```
import pandas as pd
```

IMDB (https://datasets.imdbws.com/) has data available for use, but it requires some preprocessing. For starters, the name of the movie and the rating aren't in the same dataset. So we will combine those to one pandas dataframe. We can also combine any of the IMDB datasets for further data mining, as they have a shared constant for every row (tconst)

```
imdb_basics = pd.read_csv('data_imdb_basics.tsv', sep='\t')
imdb_basics.shape
```

```
<ipython-input-127-66eece6d45dd>:1: DtypeWarning: Columns (4) have
mixed types. Specify dtype option on import or set low_memory=False.
  imdb_basics = pd.read_csv('data_imdb_basics.tsv', sep='\t')
```

```
(9621894, 9)
```

```
imdb_ratings = pd.read_csv('data_imdb_ratings.tsv', sep='\t')
imdb_ratings.shape
```

```
(1280237, 3)
```

```
# DEPRECATED DATASET WE ORIGINALLY WANTED TO USE
#netflix = pd.read_csv('netflix_titles.csv')
#netflix.shape
```

```
# DEPRECATED DATASET
# NETFLIX:\n{netflix.iloc[:1]}\n\n\n
print(f"IMDB BASICS:\n{imdb_basics.iloc[:5]}\n\n\nIMDB RATINGS:\
n{imdb_ratings.iloc[:5]}")
```

```
IMDB BASICS:
      tconst titleType            primaryTitle           originalTitle
\
0  tt0000001     short              Carmencita              Carmencita

1  tt0000002     short  Le clown et ses chiens  Le clown et ses chiens

2  tt0000003     short          Pauvre Pierrot          Pauvre Pierrot

3  tt0000004     short             Un bon bock             Un bon bock

4  tt0000005     short        Blacksmith Scene        Blacksmith Scene


  isAdult startYear endYear runtimeMinutes                    genres
0       0      1894      \N              1         Documentary,Short
1       0      1892      \N              5           Animation,Short
2       0      1892      \N              4  Animation,Comedy,Romance
3       0      1892      \N             12           Animation,Short
4       0      1893      \N              1              Comedy,Short
```

```
IMDB RATINGS:
      tconst  averageRating  numVotes
0  tt0000001            5.7      1952
1  tt0000002            5.8       264
2  tt0000003            6.5      1786
3  tt0000004            5.6       179
4  tt0000005            6.2      2586
```

```
imdb_combined = imdb_basics.merge(imdb_ratings, left_on='tconst',
right_on='tconst')
print(f"{imdb_combined.shape}")

print(f"\nIMDB COMBINED:\n{imdb_combined.iloc[:5]}")
```

```
(1280236, 11)

IMDB COMBINED:
      tconst titleType            primaryTitle           originalTitle
\
0  tt0000001     short              Carmencita              Carmencita

1  tt0000002     short  Le clown et ses chiens  Le clown et ses chiens

2  tt0000003     short          Pauvre Pierrot          Pauvre Pierrot

3  tt0000004     short             Un bon bock             Un bon bock

4  tt0000005     short        Blacksmith Scene        Blacksmith Scene


  isAdult startYear endYear runtimeMinutes
genres  \
0       0      1894      \N              1         Documentary,Short

1       0      1892      \N              5           Animation,Short

2       0      1892      \N              4  Animation,Comedy,Romance

3       0      1892      \N             12           Animation,Short

4       0      1893      \N              1              Comedy,Short


   averageRating  numVotes
0            5.7      1952
1            5.8       264
2            6.5      1786
```

```
3            5.6       179
4            6.2      2586
```

Data is now in two sets, Netflix information, and IMDB information. Lets create one data set that contains Netflix, and IMDB movies together.

```
# DEPRECATED DATASET
#netflix_imdb_combined = pd.merge(netflix,imdb_combined,
suffixes=['_netflix','_imdb'], left_on='title',
right_on='originalTitle')
#print(f"{netflix_imdb_combined.shape}\n\nNETFLIX IMDB COMBINED:\
n{netflix_imdb_combined.iloc[:1]}")
```

Now we have a combined dataframe of the shows available on netflix, and their imdb information. Lets filter out the TV shows

```
# DEPRECATED DATASET
#netflix_imdb_combined_no_tv =
netflix_imdb_combined[(netflix_imdb_combined['type'] == 'Movie')]
#print(f"{netflix_imdb_combined_no_tv.shape}\n\nCOMBINED NO TV:\
n{netflix_imdb_combined_no_tv.iloc[:1]}")
```

Lets also filter out an NaN rows

```
imdb_combined = imdb_combined.dropna()
```

Lets start by creating our test/training data split (85/15)

```
from sklearn.model_selection import train_test_split

train, test = train_test_split(imdb_combined, test_size=0.15)

print(f"TRAINING:\t{train.shape}\n{train.iloc[:1]}\n\n\nTESTING:\
t{test.shape}\n{test.iloc[:1]}")
```

```
TRAINING:  (1088198, 11)
            tconst  titleType                     primaryTitle  \
648593  tt14139080  tvEpisode  Everybody Loves the Carringtons

                         originalTitle isAdult startYear endYear  \
648593  Everybody Loves the Carringtons       0      2021      \N

       runtimeMinutes genres  averageRating  numVotes
648593             42  Drama            7.0       367


TESTING:   (192036, 11)
           tconst titleType         primaryTitle        originalTitle
isAdult  \
121124  tt0181523     movie  Ena Koritsi Gia Dyo  Ena koritsi gia dyo
```

```
       startYear endYear runtimeMinutes          genres  averageRating
\
121124      1963      \N             90  Comedy,Romance            6.3


        numVotes
121124       368
```

```
# Generate list of all genres
genre_list = []

genres = train['genres'].unique()
for genresgroup in genres:
  if genresgroup != genresgroup:
    print(genresgroup)
  genre_sep = genresgroup.split(',')
  genre_list = genre_list + genre_sep

# Find unique genres from our data.
list_set = set(genre_list)
unique_list_genre = (list(list_set))
unique_list_genre.remove("\\N")
for genre in unique_list_genre:
  print(genre)
```

```
Crime
Comedy
History
Mystery
Talk-Show
Music
Fantasy
Sport
Biography
Musical
News
Sci-Fi
Action
Thriller
Reality-TV
Family
Animation
Drama
Romance
Western
Short
Film-Noir
War
```

```
Adult
Documentary
Horror
Game-Show
Adventure
```

```
# Create dictionary for all genres
genre_split = {}
# Add each genre as a key, and its dictionary as the value
for genre in unique_list_genre:
  genre_split[genre] =
train.loc[(train['genres'].str.contains(genre))]

print(genre_split['Horror'].iloc[:1])
```

```
          tconst titleType primaryTitle originalTitle isAdult
startYear  \
959068  tt3443808     video     Chubbies      Chubbies       0
2014

       endYear runtimeMinutes                genres  averageRating
numVotes
959068      \N             89  Comedy,Horror,Sci-Fi            4.7
91
```

Now all of our data is split by genre as well.

```
genre_split_avg = {}
# for each genre
for genre in unique_list_genre:
  #initalize values
  avg = 0
  count = 0
  #declare working genre
  print(f"GENRE: {genre}:")
  #iterate over dataframe to find the average rating, and number of
movies
  for index, row in genre_split[genre].iterrows():
    #print(row['primaryTitle'], row['averageRating'])
    avg = avg + row['averageRating']
    count = count + 1
  avg = avg/count
  print(f"\nAverage: {avg}, Number: {count}\n\n")
  #split dataframe above and below average
  genre_split_avg['B'+genre] = genre_split[genre][genre_split[genre]
['averageRating'] <= avg]
  genre_split_avg['A'+genre] = genre_split[genre][genre_split[genre]
['averageRating'] > avg]
```

```
GENRE: Crime:
```

Average: 7.141819996043043, Number: 116253

GENRE: Comedy:

Average: 6.988683059152979, Number: 342840

GENRE: History:

Average: 7.346030372933504, Number: 31212

GENRE: Mystery:

Average: 7.132382305579212, Number: 52785

GENRE: Talk-Show:

Average: 6.873209364104323, Number: 30115

GENRE: Music:

Average: 6.96755002939705, Number: 45923

GENRE: Fantasy:

Average: 7.065640483583272, Number: 45411

GENRE: Sport:

Average: 7.118973884908732, Number: 21635

GENRE: Biography:

Average: 7.2209120550270836, Number: 22389

GENRE: Musical:

Average: 6.627235810746995, Number: 10589

GENRE: News:

Average: 6.693392980041285, Number: 14530

GENRE: Sci-Fi:

Average: 6.725301953205907, Number: 28978

GENRE: Action:

Average: 7.018534937412822, Number: 135571

GENRE: Thriller:

Average: 6.406603268148968, Number: 44796

GENRE: Reality-TV:

Average: 6.961207515452866, Number: 57109

GENRE: Family:

Average: 7.072674595913373, Number: 78696

GENRE: Animation:

Average: 7.099453413088723, Number: 129897

GENRE: Drama:

Average: 7.054393452064071, Number: 368116

GENRE: Romance:

Average: 6.8746902292904535, Number: 83126

GENRE: Western:

```
Average: 6.956647010524729, Number: 13397


GENRE: Short:

Average: 6.841250328383142, Number: 129422


GENRE: Film-Noir:

Average: 6.461405835543761, Number: 754


GENRE: War:

Average: 7.046074884307963, Number: 11885


GENRE: Adult:

Average: 6.333515848390834, Number: 16437


GENRE: Documentary:

Average: 7.270386523509826, Number: 135050


GENRE: Horror:

Average: 6.123888107585453, Number: 45508


GENRE: Game-Show:

Average: 7.046774860062344, Number: 26619


GENRE: Adventure:

Average: 7.093430364868467, Number: 119961


print(f"{genre_split_avg['AAction'].iloc[:1]}\n\n\
n{genre_split_avg['BAction'].iloc[:1]}")
```

```
          tconst  titleType                      primaryTitle  \
635056  tt1383574  tvEpisode  Haruka vs Shu! The Final Battle!!

                           originalTitle isAdult startYear endYear  \
635056  Haruka vs Shu! The Final Battle!!       0      2006      \N

       runtimeMinutes                      genres  averageRating
numVotes
635056             22  Action,Adventure,Animation            7.5
100



          tconst  titleType            primaryTitle
originalTitle  \
586424  tt1281261  tvEpisode  Flash Back/The Warrior  Flash Back/The
Warrior

       isAdult startYear endYear runtimeMinutes
genres  \
586424       0      1982      \N             24
Action,Adventure,Animation

        averageRating  numVotes
586424            5.8        18
```

lets plot rating against time. where the x axis is time (year) and y axis is rating

```
# rating_data = { 'genre': [title,year,rating],[title,year,rating],
'genre': [title,year,rating] }
rating_data = {}

for genre in genre_split:
  title_year_rating = []
  #print(genre, '->', genre_split[genre].iloc[:1])
  for index, row in genre_split[genre].iterrows():

    title = row['primaryTitle']
    year = row['startYear']
    rating = row['averageRating']
    tyr = [title,year,rating]
    title_year_rating.append(tyr)

  rating_data[genre] = title_year_rating


def best_fit(X, Y):

    xbar = sum(X)/len(X)
    ybar = sum(Y)/len(Y)
    n = len(X) # or len(Y)
```

```
    numer = sum([xi*yi for xi,yi in zip(X, Y)]) - n * xbar * ybar
    denum = sum([xi**2 for xi in X]) - n * xbar**2

    b = numer / denum
    a = ybar - b * xbar

    print('best fit line:\ny = {:.2f} + {:.2f}x'.format(a, b))

    return a, b

import matplotlib.pyplot as plt

for genre in genre_split:
  year = []
  rating = []

  for index, row in genre_split[genre].iterrows():
    if row['startYear'] != '\\N':

      year.append(int(row['startYear']))
      rating.append(float(row['averageRating']))
      if len(year) <= 10:
        print(year, rating)

  plt.scatter(year, rating, s=.1, alpha=.75)

  a, b = best_fit(year, rating)

  yfit = [a + b * yeari for yeari in year]
  plt.plot(year, yfit)
```

```
[1995] [5.4]
[1995, 2003] [5.4, 8.0]
[1995, 2003, 2018] [5.4, 8.0, 7.3]
[1995, 2003, 2018, 2014] [5.4, 8.0, 7.3, 8.1]
[1995, 2003, 2018, 2014, 2014] [5.4, 8.0, 7.3, 8.1, 9.0]
[1995, 2003, 2018, 2014, 2014, 1959] [5.4, 8.0, 7.3, 8.1, 9.0, 6.4]
[1995, 2003, 2018, 2014, 2014, 1959, 1996] [5.4, 8.0, 7.3, 8.1, 9.0,
6.4, 7.7]
[1995, 2003, 2018, 2014, 2014, 1959, 1996, 2020] [5.4, 8.0, 7.3, 8.1,
9.0, 6.4, 7.7, 6.3]
[1995, 2003, 2018, 2014, 2014, 1959, 1996, 2020, 1960] [5.4, 8.0, 7.3,
8.1, 9.0, 6.4, 7.7, 6.3, 8.4]
[1995, 2003, 2018, 2014, 2014, 1959, 1996, 2020, 1960, 2010] [5.4,
8.0, 7.3, 8.1, 9.0, 6.4, 7.7, 6.3, 8.4, 7.6]
best fit line:
y = -13.67 + 0.01x
[1998] [4.2]
[1998, 2014] [4.2, 7.2]
```

[1998, 2014, 2007] [4.2, 7.2, 6.6]
[1998, 2014, 2007, 1995] [4.2, 7.2, 6.6, 6.0]
[1998, 2014, 2007, 1995, 2014] [4.2, 7.2, 6.6, 6.0, 4.7]
[1998, 2014, 2007, 1995, 2014, 2008] [4.2, 7.2, 6.6, 6.0, 4.7, 6.5]
[1998, 2014, 2007, 1995, 2014, 2008, 1994] [4.2, 7.2, 6.6, 6.0, 4.7, 6.5, 8.7]
[1998, 2014, 2007, 1995, 2014, 2008, 1994, 2021] [4.2, 7.2, 6.6, 6.0, 4.7, 6.5, 8.7, 6.7]
[1998, 2014, 2007, 1995, 2014, 2008, 1994, 2021, 1986] [4.2, 7.2, 6.6, 6.0, 4.7, 6.5, 8.7, 6.7, 6.8]
[1998, 2014, 2007, 1995, 2014, 2008, 1994, 2021, 1986, 2014] [4.2, 7.2, 6.6, 6.0, 4.7, 6.5, 8.7, 6.7, 6.8, 6.0]
best fit line:
y = -11.80 + 0.01x
[2013] [6.0]
[2013, 1994] [6.0, 5.2]
[2013, 1994, 2021] [6.0, 5.2, 6.4]
[2013, 1994, 2021, 2020] [6.0, 5.2, 6.4, 7.6]
[2013, 1994, 2021, 2020, 2007] [6.0, 5.2, 6.4, 7.6, 7.9]
[2013, 1994, 2021, 2020, 2007, 2006] [6.0, 5.2, 6.4, 7.6, 7.9, 6.9]
[2013, 1994, 2021, 2020, 2007, 2006, 1994] [6.0, 5.2, 6.4, 7.6, 7.9, 6.9, 7.1]
[2013, 1994, 2021, 2020, 2007, 2006, 1994, 2020] [6.0, 5.2, 6.4, 7.6, 7.9, 6.9, 7.1, 7.7]
[2013, 1994, 2021, 2020, 2007, 2006, 1994, 2020, 2012] [6.0, 5.2, 6.4, 7.6, 7.9, 6.9, 7.1, 7.7, 8.1]
[2013, 1994, 2021, 2020, 2007, 2006, 1994, 2020, 2012, 1957] [6.0, 5.2, 6.4, 7.6, 7.9, 6.9, 7.1, 7.7, 8.1, 6.5]
best fit line:
y = -14.26 + 0.01x
[2003] [8.0]
[2003, 2014] [8.0, 8.1]
[2003, 2014, 1996] [8.0, 8.1, 7.7]
[2003, 2014, 1996, 2010] [8.0, 8.1, 7.7, 7.6]
[2003, 2014, 1996, 2010, 2021] [8.0, 8.1, 7.7, 7.6, 5.2]
[2003, 2014, 1996, 2010, 2021, 1993] [8.0, 8.1, 7.7, 7.6, 5.2, 9.2]
[2003, 2014, 1996, 2010, 2021, 1993, 2000] [8.0, 8.1, 7.7, 7.6, 5.2, 9.2, 6.5]
[2003, 2014, 1996, 2010, 2021, 1993, 2000, 2010] [8.0, 8.1, 7.7, 7.6, 5.2, 9.2, 6.5, 6.7]
[2003, 2014, 1996, 2010, 2021, 1993, 2000, 2010, 2015] [8.0, 8.1, 7.7, 7.6, 5.2, 9.2, 6.5, 6.7, 7.2]
[2003, 2014, 1996, 2010, 2021, 1993, 2000, 2010, 2015, 2011] [8.0, 8.1, 7.7, 7.6, 5.2, 9.2, 6.5, 6.7, 7.2, 7.3]
best fit line:
y = -8.47 + 0.01x
[1998] [4.2]
[1998, 2017] [4.2, 6.9]
[1998, 2017, 1967] [4.2, 6.9, 6.0]
[1998, 2017, 1967, 2016] [4.2, 6.9, 6.0, 7.0]

