Задание: Выбрать набор данных (датасет). Вы можете найти список свободно распространяемых датасетов здесь. Для лабораторных работ не рекомендуется выбирать датасеты очень большого размера.

- Создать "историю о данных" в виде юпитер-ноутбука, с учетом следующих требований:
- 1. История должна содержать не менее 5 шагов (где 5 - рекомендуемое количество шагов). Каждый шаг содержит график и его текстовую интерпретацию.
- 2. На каждом шаге наряду с удачным итоговым графиком рекомендуется в юпитер-ноутбуке оставлять результаты предварительных "неудачных" графиков.
- 3. Не рекомендуется повторять виды графиков, желательно создать 5 графиков различных видов.
- 4. Выбор графиков должен быть обоснован использованием методологии data-to-viz. Рекомендуется учитывать типичные ошибки построения выбранного вида графика по методологии data-to-viz. Если методология Вами отвергается, то просьба обосновать Ваше решение по выбору графика.
- 5. История должна содержать итоговые выводы. В реальных "историях о данных" именно эти выводы представляют собой основную ценность для предприятия.
- Сформировать отчет и разместить его в своем репозитории на github.

Подгрузка либ и датасета

В качестве датасета, был выбран датасет из kaggle ⟹ **Netflix TV Shows and Movies**

In [1]:
```
import plotly.express as px
from plotly import graph_objects
import pandas as pd
```

In [2]:
```
# Дропнем колонку id, она не нужна для анализа
df = pd.read_csv('titles.csv')
df.drop(columns=['id'], inplace=True)
```

Конвертируем колонку с жанрами в список

Это нужно для построения pie chart с процентным соотношением фильмов по жанрам

In [3]:
```
def to_list(s: str):
    s = s.replace('\'', '')
    s = s.replace('[', '')
    s = s.replace(']', '')
    s = s.replace(', ', ',')
    if s == '':
        return ['None']
    return s.split(',')

df['genres'] = df['genres'].apply(to_list)
```

Подсчитаем количество фильмов в каждом жанре

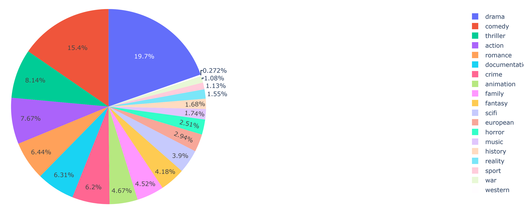
In [4]:
```
unique_genres = set()
for genres in df['genres']:
    unique_genres = unique_genres | set(genres)

unique_genres = list(unique_genres)
genres_dict = dict()
for genre in unique_genres:
    genres_dict.update({genre: 0})

for genres in df['genres']:
    for genre in genres:
        genres_dict[genre] += 1

genres_dict.pop('None')
```

Out[4]:
```
59
```

---

1. Построим Pie Chart

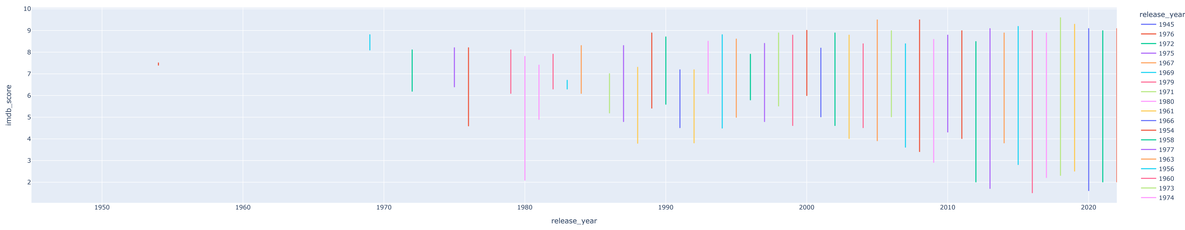
In [5]:
```
data = genres_dict.values()
labels = genres_dict.keys()

#plotly умеет работать с фреймами пандаса, но мне не нет необходимости формировать новый фрейм, я просто передам параметры на прямую
fig = px.pie(None, labels, data)
fig.show()
```



---

2. Построим график, отражающий колебания оценки взависимоости от года релиза

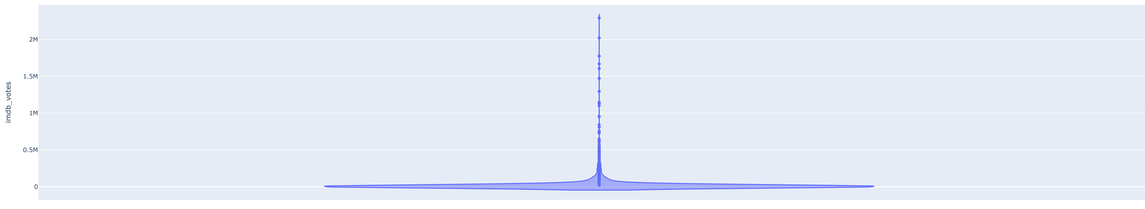
In [6]:
```
fig = px.line(df, x="release_year", y="imdb_score", color="release_year")
fig.update_traces(textposition="bottom right")
fig.show()
```



---

3. Построим график, отражающий распределение максимального значения всех войтов

In [7]:
```
fig = px.violin(df, y="imdb_votes")
fig.show()
```

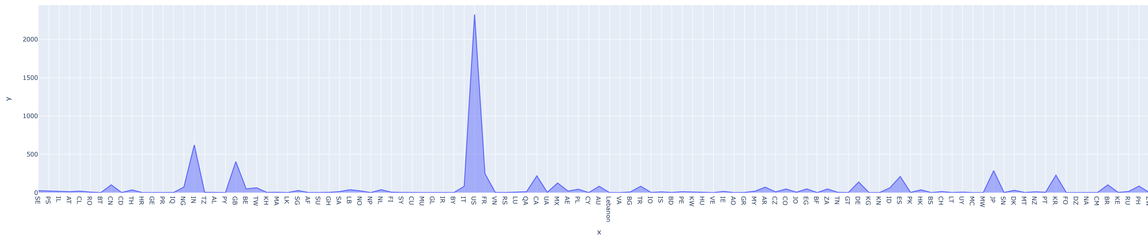


---

4. Построим график, отражающий распределение максимальной оценки, взависимости оот страны прооизводителя

In [8]:
```
# print(df.columns)
# df['age_certification'].unique()
df['production_countries'] = df['production_countries'].apply(to_list)
```

In [9]:
```
unique_countres = set()
for countres in df['production_countries']:
    unique_countres = unique_countres | set(countres)

unique_countres = list(unique_countres)
countres_dict = dict()
for genre in unique_countres:
    countres_dict.update({genre: 0})

for countres in df['production_countries']:
    for contry in countres:
        countres_dict[contry] += 1

countres_dict.pop('None')
len(countres_dict)

fig = px.area(None, x=countres_dict.keys(), y=countres_dict.values())
fig.show()
```



---

5. Построим график, отражающий распределение кол-ва игр от возрастного рейтинга

In [10]:
```
ac = dict()
for key in df['age_certification'].unique():
    ac.update({key: 0})

for key in df['age_certification']:
    ac[key] += 1


fig = px.histogram(x=ac.keys(), y=ac.values())
fig.show()
```