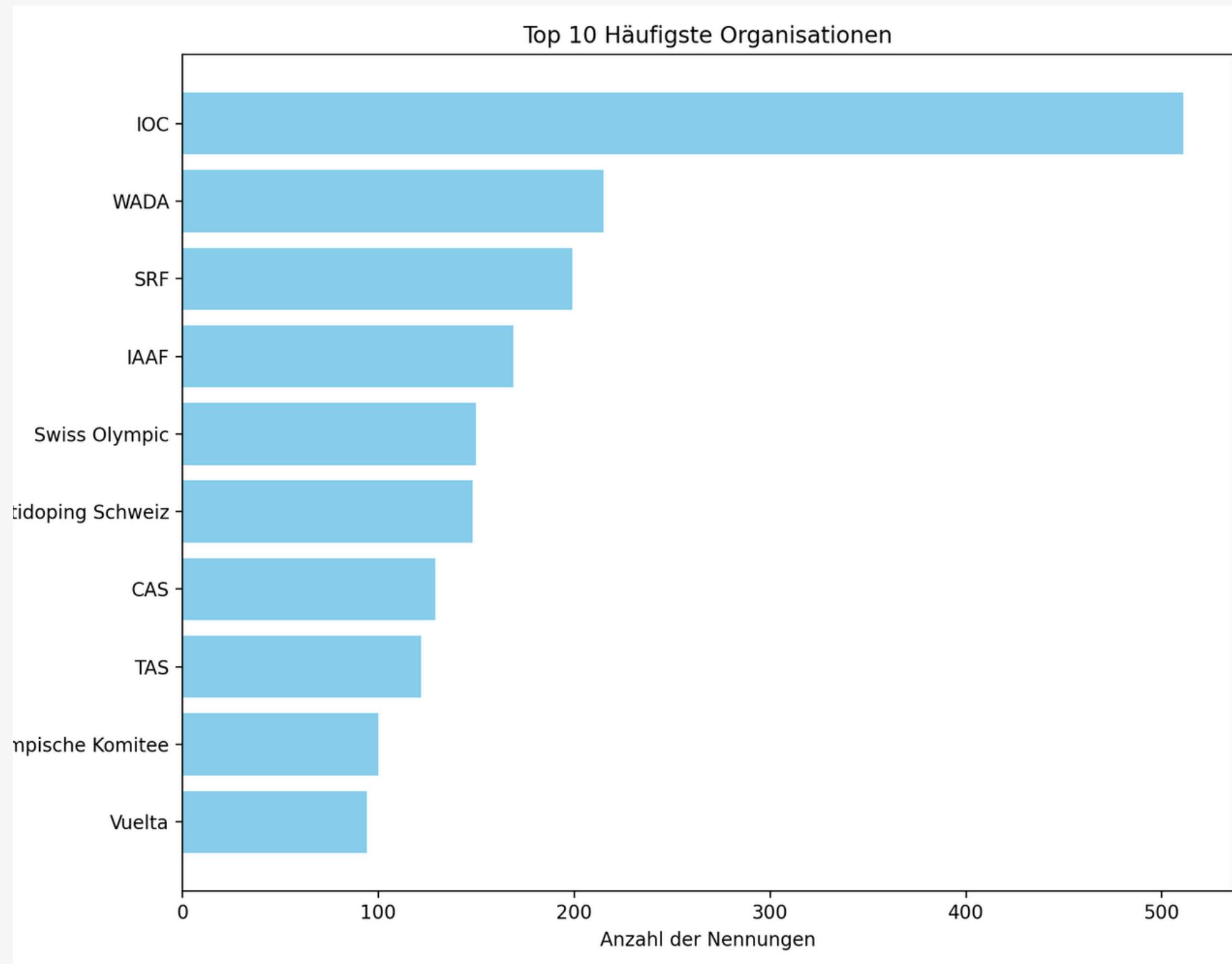
# Zeichne die Diagramme für die häufigsten Entitäten
plot_top_entities(location_counter.most_common(10), 'Top 10 Häufigste Länder/Standorte')
plot_top_entities(top_organizations, 'Top 10 Häufigste Organisationen')
plot_top_entities(top_persons, 'Top 10 Häufigste Personen')