[1998, 2017, 1967, 2016, 2019] [4.2, 6.9, 6.0, 7.0, 9.2]
[1998, 2017, 1967, 2016, 2019, 2010] [4.2, 6.9, 6.0, 7.0, 9.2, 6.5]
[1998, 2017, 1967, 2016, 2019, 2010, 2014] [4.2, 6.9, 6.0, 7.0, 9.2, 6.5, 6.2]
[1998, 2017, 1967, 2016, 2019, 2010, 2014, 1998] [4.2, 6.9, 6.0, 7.0, 9.2, 6.5, 6.2, 4.3]
[1998, 2017, 1967, 2016, 2019, 2010, 2014, 1998, 2010] [4.2, 6.9, 6.0, 7.0, 9.2, 6.5, 6.2, 4.3, 7.4]
[1998, 2017, 1967, 2016, 2019, 2010, 2014, 1998, 2010, 2015] [4.2, 6.9, 6.0, 7.0, 9.2, 6.5, 6.2, 4.3, 7.4, 6.2]
best fit line:
y = -15.88 + 0.01x
[2005] [8.9]
[2005, 2018] [8.9, 6.6]
[2005, 2018, 2017] [8.9, 6.6, 6.9]
[2005, 2018, 2017, 1967] [8.9, 6.6, 6.9, 6.0]
[2005, 2018, 2017, 1967, 2013] [8.9, 6.6, 6.9, 6.0, 8.6]
[2005, 2018, 2017, 1967, 2013, 2021] [8.9, 6.6, 6.9, 6.0, 8.6, 9.1]
[2005, 2018, 2017, 1967, 2013, 2021, 2010] [8.9, 6.6, 6.9, 6.0, 8.6, 9.1, 5.0]
[2005, 2018, 2017, 1967, 2013, 2021, 2010, 2012] [8.9, 6.6, 6.9, 6.0, 8.6, 9.1, 5.0, 6.6]
[2005, 2018, 2017, 1967, 2013, 2021, 2010, 2012, 2008] [8.9, 6.6, 6.9, 6.0, 8.6, 9.1, 5.0, 6.6, 6.6]
[2005, 2018, 2017, 1967, 2013, 2021, 2010, 2012, 2008, 1974] [8.9, 6.6, 6.9, 6.0, 8.6, 9.1, 5.0, 6.6, 6.6, 8.0]
best fit line:
y = -17.17 + 0.01x
[2000] [7.3]
[2000, 2012] [7.3, 8.4]
[2000, 2012, 2022] [7.3, 8.4, 5.2]
[2000, 2012, 2022, 1989] [7.3, 8.4, 5.2, 7.2]
[2000, 2012, 2022, 1989, 2002] [7.3, 8.4, 5.2, 7.2, 6.9]
[2000, 2012, 2022, 1989, 2002, 1984] [7.3, 8.4, 5.2, 7.2, 6.9, 5.4]
[2000, 2012, 2022, 1989, 2002, 1984, 1996] [7.3, 8.4, 5.2, 7.2, 6.9, 5.4, 7.8]
[2000, 2012, 2022, 1989, 2002, 1984, 1996, 2018] [7.3, 8.4, 5.2, 7.2, 6.9, 5.4, 7.8, 7.2]
[2000, 2012, 2022, 1989, 2002, 1984, 1996, 2018, 2022] [7.3, 8.4, 5.2, 7.2, 6.9, 5.4, 7.8, 7.2, 7.5]
[2000, 2012, 2022, 1989, 2002, 1984, 1996, 2018, 2022, 2017] [7.3, 8.4, 5.2, 7.2, 6.9, 5.4, 7.8, 7.2, 7.5, 8.7]
best fit line:
y = -16.04 + 0.01x
[2021] [6.9]
[2021, 1993] [6.9, 7.1]
[2021, 1993, 2006] [6.9, 7.1, 7.1]
[2021, 1993, 2006, 2019] [6.9, 7.1, 7.1, 6.7]
[2021, 1993, 2006, 2019, 2021] [6.9, 7.1, 7.1, 6.7, 6.0]
[2021, 1993, 2006, 2019, 2021, 2009] [6.9, 7.1, 7.1, 6.7, 6.0, 5.8]

[2021, 1993, 2006, 2019, 2021, 2009, 2021] [6.9, 7.1, 7.1, 6.7, 6.0, 5.8, 5.4]
[2021, 1993, 2006, 2019, 2021, 2009, 2021, 2021] [6.9, 7.1, 7.1, 6.7, 6.0, 5.8, 5.4, 7.7]
[2021, 1993, 2006, 2019, 2021, 2009, 2021, 2021, 2016] [6.9, 7.1, 7.1, 6.7, 6.0, 5.8, 5.4, 7.7, 5.6]
[2021, 1993, 2006, 2019, 2021, 2009, 2021, 2021, 2016, 2021] [6.9, 7.1, 7.1, 6.7, 6.0, 5.8, 5.4, 7.7, 5.6, 7.7]
best fit line:
y = -23.33 + 0.02x
[1986] [6.9]
[1986, 1994] [6.9, 5.2]
[1986, 1994, 2021] [6.9, 5.2, 6.4]
[1986, 1994, 2021, 2004] [6.9, 5.2, 6.4, 7.5]
[1986, 1994, 2021, 2004, 2019] [6.9, 5.2, 6.4, 7.5, 7.9]
[1986, 1994, 2021, 2004, 2019, 1994] [6.9, 5.2, 6.4, 7.5, 7.9, 7.1]
[1986, 1994, 2021, 2004, 2019, 1994, 1989] [6.9, 5.2, 6.4, 7.5, 7.9, 7.1, 6.5]
[1986, 1994, 2021, 2004, 2019, 1994, 1989, 1999] [6.9, 5.2, 6.4, 7.5, 7.9, 7.1, 6.5, 7.4]
[1986, 1994, 2021, 2004, 2019, 1994, 1989, 1999, 1970] [6.9, 5.2, 6.4, 7.5, 7.9, 7.1, 6.5, 7.4, 7.2]
[1986, 1994, 2021, 2004, 2019, 1994, 1989, 1999, 1970, 2021] [6.9, 5.2, 6.4, 7.5, 7.9, 7.1, 6.5, 7.4, 7.2, 7.1]
best fit line:
y = -15.07 + 0.01x
[2008] [6.6]
[2008, 2004] [6.6, 8.3]
[2008, 2004, 2019] [6.6, 8.3, 6.3]
[2008, 2004, 2019, 2016] [6.6, 8.3, 6.3, 6.8]
[2008, 2004, 2019, 2016, 1969] [6.6, 8.3, 6.3, 6.8, 8.5]
[2008, 2004, 2019, 2016, 1969, 1935] [6.6, 8.3, 6.3, 6.8, 8.5, 5.8]
[2008, 2004, 2019, 2016, 1969, 1935, 2022] [6.6, 8.3, 6.3, 6.8, 8.5, 5.8, 9.1]
[2008, 2004, 2019, 2016, 1969, 1935, 2022, 1980] [6.6, 8.3, 6.3, 6.8, 8.5, 5.8, 9.1, 7.1]
[2008, 2004, 2019, 2016, 1969, 1935, 2022, 1980, 2004] [6.6, 8.3, 6.3, 6.8, 8.5, 5.8, 9.1, 7.1, 5.1]
[2008, 2004, 2019, 2016, 1969, 1935, 2022, 1980, 2004, 1985] [6.6, 8.3, 6.3, 6.8, 8.5, 5.8, 9.1, 7.1, 5.1, 6.3]
best fit line:
y = -19.32 + 0.01x
[1998] [4.2]
[1998, 2016] [4.2, 7.0]
[1998, 2016, 2021] [4.2, 7.0, 8.7]
[1998, 2016, 2021, 2016] [4.2, 7.0, 8.7, 7.6]
[1998, 2016, 2021, 2016, 1998] [4.2, 7.0, 8.7, 7.6, 4.3]
[1998, 2016, 2021, 2016, 1998, 2010] [4.2, 7.0, 8.7, 7.6, 4.3, 7.4]
[1998, 2016, 2021, 2016, 1998, 2010, 2013] [4.2, 7.0, 8.7, 7.6, 4.3, 7.4, 2.4]

[1998, 2016, 2021, 2016, 1998, 2010, 2013, 2019] [4.2, 7.0, 8.7, 7.6, 4.3, 7.4, 2.4, 7.8]
[1998, 2016, 2021, 2016, 1998, 2010, 2013, 2019, 2020] [4.2, 7.0, 8.7, 7.6, 4.3, 7.4, 2.4, 7.8, 2.8]
[1998, 2016, 2021, 2016, 1998, 2010, 2013, 2019, 2020, 2018] [4.2, 7.0, 8.7, 7.6, 4.3, 7.4, 2.4, 7.8, 2.8, 8.5]
best fit line:
y = -43.24 + 0.02x
[2014] [4.7]
[2014, 2010] [4.7, 6.5]
[2014, 2010, 2010] [4.7, 6.5, 8.4]
[2014, 2010, 2010, 1981] [4.7, 6.5, 8.4, 5.2]
[2014, 2010, 2010, 1981, 1964] [4.7, 6.5, 8.4, 5.2, 7.2]
[2014, 2010, 2010, 1981, 1964, 2007] [4.7, 6.5, 8.4, 5.2, 7.2, 8.2]
[2014, 2010, 2010, 1981, 1964, 2007, 1978] [4.7, 6.5, 8.4, 5.2, 7.2, 8.2, 6.1]
[2014, 2010, 2010, 1981, 1964, 2007, 1978, 1983] [4.7, 6.5, 8.4, 5.2, 7.2, 8.2, 6.1, 3.8]
[2014, 2010, 2010, 1981, 1964, 2007, 1978, 1983, 2017] [4.7, 6.5, 8.4, 5.2, 7.2, 8.2, 6.1, 3.8, 8.6]
[2014, 2010, 2010, 1981, 1964, 2007, 1978, 1983, 2017, 2011] [4.7, 6.5, 8.4, 5.2, 7.2, 8.2, 6.1, 3.8, 8.6, 7.8]
best fit line:
y = -14.25 + 0.01x
[2006] [7.5]
[2006, 2021] [7.5, 8.9]
[2006, 2021, 2014] [7.5, 8.9, 9.0]
[2006, 2021, 2014, 1982] [7.5, 8.9, 9.0, 5.8]
[2006, 2021, 2014, 1982, 2016] [7.5, 8.9, 9.0, 5.8, 8.0]
[2006, 2021, 2014, 1982, 2016, 2009] [7.5, 8.9, 9.0, 5.8, 8.0, 2.4]
[2006, 2021, 2014, 1982, 2016, 2009, 2015] [7.5, 8.9, 9.0, 5.8, 8.0, 2.4, 3.1]
[2006, 2021, 2014, 1982, 2016, 2009, 2015, 1964] [7.5, 8.9, 9.0, 5.8, 8.0, 2.4, 3.1, 6.4]
[2006, 2021, 2014, 1982, 2016, 2009, 2015, 1964, 2017] [7.5, 8.9, 9.0, 5.8, 8.0, 2.4, 3.1, 6.4, 7.5]
[2006, 2021, 2014, 1982, 2016, 2009, 2015, 1964, 2017, 1991] [7.5, 8.9, 9.0, 5.8, 8.0, 2.4, 3.1, 6.4, 7.5, 6.8]
best fit line:
y = -22.31 + 0.01x
[1995] [5.4]
[1995, 2003] [5.4, 3.8]
[1995, 2003, 2007] [5.4, 3.8, 5.5]
[1995, 2003, 2007, 2016] [5.4, 3.8, 5.5, 6.6]
[1995, 2003, 2007, 2016, 2021] [5.4, 3.8, 5.5, 6.6, 5.2]
[1995, 2003, 2007, 2016, 2021, 1975] [5.4, 3.8, 5.5, 6.6, 5.2, 6.4]
[1995, 2003, 2007, 2016, 2021, 1975, 2010] [5.4, 3.8, 5.5, 6.6, 5.2, 6.4, 6.1]
[1995, 2003, 2007, 2016, 2021, 1975, 2010, 2011] [5.4, 3.8, 5.5, 6.6, 5.2, 6.4, 6.1, 7.3]

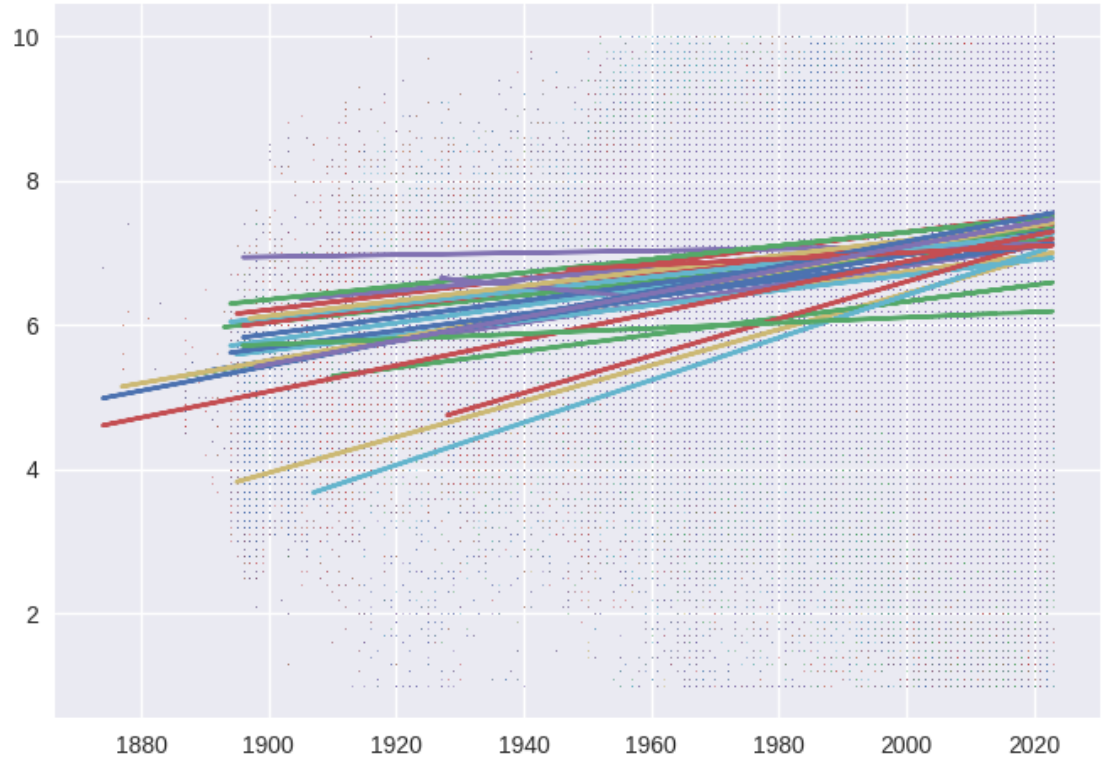
[1995, 2003, 2007, 2016, 2021, 1975, 2010, 2011, 1996] [5.4, 3.8, 5.5, 6.6, 5.2, 6.4, 6.1, 7.3, 6.0]
[1995, 2003, 2007, 2016, 2021, 1975, 2010, 2011, 1996, 2017] [5.4, 3.8, 5.5, 6.6, 5.2, 6.4, 6.1, 7.3, 6.0, 4.3]
best fit line:
y = -16.63 + 0.01x
[2019] [5.2]
[2019, 2015] [5.2, 7.5]
[2019, 2015, 2016] [5.2, 7.5, 5.8]
[2019, 2015, 2016, 2009] [5.2, 7.5, 5.8, 7.4]
[2019, 2015, 2016, 2009, 2020] [5.2, 7.5, 5.8, 7.4, 6.2]
[2019, 2015, 2016, 2009, 2020, 2019] [5.2, 7.5, 5.8, 7.4, 6.2, 9.0]
[2019, 2015, 2016, 2009, 2020, 2019, 2010] [5.2, 7.5, 5.8, 7.4, 6.2, 9.0, 5.0]
[2019, 2015, 2016, 2009, 2020, 2019, 2010, 2020] [5.2, 7.5, 5.8, 7.4, 6.2, 9.0, 5.0, 4.4]
[2019, 2015, 2016, 2009, 2020, 2019, 2010, 2020, 2017] [5.2, 7.5, 5.8, 7.4, 6.2, 9.0, 5.0, 4.4, 4.5]
[2019, 2015, 2016, 2009, 2020, 2019, 2010, 2020, 2017, 2015] [5.2, 7.5, 5.8, 7.4, 6.2, 9.0, 5.0, 4.4, 4.5, 7.2]
best fit line:
y = -44.80 + 0.03x
[2008] [6.5]
[2008, 2017] [6.5, 8.4]
[2008, 2017, 1998] [6.5, 8.4, 6.6]
[2008, 2017, 1998, 1986] [6.5, 8.4, 6.6, 6.9]
[2008, 2017, 1998, 1986, 2015] [6.5, 8.4, 6.6, 6.9, 7.5]
[2008, 2017, 1998, 1986, 2015, 1990] [6.5, 8.4, 6.6, 6.9, 7.5, 8.8]
[2008, 2017, 1998, 1986, 2015, 1990, 1991] [6.5, 8.4, 6.6, 6.9, 7.5, 8.8, 7.2]
[2008, 2017, 1998, 1986, 2015, 1990, 1991, 1939] [6.5, 8.4, 6.6, 6.9, 7.5, 8.8, 7.2, 6.3]
[2008, 2017, 1998, 1986, 2015, 1990, 1991, 1939, 2017] [6.5, 8.4, 6.6, 6.9, 7.5, 8.8, 7.2, 6.3, 7.5]
[2008, 2017, 1998, 1986, 2015, 1990, 1991, 1939, 2017, 2013] [6.5, 8.4, 6.6, 6.9, 7.5, 8.8, 7.2, 6.3, 7.5, 8.4]
best fit line:
y = 4.28 + 0.00x
[2006] [7.5]
[2006, 1986] [7.5, 6.8]
[2006, 1986, 1982] [7.5, 6.8, 5.8]
[2006, 1986, 1982, 1986] [7.5, 6.8, 5.8, 7.1]
[2006, 1986, 1982, 1986, 2016] [7.5, 6.8, 5.8, 7.1, 8.0]
[2006, 1986, 1982, 1986, 2016, 2017] [7.5, 6.8, 5.8, 7.1, 8.0, 7.5]
[2006, 1986, 1982, 1986, 2016, 2017, 2011] [7.5, 6.8, 5.8, 7.1, 8.0, 7.5, 7.7]
[2006, 1986, 1982, 1986, 2016, 2017, 2011, 1991] [7.5, 6.8, 5.8, 7.1, 8.0, 7.5, 7.7, 6.8]
[2006, 1986, 1982, 1986, 2016, 2017, 2011, 1991, 2017] [7.5, 6.8, 5.8, 7.1, 8.0, 7.5, 7.7, 6.8, 7.2]