```

RESULTS



Most frequent countries:
Russland (830)
Schweiz (600)
USA (450)

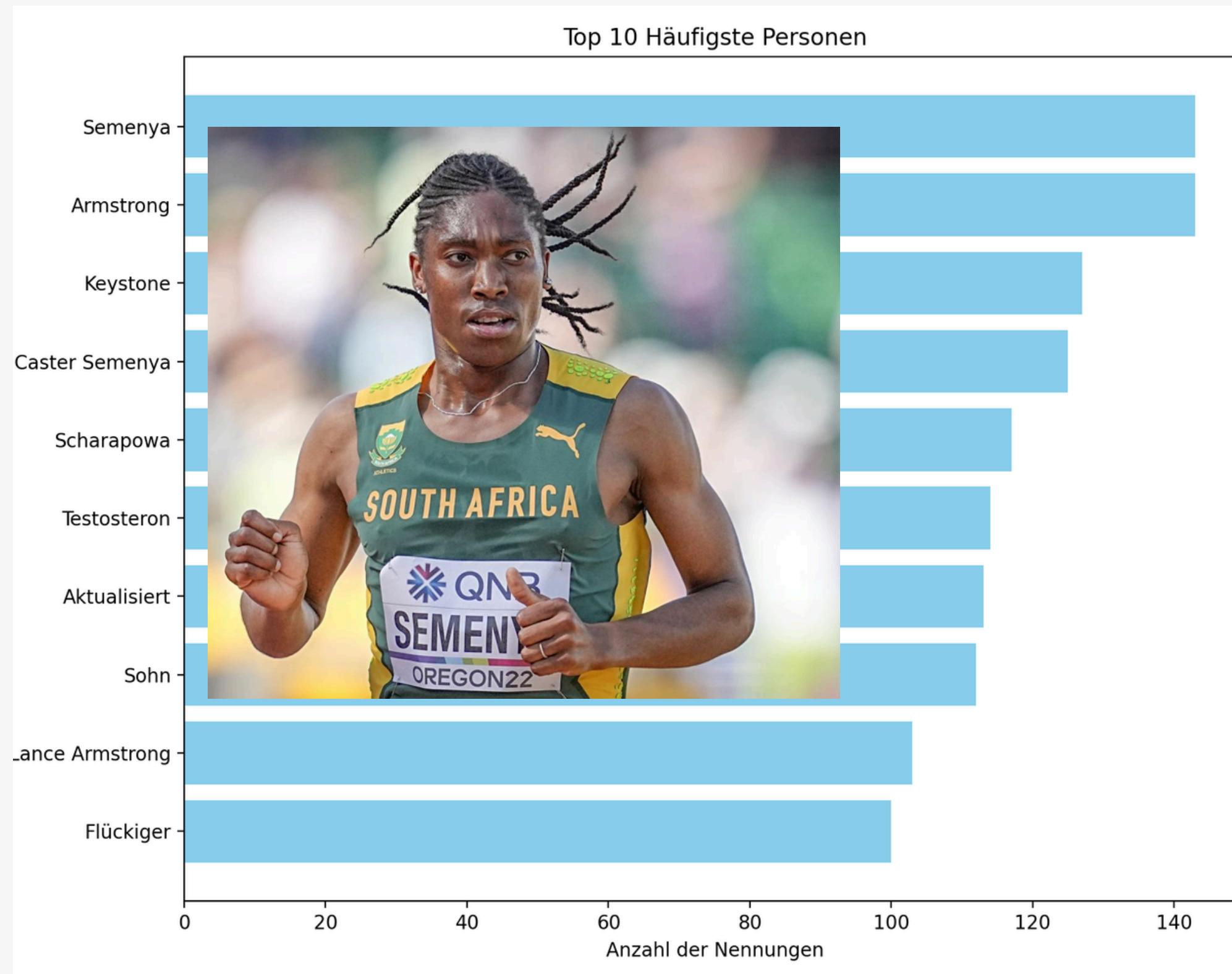
RESULTS



Most frequent organization:
International Olympic Committee IOC (500)

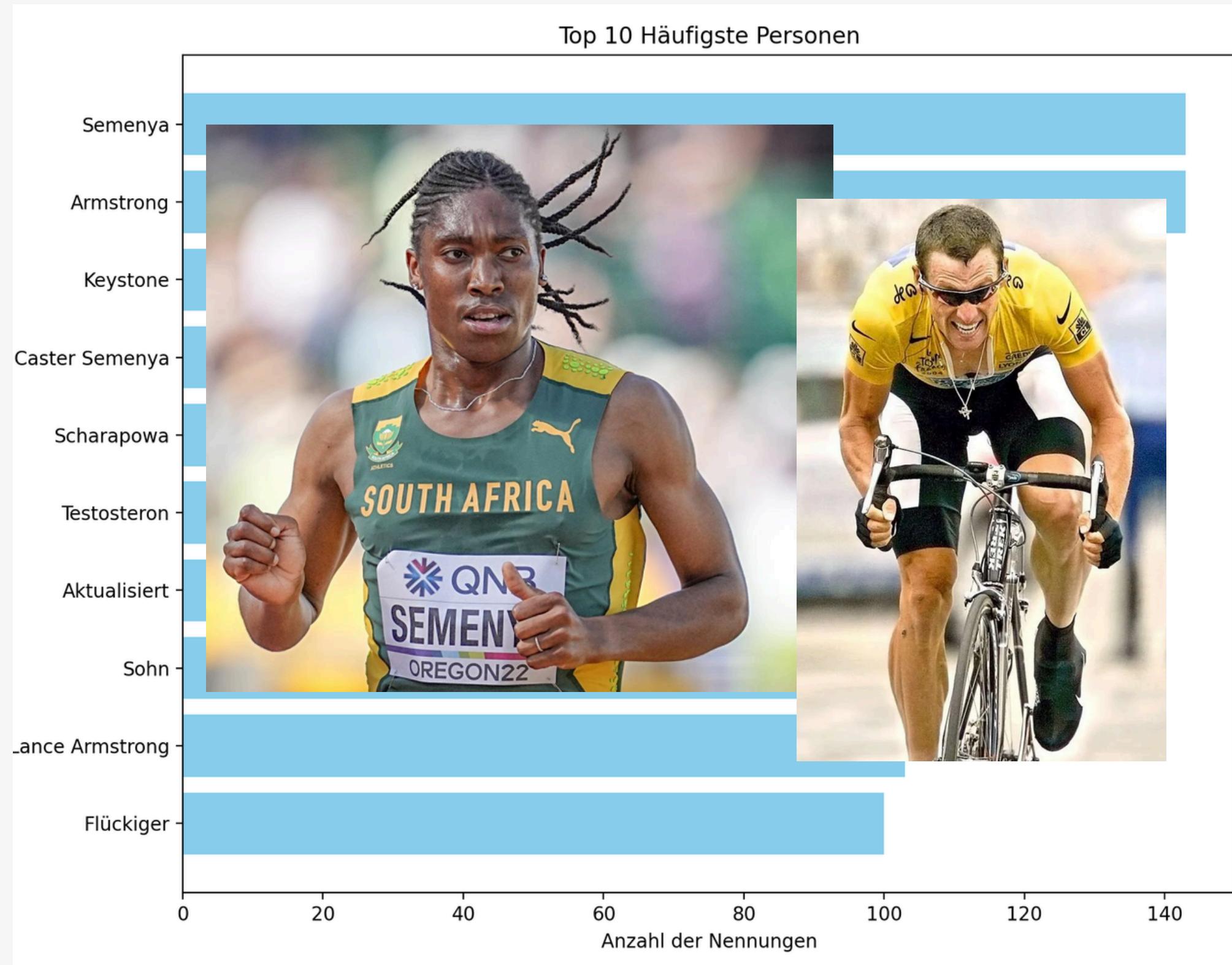


RESULTS



Most frequent individuals:
Caster Semenya (140)
as intersex woman

RESULTS

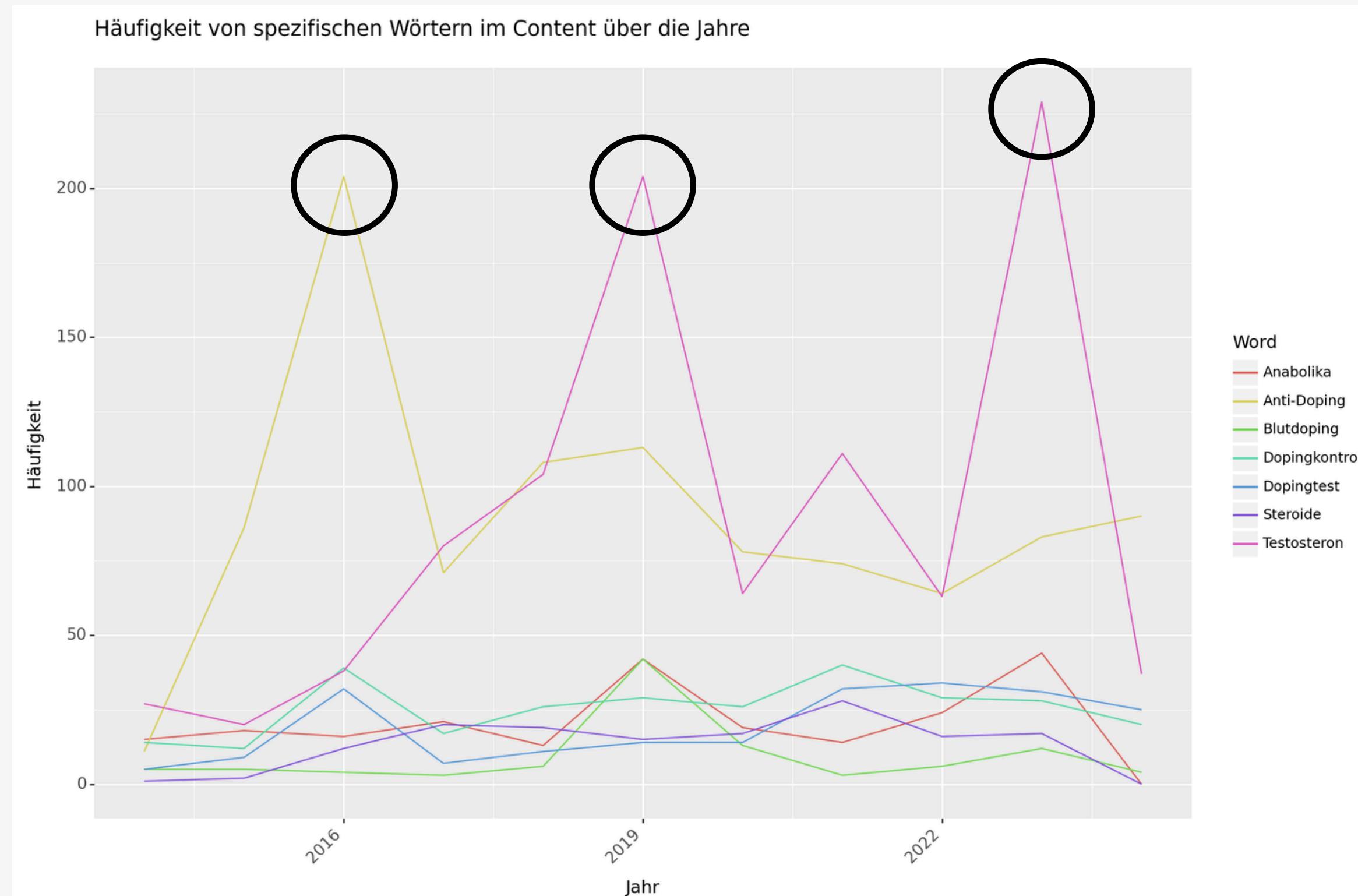


Most frequent individuals:

Caster Semenya (140)
as intersex woman

Lance Armstrong (140)

RESULTS

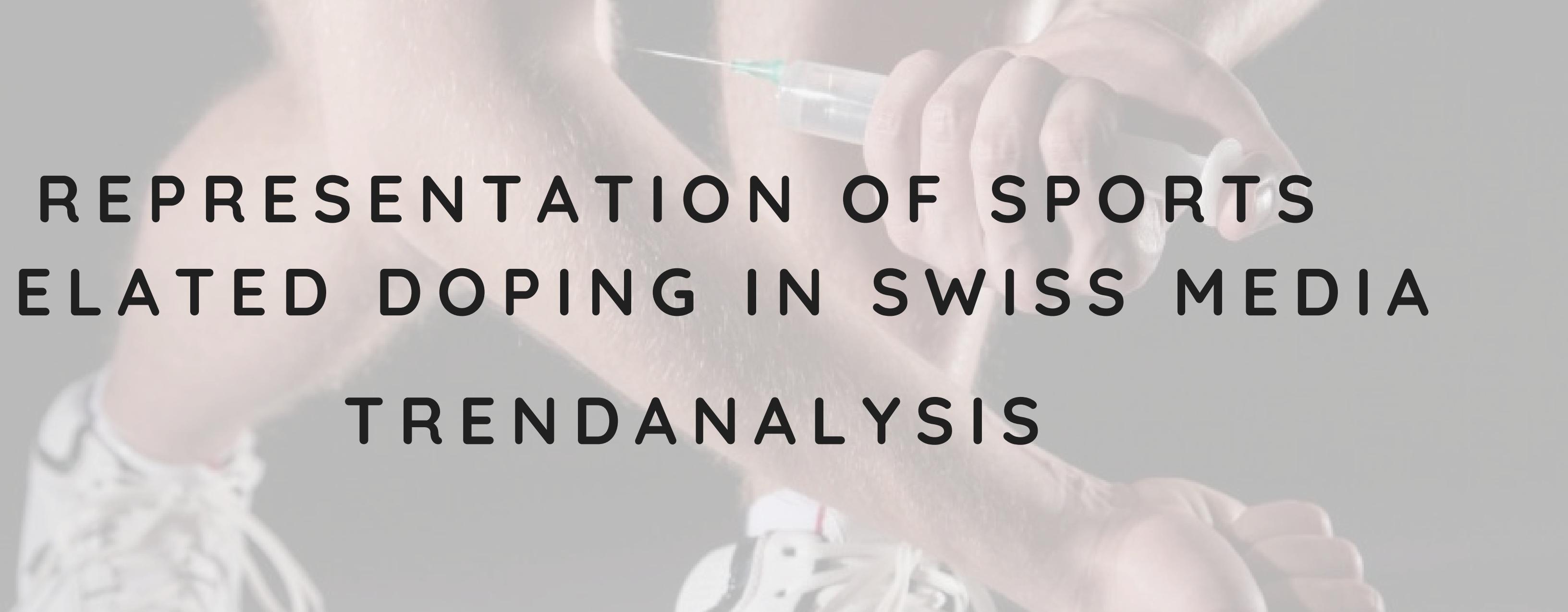


CONCLUSION

- 注射器图标 The reporting of sports-related doping in Swiss media correlates with relevant sport events such as the Olympics or other significant rulings and judgements that impact well-known athletes.
- 注射器图标 The reporting of sports-related doping in Swiss media has remained mostly stable over the years and events.

LEARNINGS

- 💉 Next time we would choose a different time period, because in 2013 Lance Armstrong confessed to his doping abuse or the Winter Olympics took place in Sochi in February 2014.
- 📱 The code for the entities (countries / organizations / people) still needs to be refined.



**REPRESENTATION OF SPORTS
RELATED DOPING IN SWISS MEDIA
TRENDANALYSIS**

Aaron Steiner, Livia Köppel, Silvan Lachmuth