[2006, 1986, 1982, 1986, 2016, 2017, 2011, 1991, 2017, 2002] [7.5, 6.8, 5.8, 7.1, 8.0, 7.5, 7.7, 6.8, 7.2, 5.9]
best fit line:
y = -24.08 + 0.02x
[2021] [7.0]
[2021, 1995] [7.0, 5.4]
[2021, 1995, 2013] [7.0, 5.4, 6.0]
[2021, 1995, 2013, 2021] [7.0, 5.4, 6.0, 6.9]
[2021, 1995, 2013, 2021, 2003] [7.0, 5.4, 6.0, 6.9, 8.0]
[2021, 1995, 2013, 2021, 2003, 1933] [7.0, 5.4, 6.0, 6.9, 8.0, 5.3]
[2021, 1995, 2013, 2021, 2003, 1933, 2006] [7.0, 5.4, 6.0, 6.9, 8.0, 5.3, 5.7]
[2021, 1995, 2013, 2021, 2003, 1933, 2006, 1957] [7.0, 5.4, 6.0, 6.9, 8.0, 5.3, 5.7, 6.6]
[2021, 1995, 2013, 2021, 2003, 1933, 2006, 1957, 2014] [7.0, 5.4, 6.0, 6.9, 8.0, 5.3, 5.7, 6.6, 8.1]
[2021, 1995, 2013, 2021, 2003, 1933, 2006, 1957, 2014, 2014] [7.0, 5.4, 6.0, 6.9, 8.0, 5.3, 5.7, 6.6, 8.1, 9.0]
best fit line:
y = -11.87 + 0.01x
[2022] [6.9]
[2022, 2007] [6.9, 5.5]
[2022, 2007, 2017] [6.9, 5.5, 8.4]
[2022, 2007, 2017, 1964] [6.9, 5.5, 8.4, 6.4]
[2022, 2007, 2017, 1964, 1995] [6.9, 5.5, 8.4, 6.4, 6.7]
[2022, 2007, 2017, 1964, 1995, 2007] [6.9, 5.5, 8.4, 6.4, 6.7, 8.6]
[2022, 2007, 2017, 1964, 1995, 2007, 2013] [6.9, 5.5, 8.4, 6.4, 6.7, 8.6, 5.7]
[2022, 2007, 2017, 1964, 1995, 2007, 2013, 2016] [6.9, 5.5, 8.4, 6.4, 6.7, 8.6, 5.7, 5.8]
[2022, 2007, 2017, 1964, 1995, 2007, 2013, 2016, 2010] [6.9, 5.5, 8.4, 6.4, 6.7, 8.6, 5.7, 5.8, 7.6]
[2022, 2007, 2017, 1964, 1995, 2007, 2013, 2016, 2010, 2007] [6.9, 5.5, 8.4, 6.4, 6.7, 8.6, 5.7, 5.8, 7.6, 7.9]
best fit line:
y = -16.56 + 0.01x
[1957] [6.6]
[1957, 1962] [6.6, 7.9]
[1957, 1962, 1955] [6.6, 7.9, 6.7]
[1957, 1962, 1955, 1953] [6.6, 7.9, 6.7, 7.7]
[1957, 1962, 1955, 1953, 1960] [6.6, 7.9, 6.7, 7.7, 6.5]
[1957, 1962, 1955, 1953, 1960, 1969] [6.6, 7.9, 6.7, 7.7, 6.5, 5.9]
[1957, 1962, 1955, 1953, 1960, 1969, 1932] [6.6, 7.9, 6.7, 7.7, 6.5, 5.9, 6.1]
[1957, 1962, 1955, 1953, 1960, 1969, 1932, 1968] [6.6, 7.9, 6.7, 7.7, 6.5, 5.9, 6.1, 6.9]
[1957, 1962, 1955, 1953, 1960, 1969, 1932, 1968, 1972] [6.6, 7.9, 6.7, 7.7, 6.5, 5.9, 6.1, 6.9, 8.1]
[1957, 1962, 1955, 1953, 1960, 1969, 1932, 1968, 1972, 1959] [6.6, 7.9, 6.7, 7.7, 6.5, 5.9, 6.1, 6.9, 8.1, 7.7]

best fit line:
y = -11.40 + 0.01x
[2013] [6.0]
[2013, 2007] [6.0, 6.6]
[2013, 2007, 1995] [6.0, 6.6, 6.0]
[2013, 2007, 1995, 2021] [6.0, 6.6, 6.0, 8.9]
[2013, 2007, 1995, 2021, 2008] [6.0, 6.6, 6.0, 8.9, 7.9]
[2013, 2007, 1995, 2021, 2008, 2021] [6.0, 6.6, 6.0, 8.9, 7.9, 6.7]
[2013, 2007, 1995, 2021, 2008, 2021, 2011] [6.0, 6.6, 6.0, 8.9, 7.9, 6.7, 7.7]
[2013, 2007, 1995, 2021, 2008, 2021, 2011, 1962] [6.0, 6.6, 6.0, 8.9, 7.9, 6.7, 7.7, 7.3]
[2013, 2007, 1995, 2021, 2008, 2021, 2011, 1962, 1976] [6.0, 6.6, 6.0, 8.9, 7.9, 6.7, 7.7, 7.3, 9.8]
[2013, 2007, 1995, 2021, 2008, 2021, 2011, 1962, 1976, 2009] [6.0, 6.6, 6.0, 8.9, 7.9, 6.7, 7.7, 7.3, 9.8, 9.3]
best fit line:
y = -29.18 + 0.02x
[1951] [7.2]
[1951, 1950] [7.2, 6.5]
[1951, 1950, 1949] [7.2, 6.5, 7.1]
[1951, 1950, 1949, 1931] [7.2, 6.5, 7.1, 6.8]
[1951, 1950, 1949, 1931, 1942] [7.2, 6.5, 7.1, 6.8, 6.1]
[1951, 1950, 1949, 1931, 1942, 1946] [7.2, 6.5, 7.1, 6.8, 6.1, 7.4]
[1951, 1950, 1949, 1931, 1942, 1946, 1945] [7.2, 6.5, 7.1, 6.8, 6.1, 7.4, 6.3]
[1951, 1950, 1949, 1931, 1942, 1946, 1945, 1949] [7.2, 6.5, 7.1, 6.8, 6.1, 7.4, 6.3, 6.6]
[1951, 1950, 1949, 1931, 1942, 1946, 1945, 1949, 1947] [7.2, 6.5, 7.1, 6.8, 6.1, 7.4, 6.3, 6.6, 6.4]
[1951, 1950, 1949, 1931, 1942, 1946, 1945, 1949, 1947, 1928] [7.2, 6.5, 7.1, 6.8, 6.1, 7.4, 6.3, 6.6, 6.4, 7.5]
best fit line:
y = 23.14 + -0.01x
[2017] [6.9]
[2017, 1964] [6.9, 6.7]
[2017, 1964, 2003] [6.9, 6.7, 6.7]
[2017, 1964, 2003, 1977] [6.9, 6.7, 6.7, 6.0]
[2017, 1964, 2003, 1977, 1970] [6.9, 6.7, 6.7, 6.0, 7.6]
[2017, 1964, 2003, 1977, 1970, 1970] [6.9, 6.7, 6.7, 6.0, 7.6, 5.6]
[2017, 1964, 2003, 1977, 1970, 1970, 1973] [6.9, 6.7, 6.7, 6.0, 7.6, 5.6, 7.6]
[2017, 1964, 2003, 1977, 1970, 1970, 1973, 1995] [6.9, 6.7, 6.7, 6.0, 7.6, 5.6, 7.6, 8.0]
[2017, 1964, 2003, 1977, 1970, 1970, 1973, 1995, 2016] [6.9, 6.7, 6.7, 6.0, 7.6, 5.6, 7.6, 8.0, 7.9]
[2017, 1964, 2003, 1977, 1970, 1970, 1973, 1995, 2016, 2007] [6.9, 6.7, 6.7, 6.0, 7.6, 5.6, 7.6, 8.0, 7.9, 3.5]
best fit line:
y = -13.80 + 0.01x

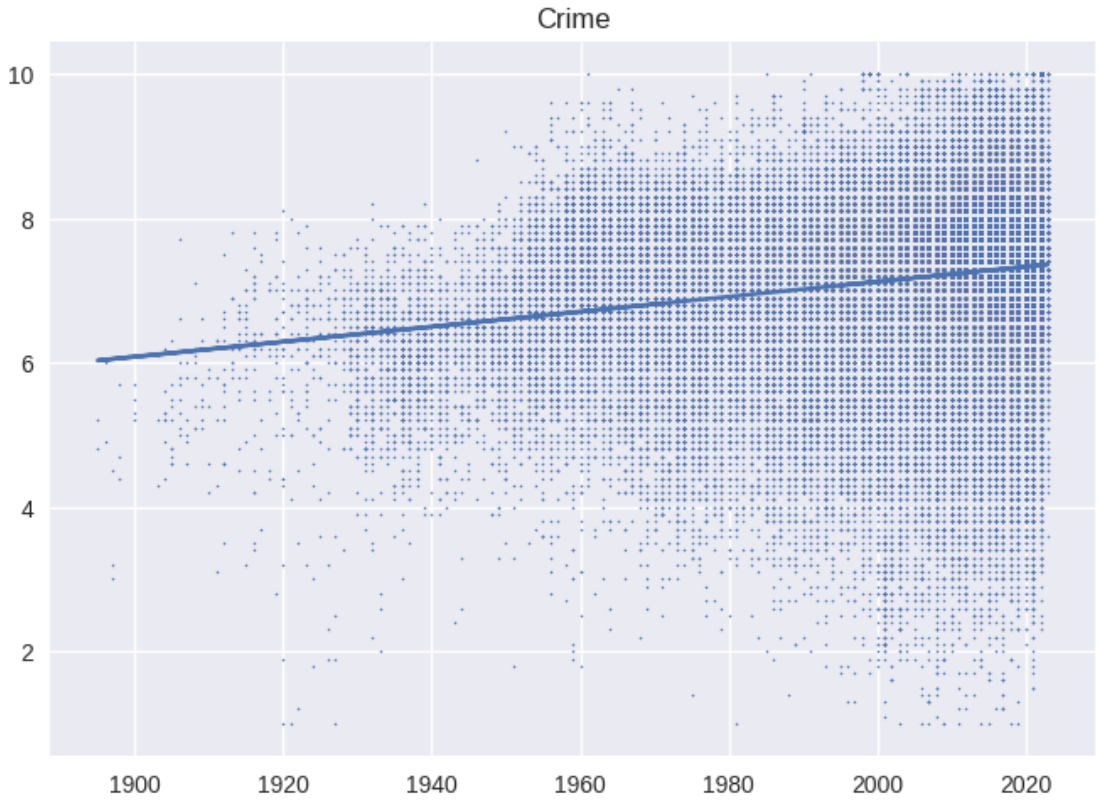
[1989] [6.2]
[1989, 1999] [6.2, 7.1]
[1989, 1999, 2013] [6.2, 7.1, 7.4]
[1989, 1999, 2013, 2010] [6.2, 7.1, 7.4, 7.7]
[1989, 1999, 2013, 2010, 1996] [6.2, 7.1, 7.4, 7.7, 8.0]
[1989, 1999, 2013, 2010, 1996, 1983] [6.2, 7.1, 7.4, 7.7, 8.0, 7.2]
[1989, 1999, 2013, 2010, 1996, 1983, 2012] [6.2, 7.1, 7.4, 7.7, 8.0, 7.2, 6.0]
[1989, 1999, 2013, 2010, 1996, 1983, 2012, 1982] [6.2, 7.1, 7.4, 7.7, 8.0, 7.2, 6.0, 5.6]
[1989, 1999, 2013, 2010, 1996, 1983, 2012, 1982, 2015] [6.2, 7.1, 7.4, 7.7, 8.0, 7.2, 6.0, 5.6, 4.3]
[1989, 1999, 2013, 2010, 1996, 1983, 2012, 1982, 2015, 1988] [6.2, 7.1, 7.4, 7.7, 8.0, 7.2, 6.0, 5.6, 4.3, 5.9]
best fit line:
y = -52.38 + 0.03x
[2013] [6.5]
[2013, 2001] [6.5, 7.3]
[2013, 2001, 2013] [6.5, 7.3, 7.0]
[2013, 2001, 2013, 2018] [6.5, 7.3, 7.0, 7.8]
[2013, 2001, 2013, 2018, 2018] [6.5, 7.3, 7.0, 7.8, 6.6]
[2013, 2001, 2013, 2018, 2018, 2009] [6.5, 7.3, 7.0, 7.8, 6.6, 6.2]
[2013, 2001, 2013, 2018, 2018, 2009, 2020] [6.5, 7.3, 7.0, 7.8, 6.6, 6.2, 6.3]
[2013, 2001, 2013, 2018, 2018, 2009, 2020, 1994] [6.5, 7.3, 7.0, 7.8, 6.6, 6.2, 6.3, 5.2]
[2013, 2001, 2013, 2018, 2018, 2009, 2020, 1994, 2021] [6.5, 7.3, 7.0, 7.8, 6.6, 6.2, 6.3, 5.2, 6.4]
[2013, 2001, 2013, 2018, 2018, 2009, 2020, 1994, 2021, 2020] [6.5, 7.3, 7.0, 7.8, 6.6, 6.2, 6.3, 5.2, 6.4, 7.6]
best fit line:
y = -27.32 + 0.02x
[2014] [4.7]
[2014, 2008] [4.7, 7.9]
[2014, 2008, 2021] [4.7, 7.9, 6.7]
[2014, 2008, 2021, 2015] [4.7, 7.9, 6.7, 3.1]
[2014, 2008, 2021, 2015, 1976] [4.7, 7.9, 6.7, 3.1, 9.8]
[2014, 2008, 2021, 2015, 1976, 1991] [4.7, 7.9, 6.7, 3.1, 9.8, 4.7]
[2014, 2008, 2021, 2015, 1976, 1991, 2016] [4.7, 7.9, 6.7, 3.1, 9.8, 4.7, 7.3]
[2014, 2008, 2021, 2015, 1976, 1991, 2016, 1989] [4.7, 7.9, 6.7, 3.1, 9.8, 4.7, 7.3, 5.9]
[2014, 2008, 2021, 2015, 1976, 1991, 2016, 1989, 2015] [4.7, 7.9, 6.7, 3.1, 9.8, 4.7, 7.3, 5.9, 5.2]
[2014, 2008, 2021, 2015, 1976, 1991, 2016, 1989, 2015, 1989] [4.7, 7.9, 6.7, 3.1, 9.8, 4.7, 7.3, 5.9, 5.2, 7.2]
best fit line:
y = -1.38 + 0.00x
[2021] [4.7]
[2021, 2015] [4.7, 7.5]

[2021, 2015, 2013] [4.7, 7.5, 6.5]
[2021, 2015, 2013, 2016] [4.7, 7.5, 6.5, 5.8]
[2021, 2015, 2013, 2016, 2019] [4.7, 7.5, 6.5, 5.8, 9.0]
[2021, 2015, 2013, 2016, 2019, 2010] [4.7, 7.5, 6.5, 5.8, 9.0, 5.0]
[2021, 2015, 2013, 2016, 2019, 2010, 2018] [4.7, 7.5, 6.5, 5.8, 9.0, 5.0, 9.0]
[2021, 2015, 2013, 2016, 2019, 2010, 2018, 2005] [4.7, 7.5, 6.5, 5.8, 9.0, 5.0, 9.0, 4.8]
[2021, 2015, 2013, 2016, 2019, 2010, 2018, 2005, 2022] [4.7, 7.5, 6.5, 5.8, 9.0, 5.0, 9.0, 4.8, 7.1]
[2021, 2015, 2013, 2016, 2019, 2010, 2018, 2005, 2022, 2022] [4.7, 7.5, 6.5, 5.8, 9.0, 5.0, 9.0, 4.8, 7.1, 8.6]
best fit line:
y = -1.64 + 0.00x
[2006] [7.5]
[2006, 2021] [7.5, 8.9]
[2006, 2021, 1986] [7.5, 8.9, 6.8]
[2006, 2021, 1986, 1982] [7.5, 8.9, 6.8, 5.8]
[2006, 2021, 1986, 1982, 1986] [7.5, 8.9, 6.8, 5.8, 7.1]
[2006, 2021, 1986, 1982, 1986, 2016] [7.5, 8.9, 6.8, 5.8, 7.1, 8.0]
[2006, 2021, 1986, 1982, 1986, 2016, 1964] [7.5, 8.9, 6.8, 5.8, 7.1, 8.0, 6.4]
[2006, 2021, 1986, 1982, 1986, 2016, 1964, 2017] [7.5, 8.9, 6.8, 5.8, 7.1, 8.0, 6.4, 7.5]
[2006, 2021, 1986, 1982, 1986, 2016, 1964, 2017, 1991] [7.5, 8.9, 6.8, 5.8, 7.1, 8.0, 6.4, 7.5, 6.8]
[2006, 2021, 1986, 1982, 1986, 2016, 1964, 2017, 1991, 2017] [7.5, 8.9, 6.8, 5.8, 7.1, 8.0, 6.4, 7.5, 6.8, 7.2]
best fit line:
y = -25.62 + 0.02x

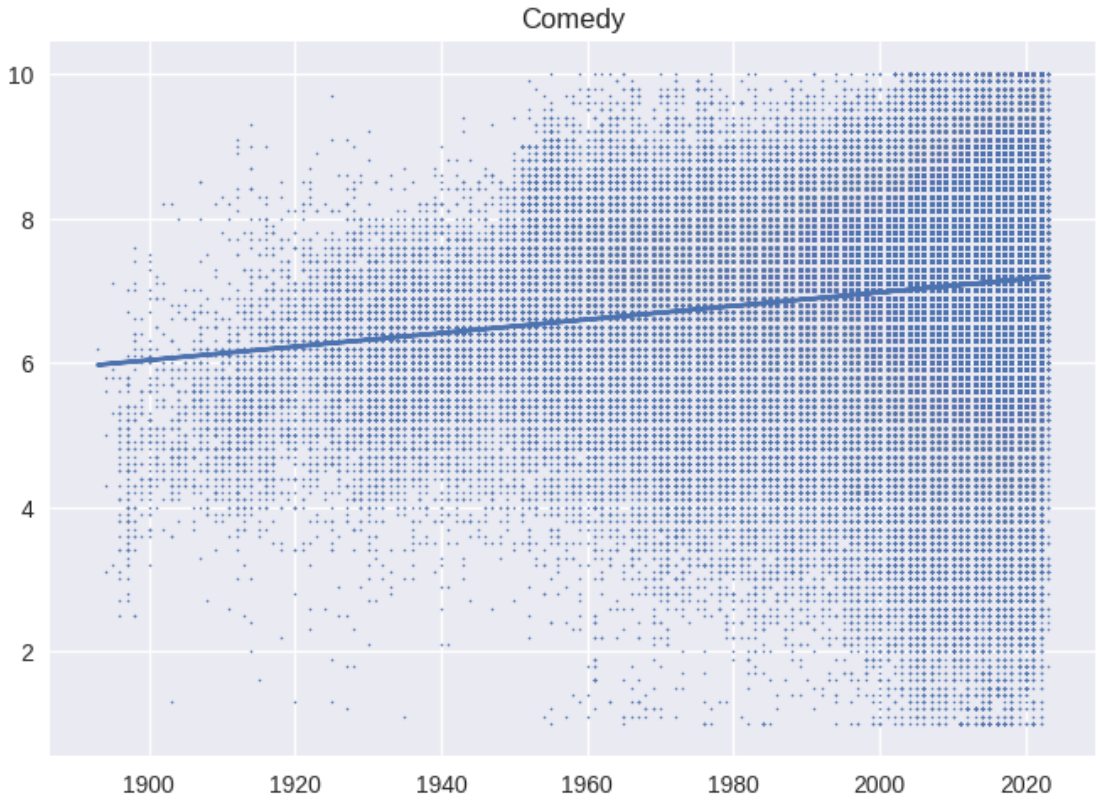


Putting all this info onto one graph is extremely messy. Lets clean it up a bit

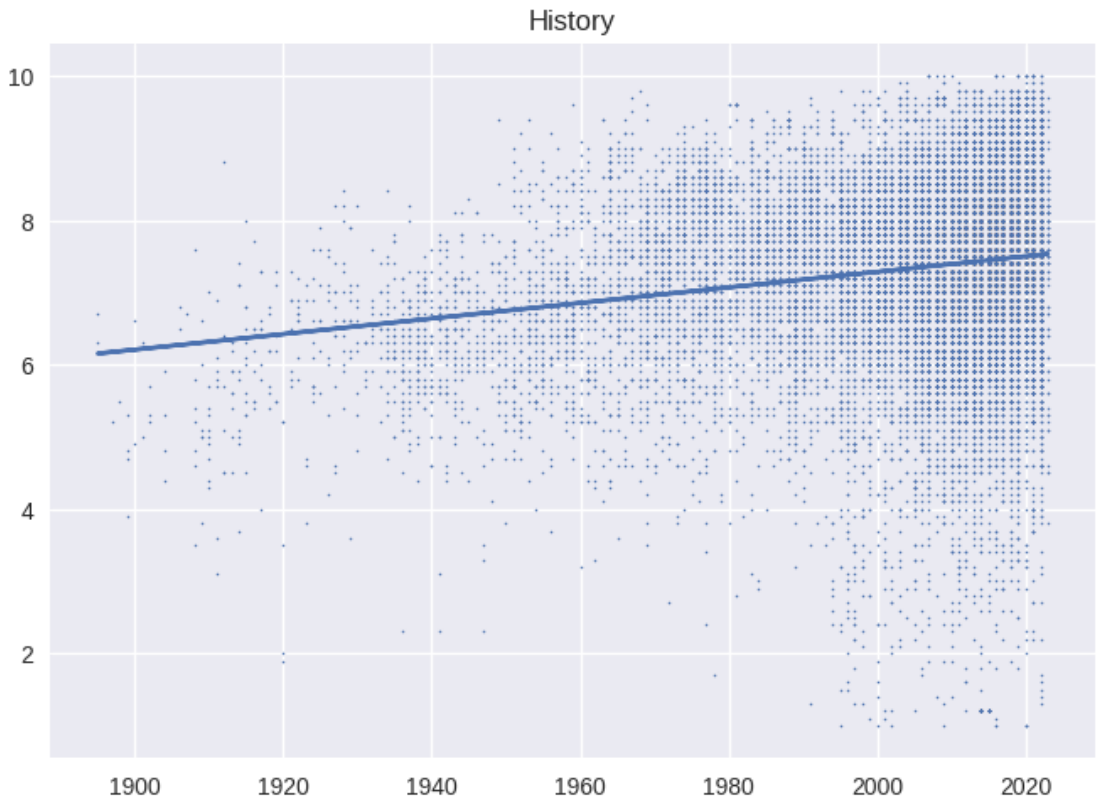
```
import matplotlib.pyplot as plt
bestfits = {}
for genre in genre_split:
  year = []
  rating = []

  for index, row in genre_split[genre].iterrows():
    if row['startYear'] != '\\N':
      year.append(int(row['startYear']))
      rating.append(float(row['averageRating']))

  plt.title(genre)
  plt.scatter(year, rating, s=.6)

  a, b = best_fit(year, rating)
  bestfits[genre] = [a,b]
  yfit = [a + b * yeari for yeari in year]
  plt.plot(year, yfit)
  plt.show()
  plt.clf()

best fit line:
y = -13.67 + 0.01x
```



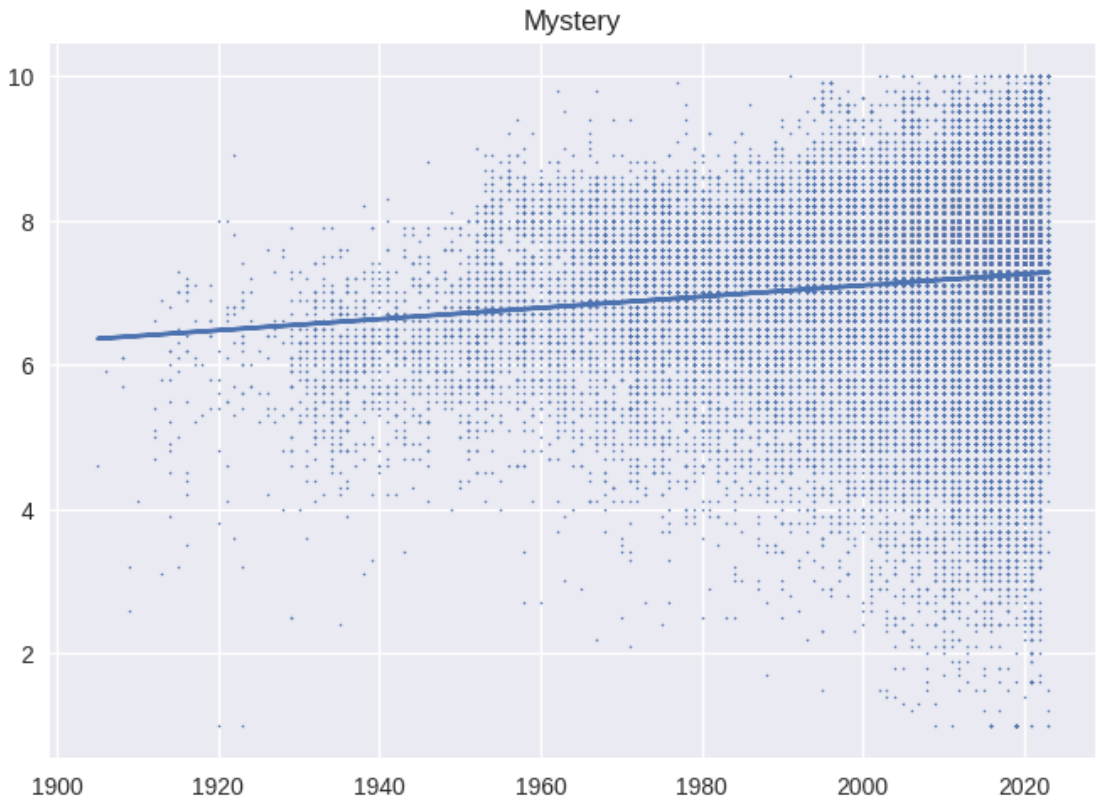
best fit line:
y = -11.80 + 0.01x



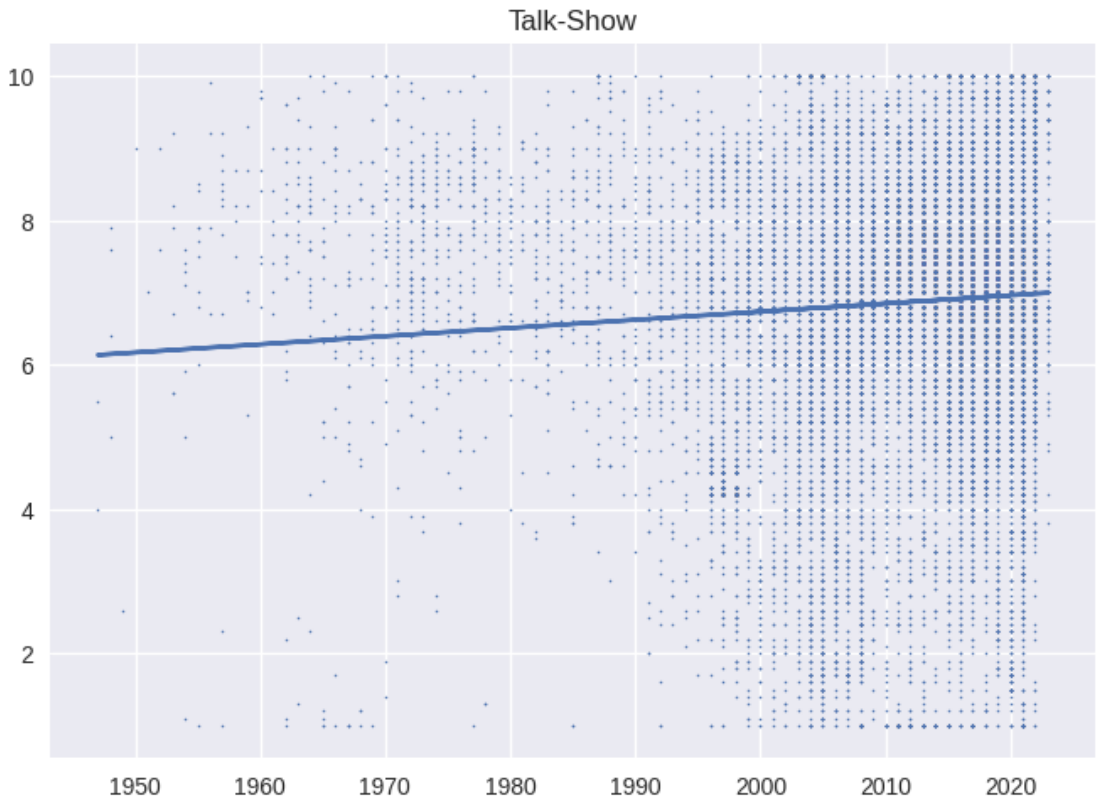
best fit line:
y = -14.26 + 0.01x



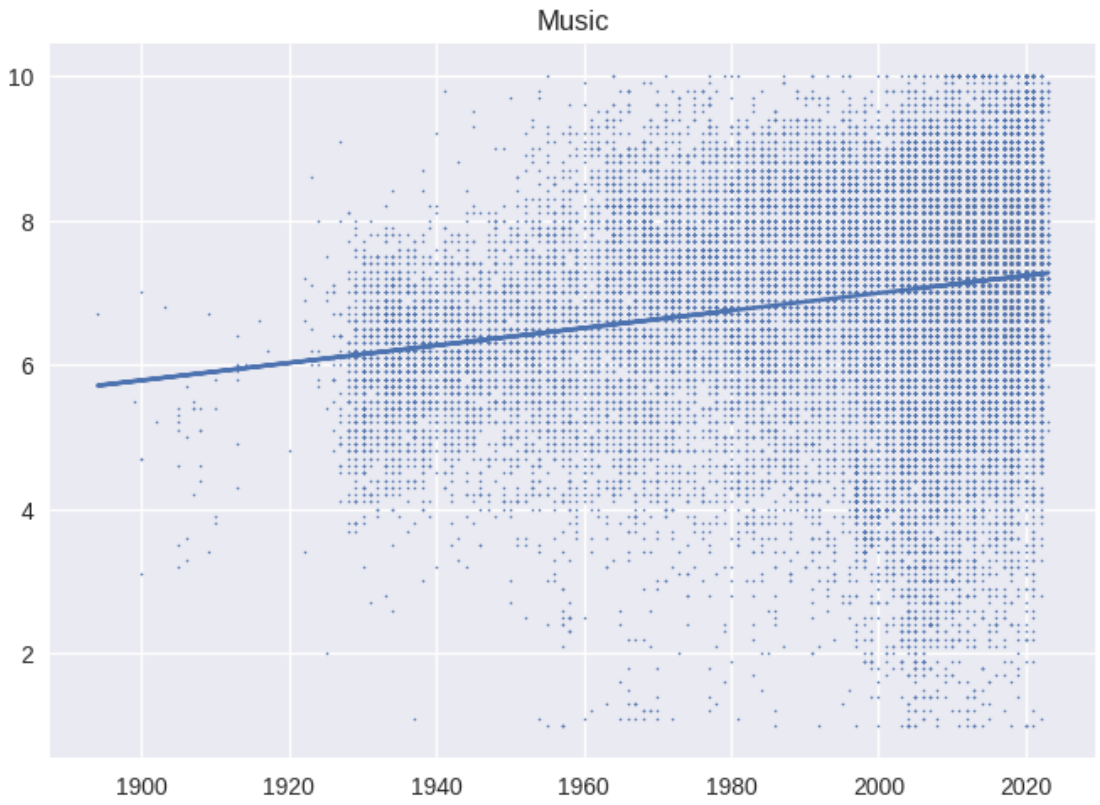
best fit line:
y = -8.47 + 0.01x



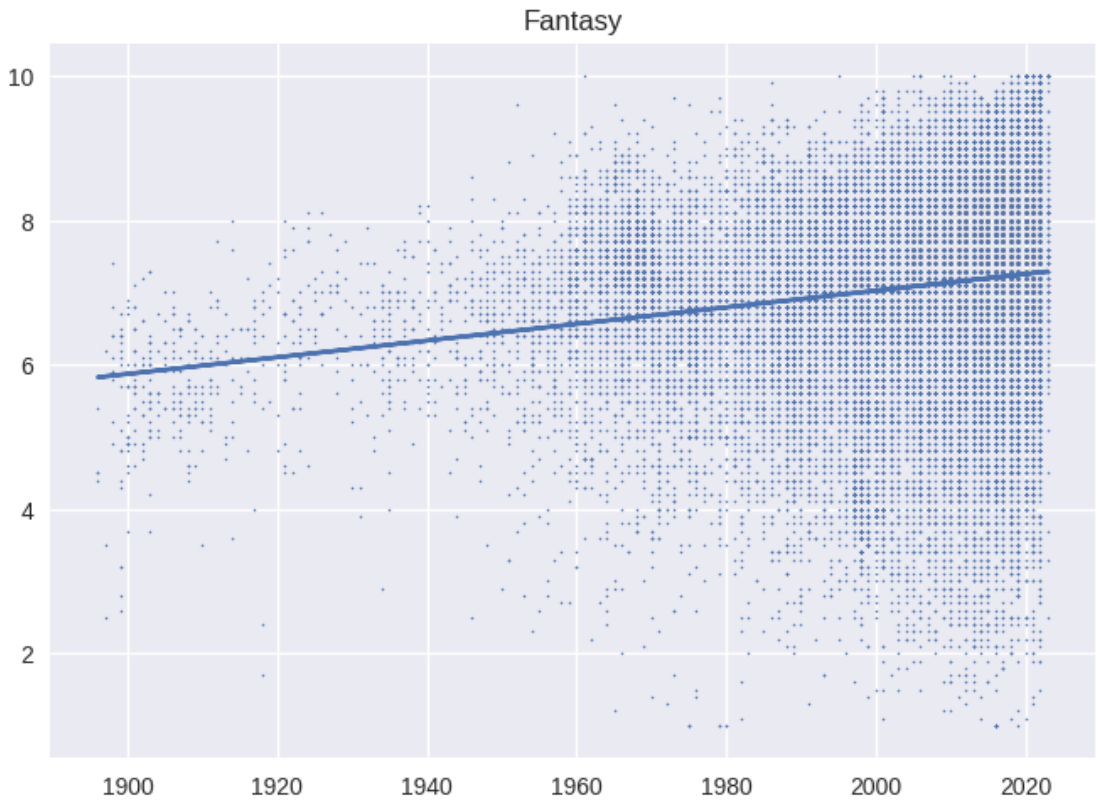
best fit line:
y = -15.88 + 0.01x



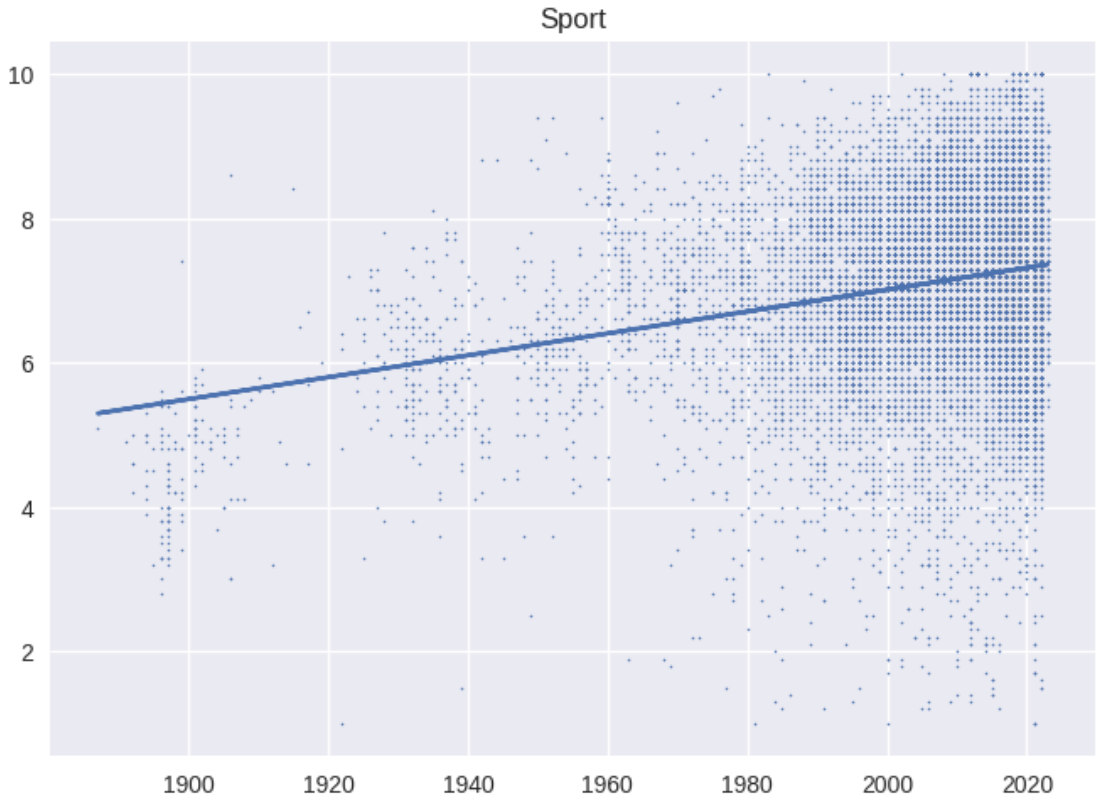
best fit line:
y = -17.17 + 0.01x



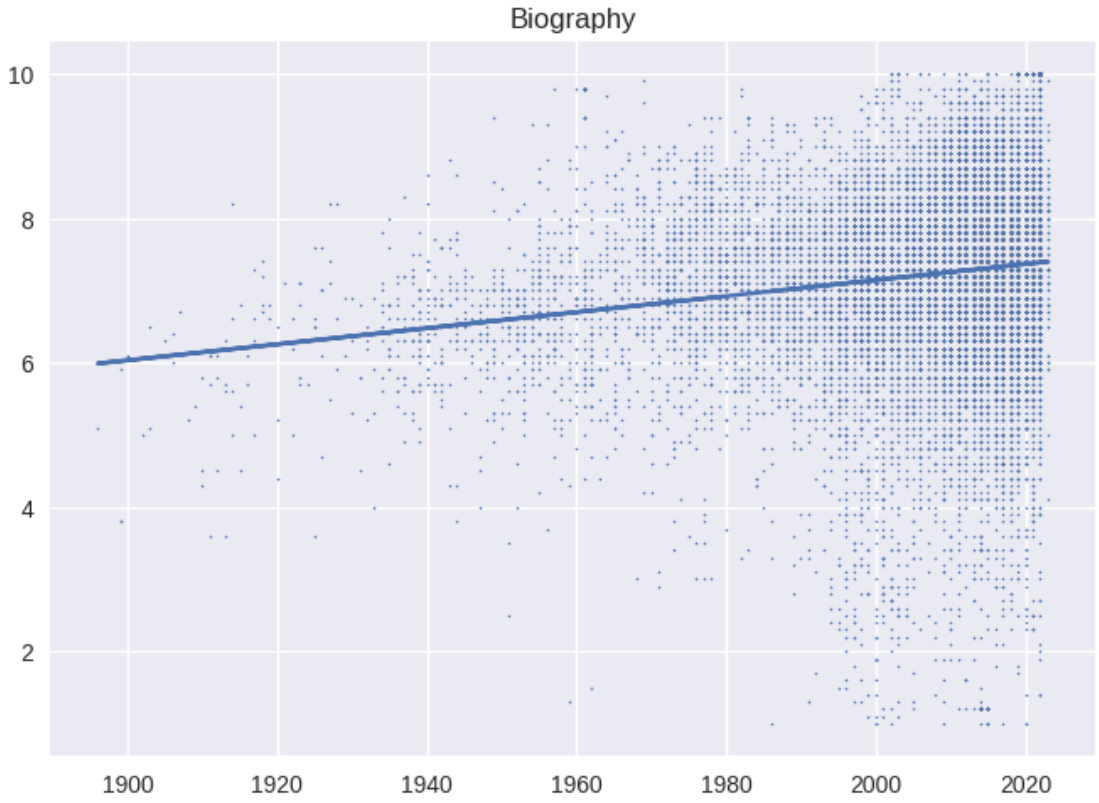
best fit line:
y = -16.04 + 0.01x



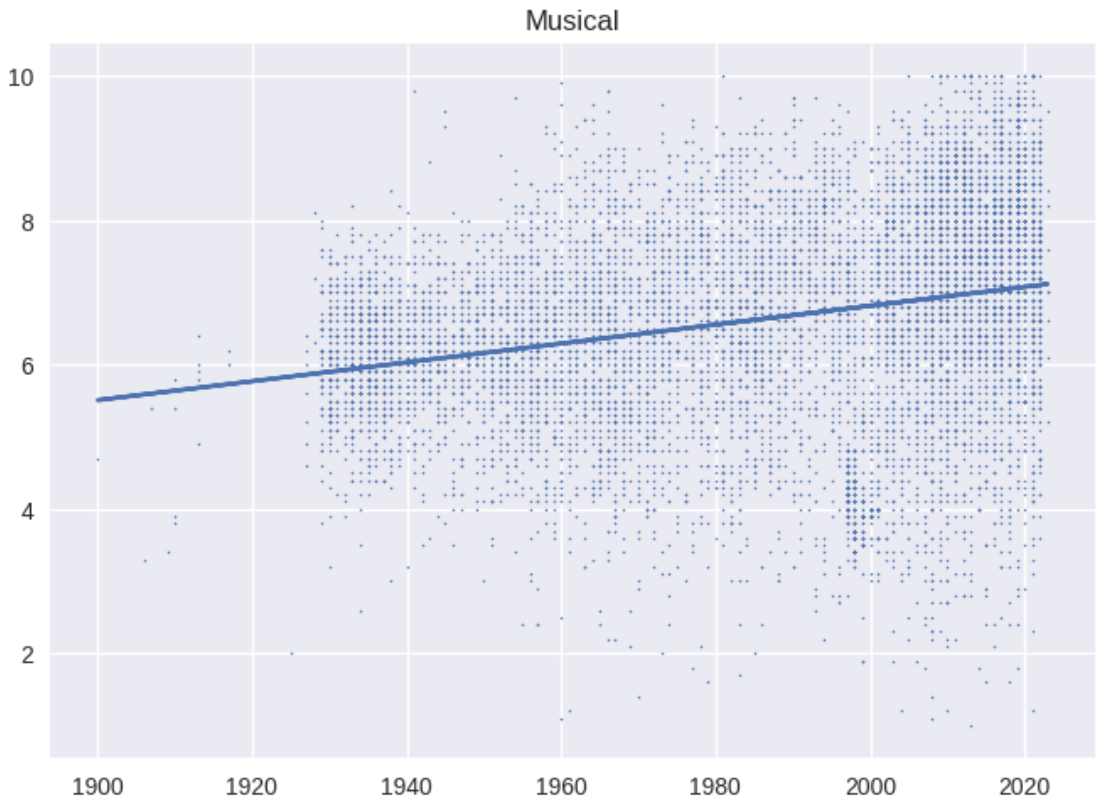
best fit line:
y = -23.33 + 0.02x



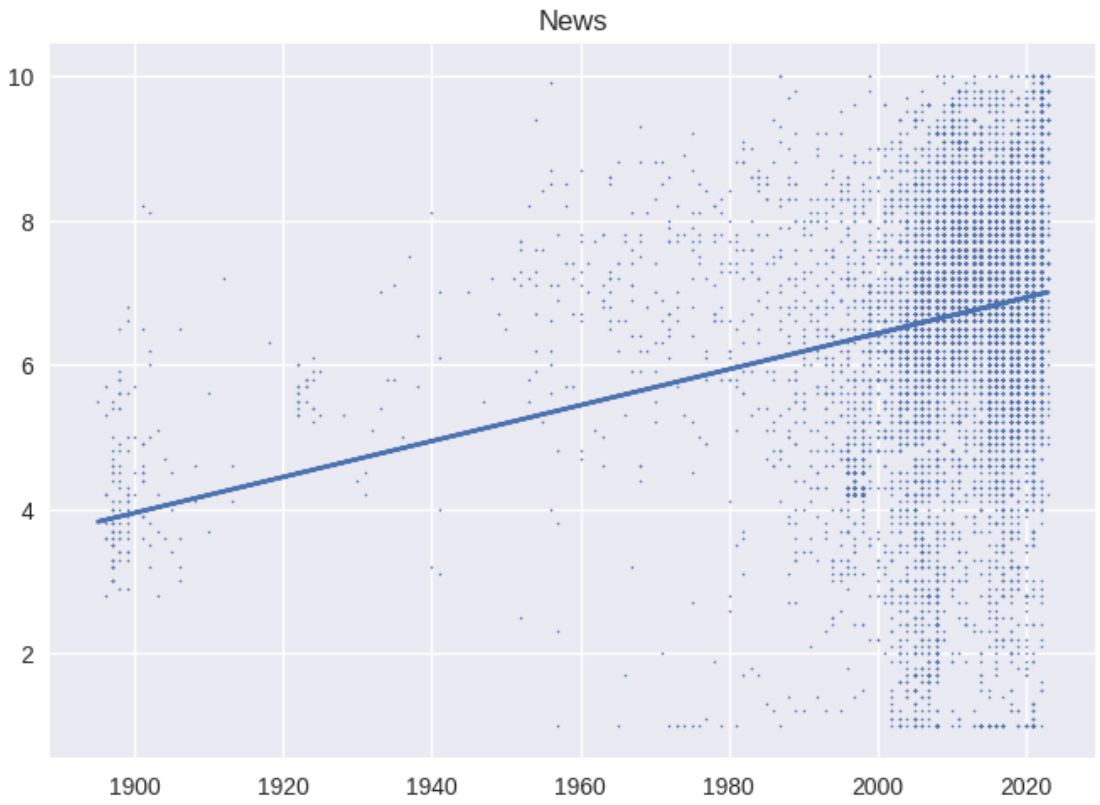
best fit line:
y = -15.07 + 0.01x



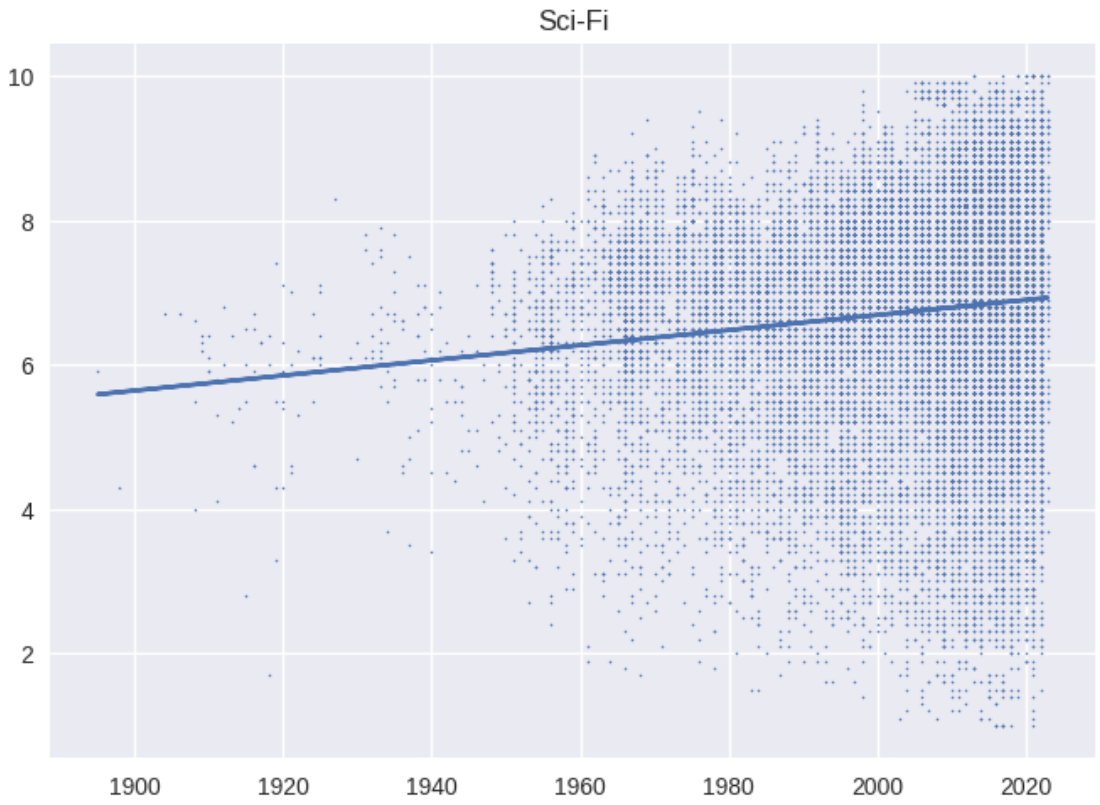
best fit line:
y = -19.32 + 0.01x



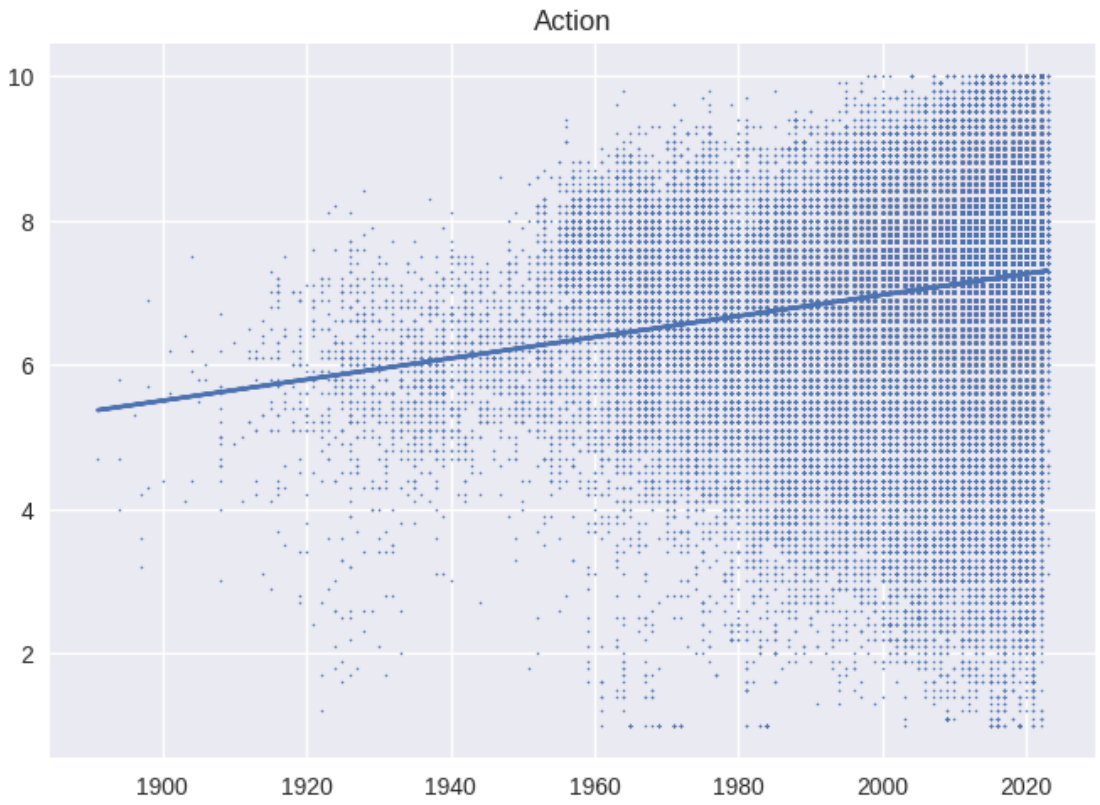
best fit line:
y = -43.24 + 0.02x



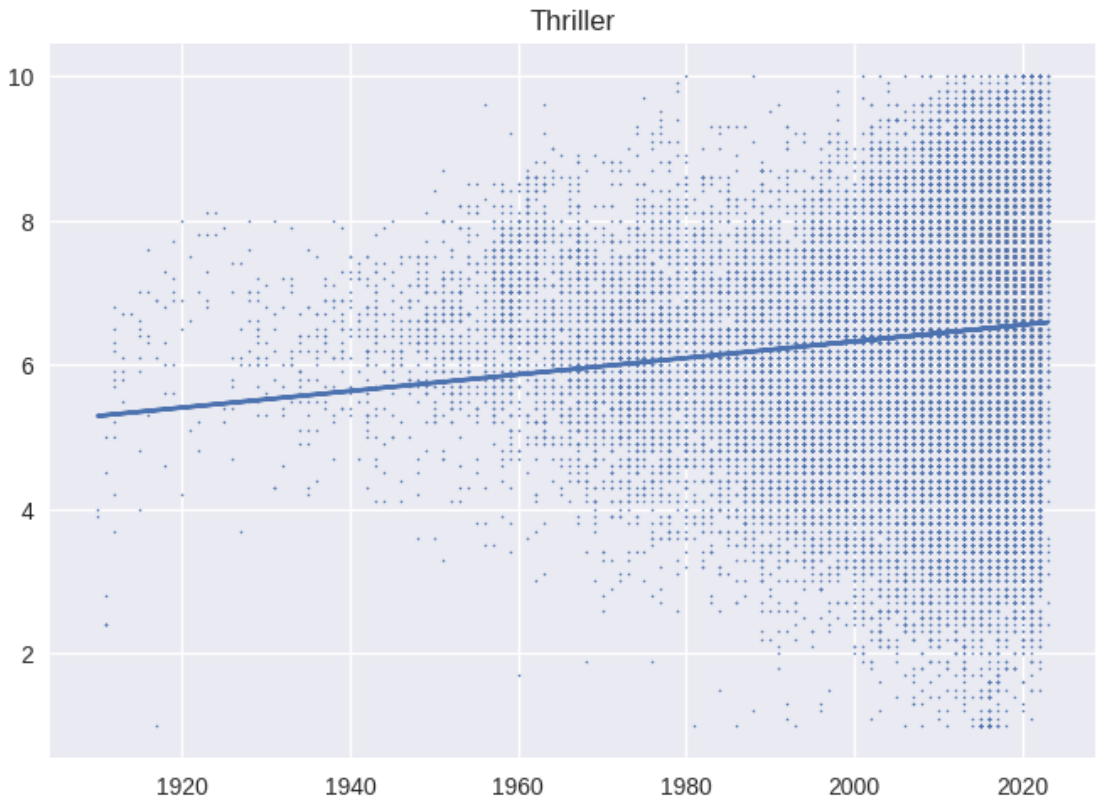

```
best fit line:
y = -14.25 + 0.01x
```



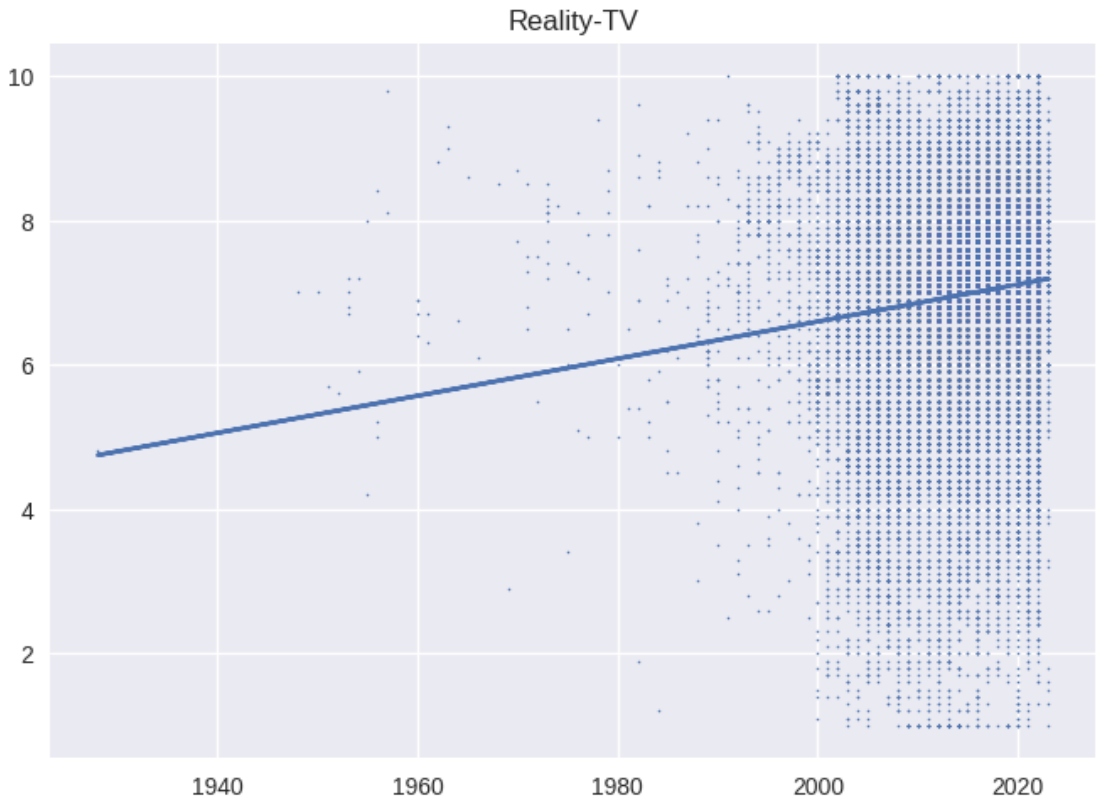
best fit line:
y = -22.31 + 0.01x



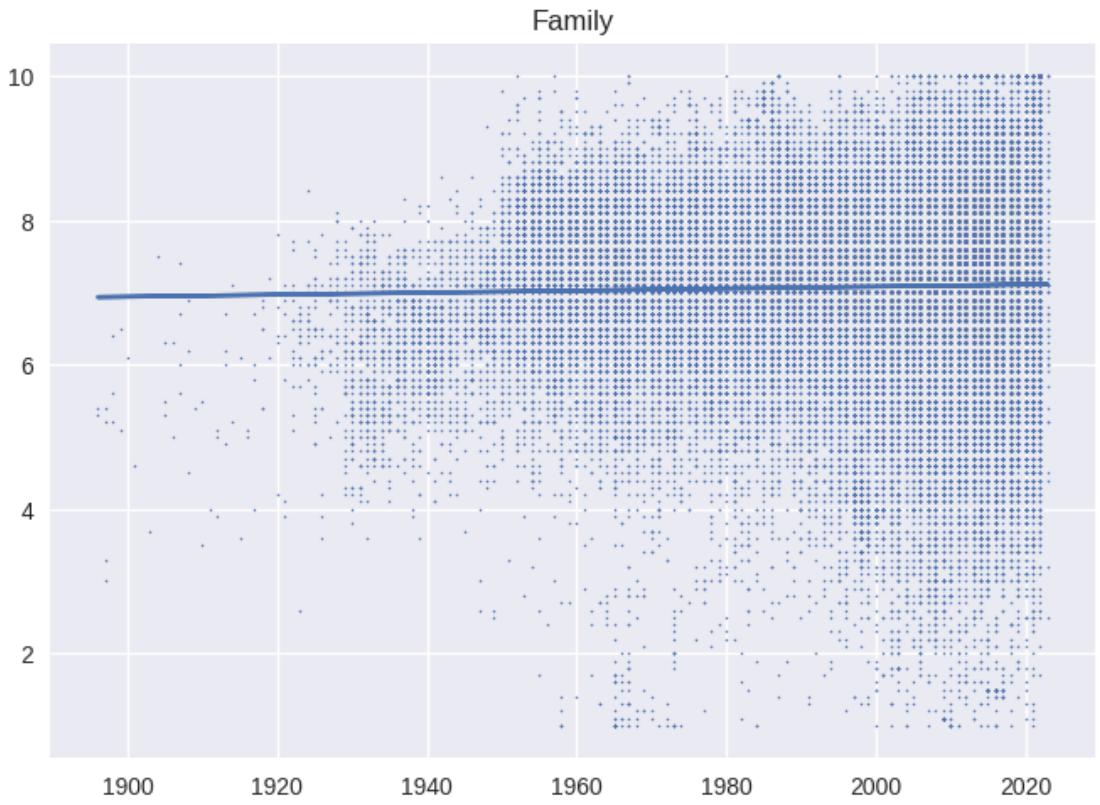
best fit line:
y = -16.63 + 0.01x



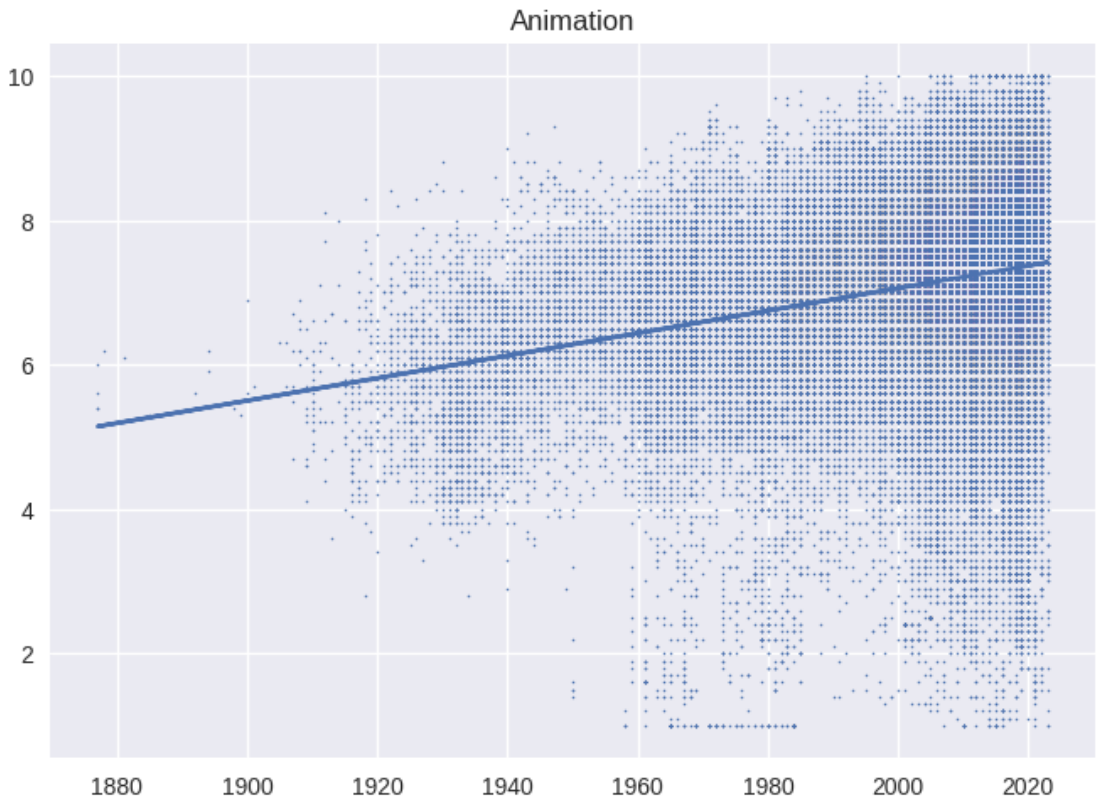

```
best fit line:
y = -44.80 + 0.03x
```



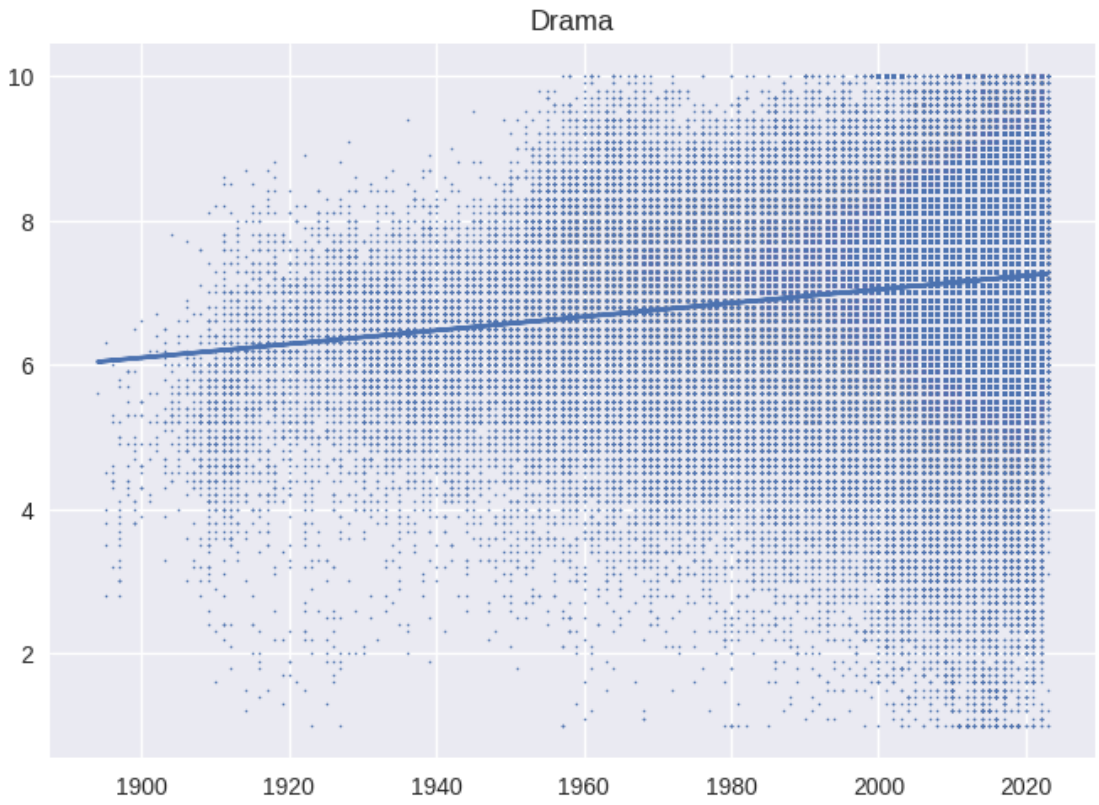

```
best fit line:
y = 4.28 + 0.00x
```



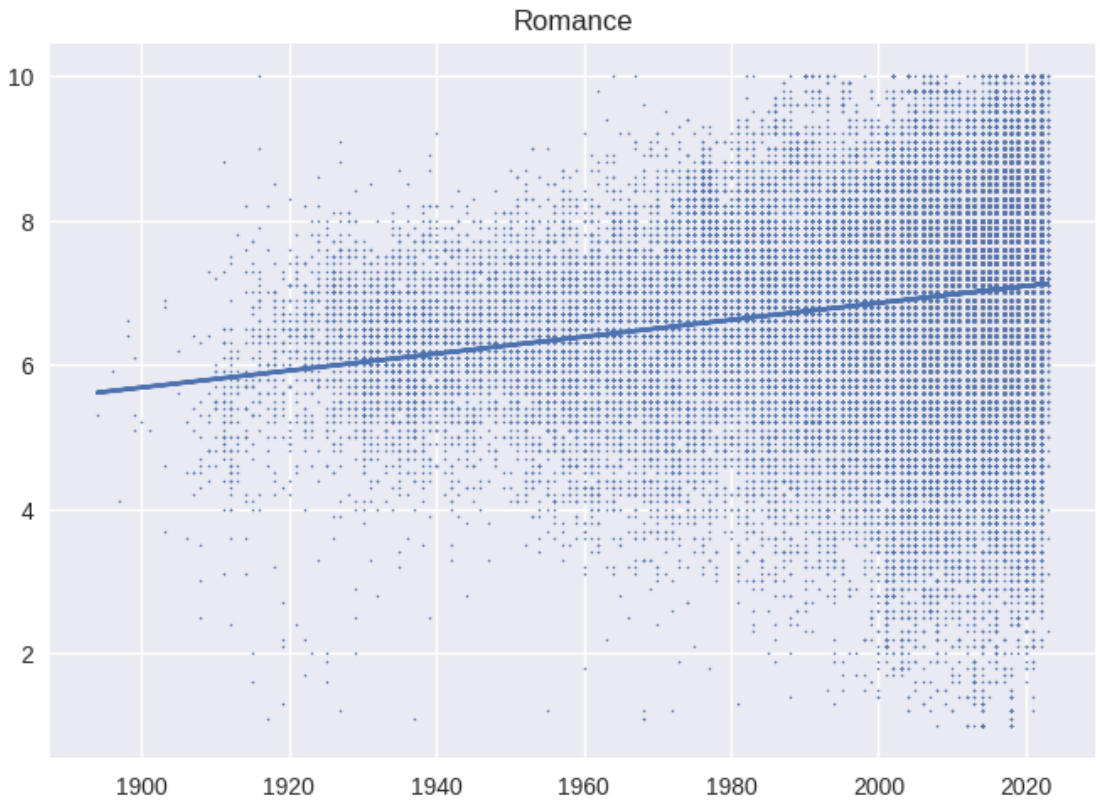
best fit line:
y = -24.08 + 0.02x



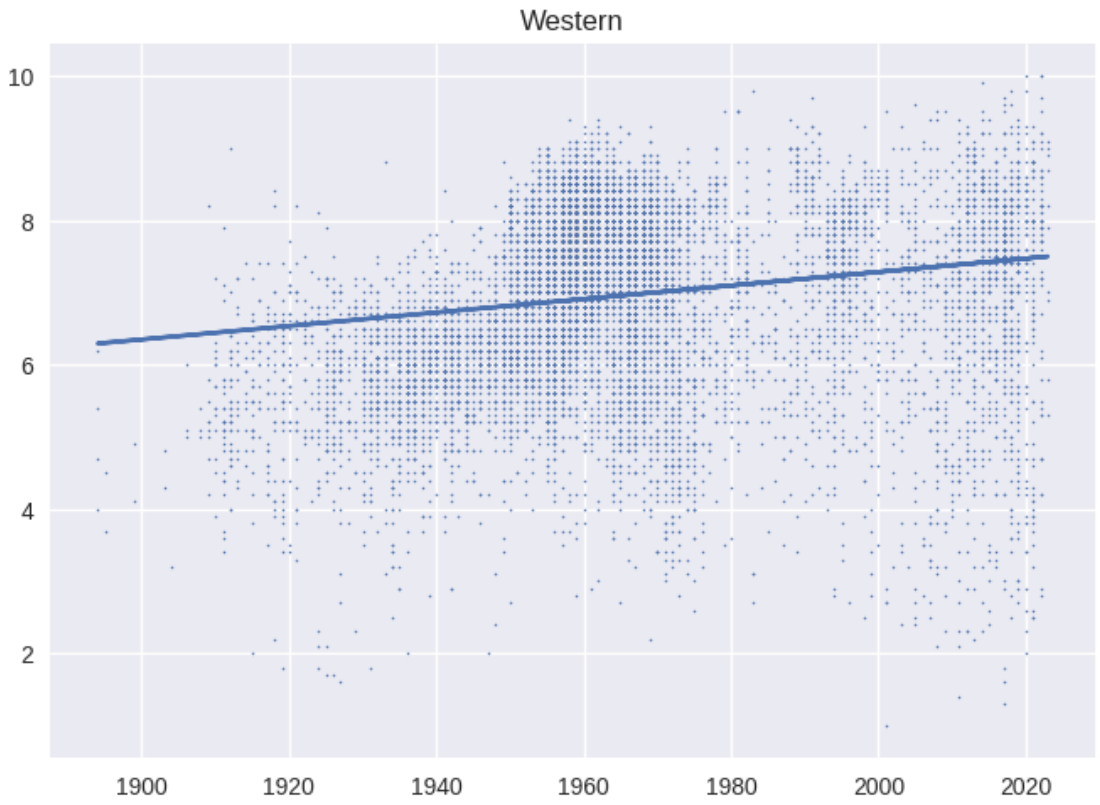

```
best fit line:
y = -11.87 + 0.01x
```



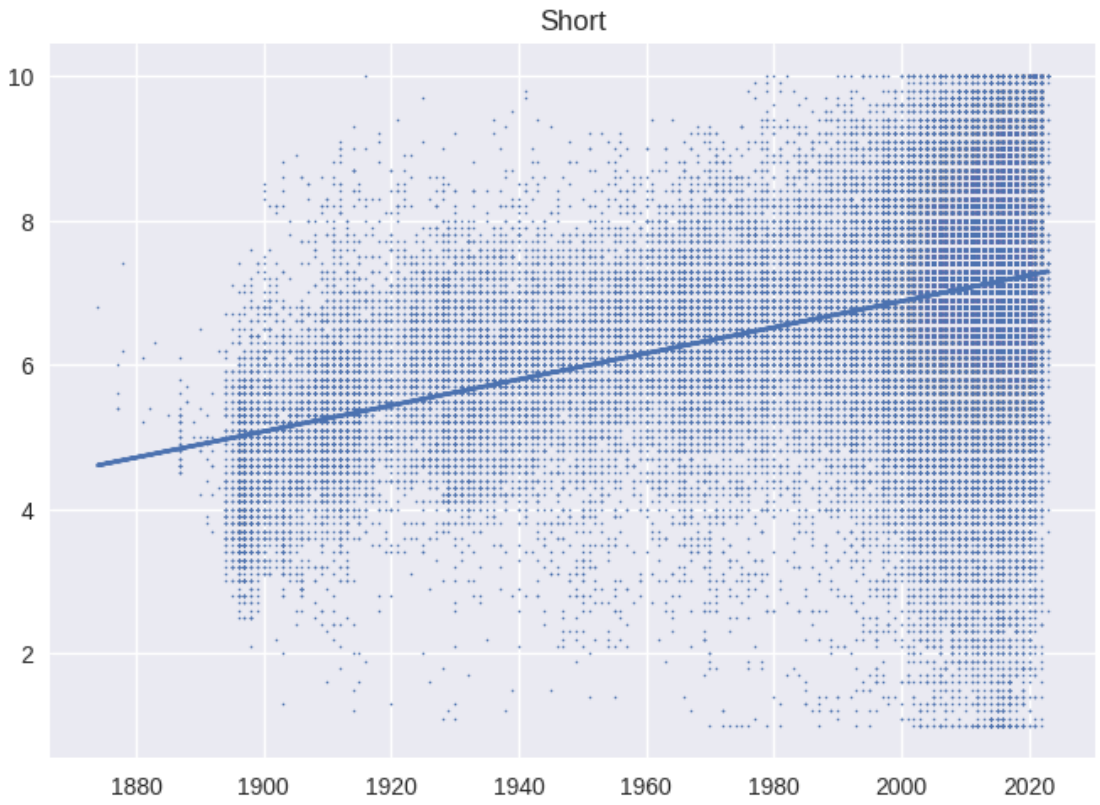
best fit line:
y = -16.56 + 0.01x



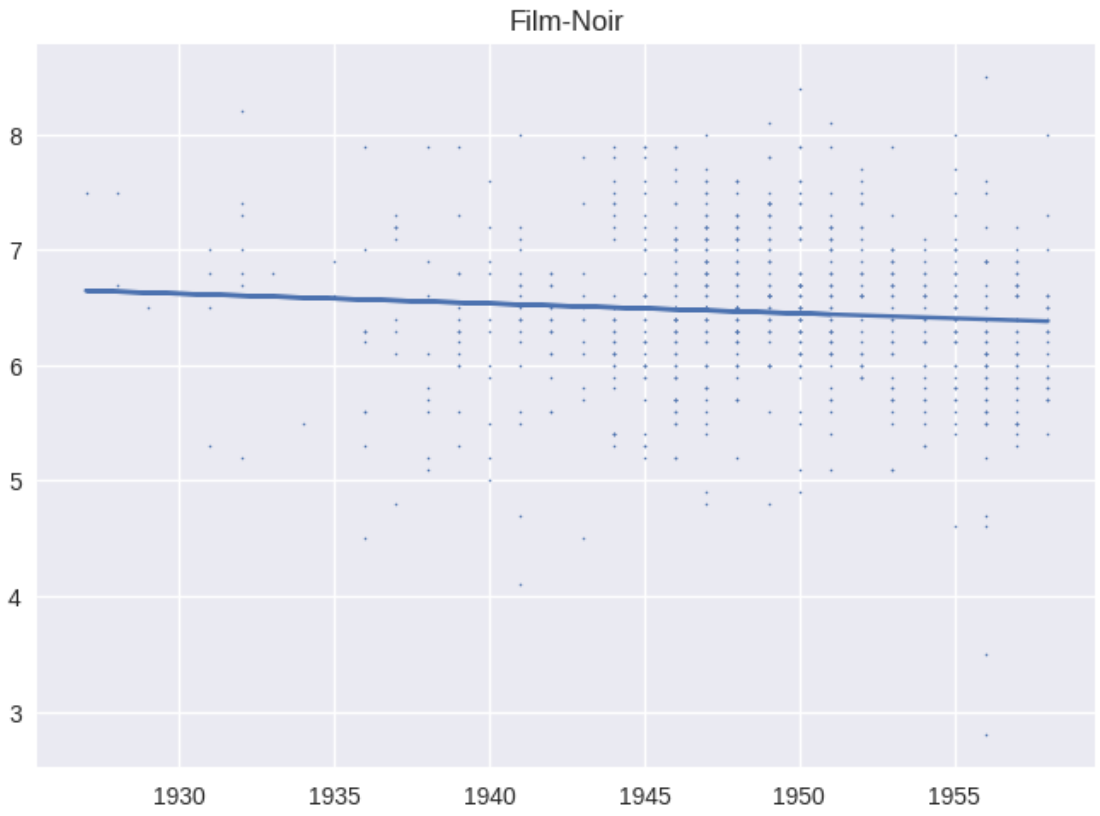

```
best fit line:
y = -11.40 + 0.01x
```



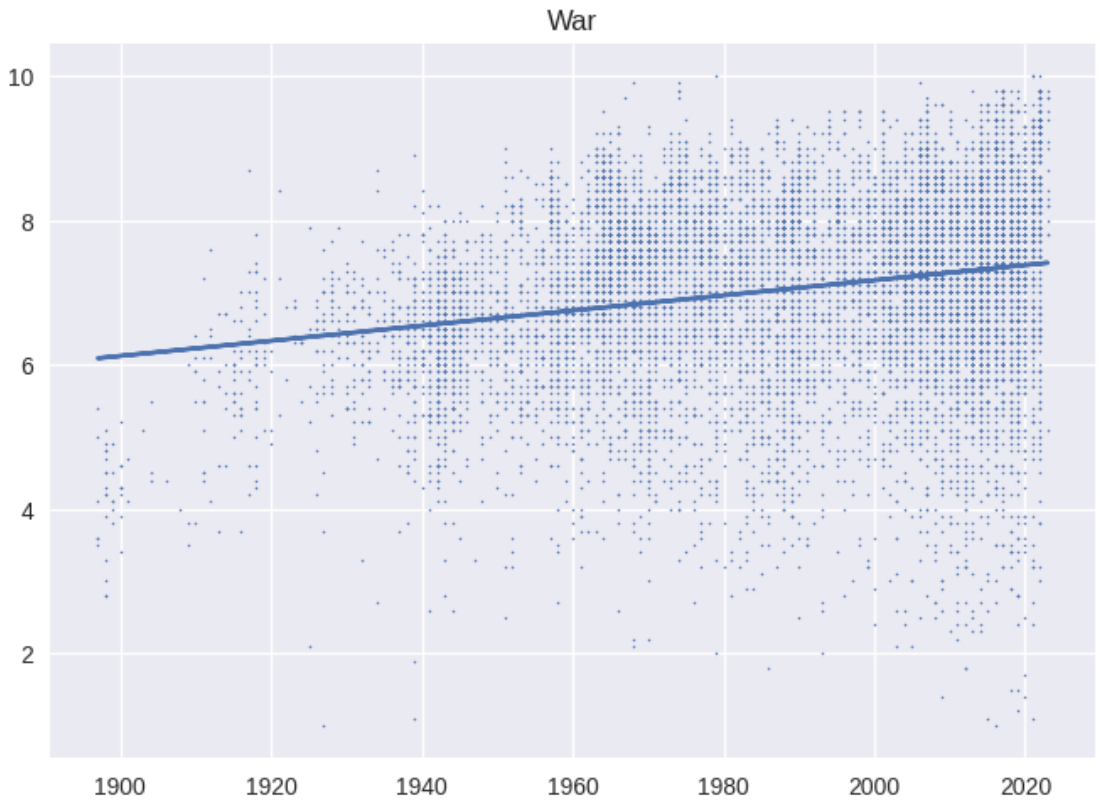
best fit line:
y = -29.18 + 0.02x



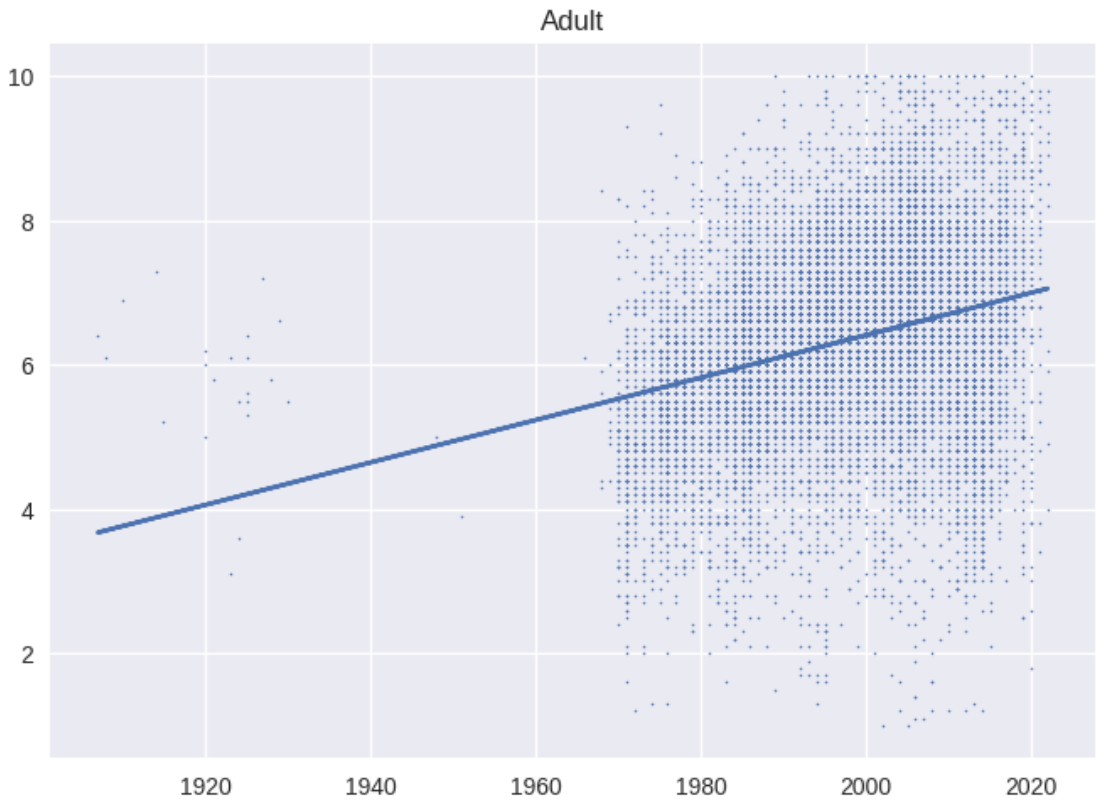
best fit line:
y = 23.14 + -0.01x



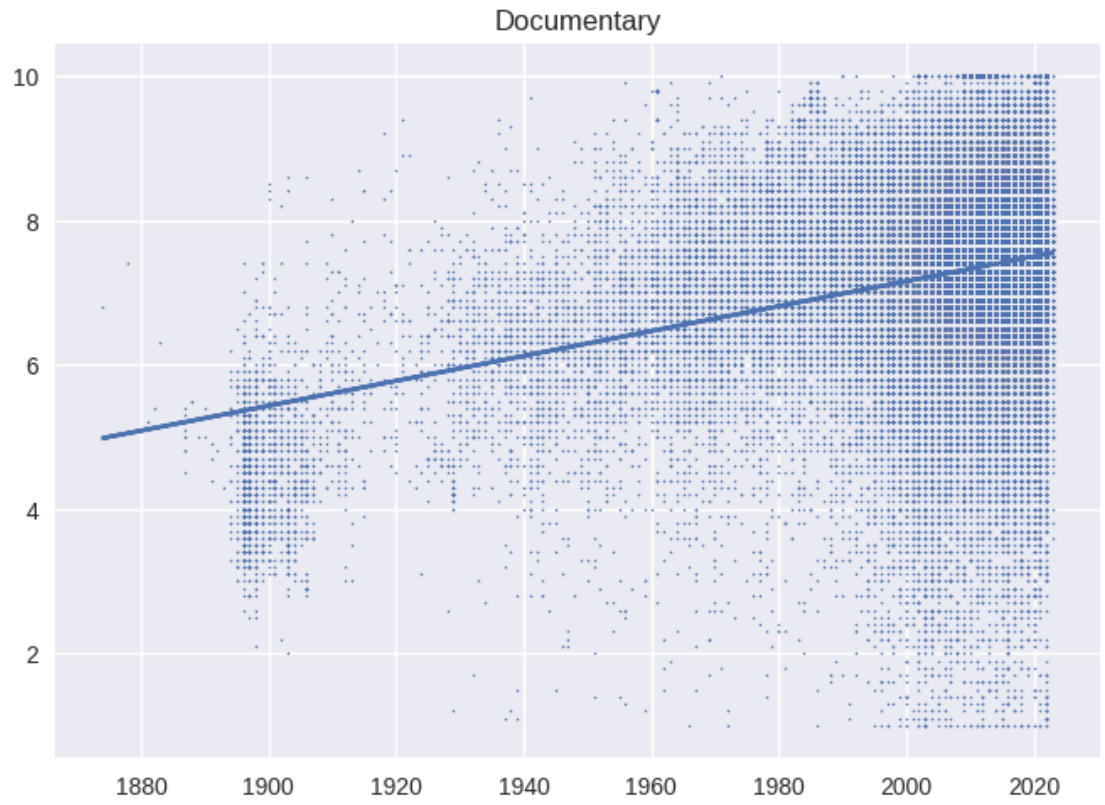

```
best fit line:
y = -13.80 + 0.01x
```



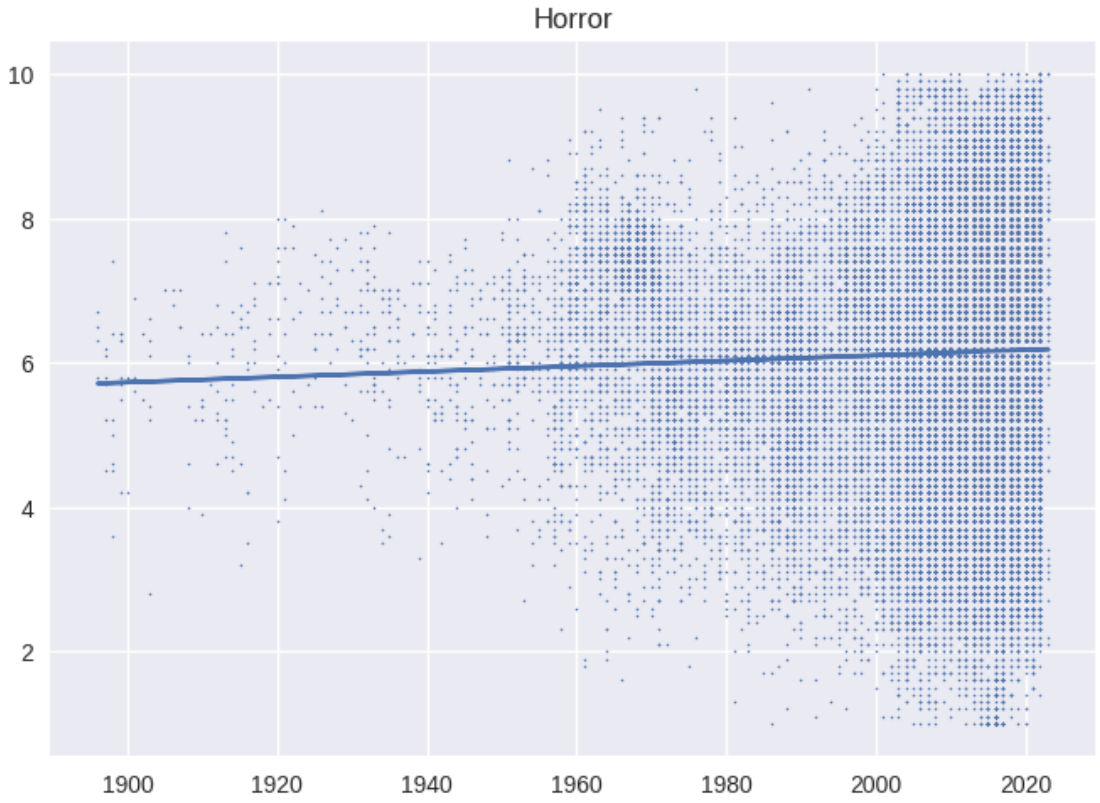
best fit line:
y = -52.38 + 0.03x



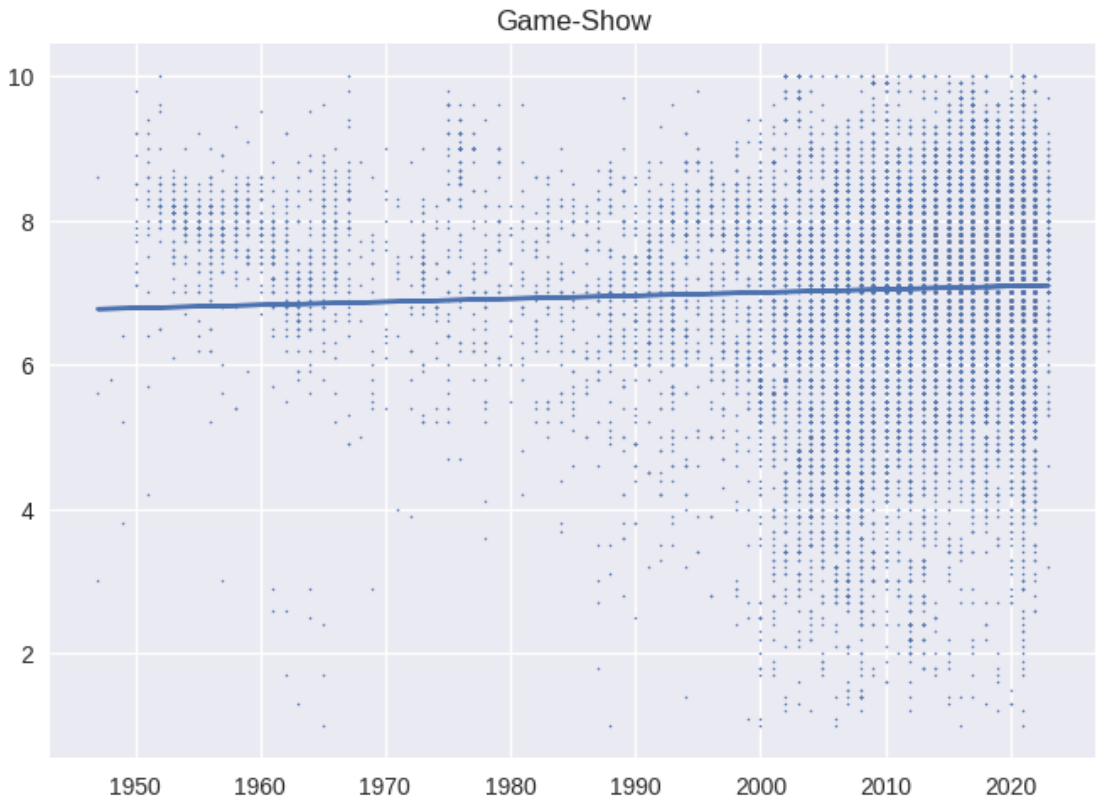
best fit line:
y = -27.32 + 0.02x



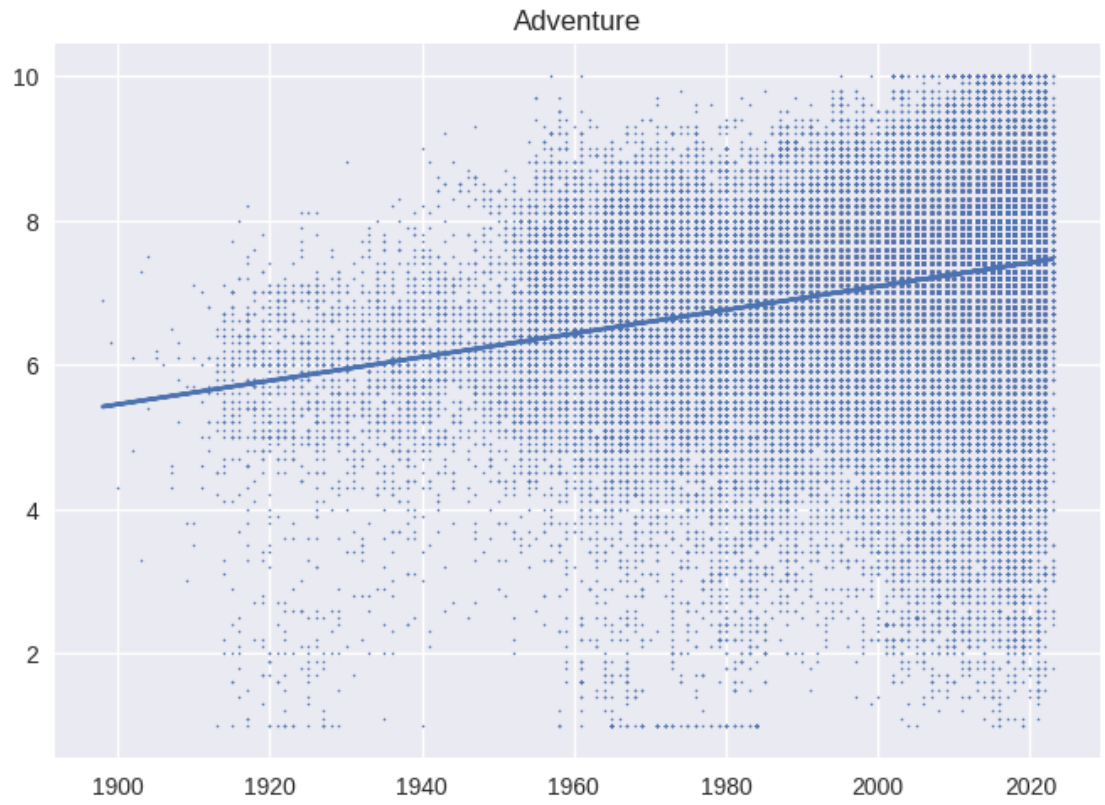

```
best fit line:
y = -1.38 + 0.00x
```




```
best fit line:
y = -1.64 + 0.00x
```



best fit line:
y = -25.62 + 0.02x




```
<Figure size 800x550 with 0 Axes>
```

```
print(bestfits)
highestb = 0
highestbg = ""
highesta = 0
highestag = ""

lowestb = 0
lowestbg = ""
lowesta = 0
lowestag = ""

for genre in bestfits:
  if bestfits[genre][0] > highestb:
    highestb = bestfits[genre][0]
    highestbg = genre
  if bestfits[genre][1] > highesta:
    highesta = bestfits[genre][1]
    highestag = genre
  if bestfits[genre][0] < lowestb:
    lowestb = bestfits[genre][0]
    lowestbg = genre
  if bestfits[genre][1] < lowesta:
    lowesta = bestfits[genre][1]
```

```
    lowestag = genre

print(f"Highest starting value: {highestbg} with {highestb}\nMost
improvement over the years: {highestag} with {highesta}")
print(f"Lowest starting value: {lowestbg} with {lowestb}\nMost
degraded over the years: {lowestag} with {lowesta}")
```

```
{'Crime': [-13.671131278847193, 0.010395788689438055], 'Comedy': [-
11.795263229268372, 0.009838963677214 97], 'History': [-
14.26194920677155, 0.010773194898180736], 'Mystery': [-
8.466216847828097, 0.007781740083392541], 'Talk-Show': [-
15.88283760667808, 0.011307874820479095], 'Music': [-17.1673809843522,
0.01207869555941277], 'Fantasy': [-16.04128647052476,
0.011532544104470056], 'Sport': [-23.33018962057568,
0.015170432738985837], 'Biography': [-15.070878855792985,
0.011106796646604895], 'Musical': [-19.32058360953547,
0.013067473863163316], 'News': [-43.23757993147416,
0.024833859310266243], 'Sci-Fi': [-14.246102994280857,
0.010466334309625723], 'Action': [-22.312620581447334,
0.014638582852197775], 'Thriller': [-16.62725084935351,
0.011474376965610 87], 'Reality-TV': [-44.803702557104444,
0.025699811711743206], 'Family': [4.282853321573924,
0.0013987242237185034], 'Animation': [-24.080781062233218,
0.015569486341862723], 'Drama': [-11.871671560238065,
0.009456469018614395], 'Romance': [-16.55990523238937,
0.011706751384420212], 'Western': [-11.402267414815412,
0.009342341883102066], 'Short': [-29.181550214166915,
0.01802888092125245], 'Film-Noir': [23.14437657145985, -
0.008561425435171945], 'War': [-13.802720984943392,
0.010485570230823362], 'Adult': [-52.37935764784966,
0.029393270694227626], 'Documentary': [-27.323167878926267,
0.017238287709639758], 'Horror': [-1.3837196689972426,
0.0037421571538692896], 'Game-Show': [-1.6358118914238595,
0.004317070650578428], 'Adventure': [-25.621622480216825,
0.016353280429719237]}
Highest starting value: Film-Noir with 23.14437657145985
Most improvement over the years: Adult with 0.029393270694227626
Lowest starting value: Adult with -52.37935764784966
Most degraded over the years: Film-Noir with -0.008561425435171945
```

This is interesting information since the it implies the starting values are determining future popularity, what was once the most popular, is now least, and what was unpopular, is now popular

Lets verify by applying this to our test set. we've previously been using our training data. Lets see if we get similar results.

Lets save the data from our bestfit lines for comparisson later.

```
testbestfits = bestfits
```

```
# Generate list of all genres
genre_list = []

genres = test['genres'].unique()
for genresgroup in genres:
  if genresgroup != genresgroup:
    print(genresgroup)
  genre_sep = genresgroup.split(',')
  genre_list = genre_list + genre_sep

# Find unique genres from our data.
list_set = set(genre_list)
unique_list_genre = (list(list_set))
unique_list_genre.remove("\\N")
for genre in unique_list_genre:
  print(genre)

print("\n\n\n")

# Create dictionary for all genres
genre_split = {}
# Add each genre as a key, and its dictionary as the value
for genre in unique_list_genre:
  genre_split[genre] =
train.loc[(train['genres'].str.contains(genre))]

print(genre_split['Horror'].iloc[:1])
```

```
Crime
Comedy
Mystery
History
Talk-Show
Music
Fantasy
Sport
Biography
Musical
News
Sci-Fi
Action
Thriller
Reality-TV
Family
Animation
Drama
Romance
Western
Short
Film-Noir
```

```
War
Adult
Documentary
Horror
Game-Show
Adventure
```

```
          tconst titleType primaryTitle originalTitle isAdult
startYear  \
959068  tt3443808     video     Chubbies      Chubbies       0
2014

       endYear runtimeMinutes                genres  averageRating
numVotes
959068      \N             89  Comedy,Horror,Sci-Fi            4.7
91
```

```
genre_split_avg = {}
# for each genre
for genre in unique_list_genre:
  #initalize values
  avg = 0
  count = 0
  #declare working genre
  print(f"GENRE: {genre}:")
  #iterate over dataframe to find the average rating, and number of
movies
  for index, row in genre_split[genre].iterrows():
    #print(row['primaryTitle'], row['averageRating'])
    avg = avg + row['averageRating']
    count = count + 1
  avg = avg/count
  print(f"\nAverage: {avg}, Number: {count}\n\n")
  #split dataframe above and below average
  genre_split_avg['B'+genre] = genre_split[genre][genre_split[genre]
['averageRating'] <= avg]
  genre_split_avg['A'+genre] = genre_split[genre][genre_split[genre]
['averageRating'] > avg]
```

```
GENRE: Crime:

Average: 7.141819996043043, Number: 116253


GENRE: Comedy:

Average: 6.988683059152979, Number: 342840
```

GENRE: Mystery:

Average: 7.132382305579212, Number: 52785

GENRE: History:

Average: 7.346030372933504, Number: 31212

GENRE: Talk-Show:

Average: 6.873209364104323, Number: 30115

GENRE: Music:

Average: 6.96755002939705, Number: 45923

GENRE: Fantasy:

Average: 7.065640483583272, Number: 45411

GENRE: Sport:

Average: 7.118973884908732, Number: 21635

GENRE: Biography:

Average: 7.2209120550270836, Number: 22389

GENRE: Musical:

Average: 6.627235810746995, Number: 10589

GENRE: News:

Average: 6.693392980041285, Number: 14530

GENRE: Sci-Fi:

Average: 6.725301953205907, Number: 28978

GENRE: Action:

Average: 7.018534937412822, Number: 135571

GENRE: Thriller:

Average: 6.406603268148968, Number: 44796

GENRE: Reality-TV:

Average: 6.961207515452866, Number: 57109

GENRE: Family:

Average: 7.072674595913373, Number: 78696

GENRE: Animation:

Average: 7.099453413088723, Number: 129897

GENRE: Drama:

Average: 7.054393452064071, Number: 368116

GENRE: Romance:

Average: 6.8746902292904535, Number: 83126

GENRE: Western:

Average: 6.956647010524729, Number: 13397

GENRE: Short:

```
Average: 6.841250328383142, Number: 129422


GENRE: Film-Noir:

Average: 6.461405835543761, Number: 754


GENRE: War:

Average: 7.046074884307963, Number: 11885


GENRE: Adult:

Average: 6.333515848390834, Number: 16437


GENRE: Documentary:

Average: 7.270386523509826, Number: 135050


GENRE: Horror:

Average: 6.123888107585453, Number: 45508


GENRE: Game-Show:

Average: 7.046774860062344, Number: 26619


GENRE: Adventure:

Average: 7.093430364868467, Number: 119961
```

```
print(f"{genre_split_avg['AAction'].iloc[:1]}\n\n\
n{genre_split_avg['BAction'].iloc[:1]}")
```

```
           tconst  titleType                      primaryTitle  \
635056  tt1383574  tvEpisode  Haruka vs Shu! The Final Battle!!

                            originalTitle isAdult startYear endYear  \
635056  Haruka vs Shu! The Final Battle!!       0      2006      \N
```

```
       runtimeMinutes                     genres  averageRating
numVotes
635056             22  Action,Adventure,Animation            7.5
100


          tconst  titleType            primaryTitle
originalTitle  \
586424  tt1281261  tvEpisode  Flash Back/The Warrior  Flash Back/The
Warrior

       isAdult startYear endYear runtimeMinutes
genres  \
586424       0      1982      \N             24
Action,Adventure,Animation

        averageRating  numVotes
586424            5.8        18
```

```
# rating_data = { 'genre': [title,year,rating],[title,year,rating],
'genre': [title,year,rating] }
rating_data = {}

for genre in genre_split:
  title_year_rating = []
  #print(genre, '->', genre_split[genre].iloc[:1])
  for index, row in genre_split[genre].iterrows():

    title = row['primaryTitle']
    year = row['startYear']
    rating = row['averageRating']
    tyr = [title,year,rating]
    title_year_rating.append(tyr)

  rating_data[genre] = title_year_rating


def best_fit(X, Y):

    xbar = sum(X)/len(X)
    ybar = sum(Y)/len(Y)
    n = len(X) # or len(Y)

    numer = sum([xi*yi for xi,yi in zip(X, Y)]) - n * xbar * ybar
    denum = sum([xi**2 for xi in X]) - n * xbar**2

    b = numer / denum
    a = ybar - b * xbar
```

```
    print('best fit line:\ny = {:.2f} + {:.2f}x'.format(a, b))

    return a, b

import matplotlib.pyplot as plt

for genre in genre_split:
  year = []
  rating = []

  for index, row in genre_split[genre].iterrows():
    if row['startYear'] != '\\N':

      year.append(int(row['startYear']))
      rating.append(float(row['averageRating']))
      if len(year) <= 10:
        print(year, rating)

  plt.scatter(year, rating, s=.1, alpha=.75)

  a, b = best_fit(year, rating)

  yfit = [a + b * yeari for yeari in year]
  plt.plot(year, yfit)
```

```
[1995] [5.4]
[1995, 2003] [5.4, 8.0]
[1995, 2003, 2018] [5.4, 8.0, 7.3]
[1995, 2003, 2018, 2014] [5.4, 8.0, 7.3, 8.1]
[1995, 2003, 2018, 2014, 2014] [5.4, 8.0, 7.3, 8.1, 9.0]
[1995, 2003, 2018, 2014, 2014, 1959] [5.4, 8.0, 7.3, 8.1, 9.0, 6.4]
[1995, 2003, 2018, 2014, 2014, 1959, 1996] [5.4, 8.0, 7.3, 8.1, 9.0,
6.4, 7.7]
[1995, 2003, 2018, 2014, 2014, 1959, 1996, 2020] [5.4, 8.0, 7.3, 8.1,
9.0, 6.4, 7.7, 6.3]
[1995, 2003, 2018, 2014, 2014, 1959, 1996, 2020, 1960] [5.4, 8.0, 7.3,
8.1, 9.0, 6.4, 7.7, 6.3, 8.4]
[1995, 2003, 2018, 2014, 2014, 1959, 1996, 2020, 1960, 2010] [5.4,
8.0, 7.3, 8.1, 9.0, 6.4, 7.7, 6.3, 8.4, 7.6]
best fit line:
y = -13.67 + 0.01x
[1998] [4.2]
[1998, 2014] [4.2, 7.2]
[1998, 2014, 2007] [4.2, 7.2, 6.6]
[1998, 2014, 2007, 1995] [4.2, 7.2, 6.6, 6.0]
[1998, 2014, 2007, 1995, 2014] [4.2, 7.2, 6.6, 6.0, 4.7]
[1998, 2014, 2007, 1995, 2014, 2008] [4.2, 7.2, 6.6, 6.0, 4.7, 6.5]
[1998, 2014, 2007, 1995, 2014, 2008, 1994] [4.2, 7.2, 6.6, 6.0, 4.7,
6.5, 8.7]
[1998, 2014, 2007, 1995, 2014, 2008, 1994, 2021] [4.2, 7.2, 6.6, 6.0,
```

4.7, 6.5, 8.7, 6.7]
[1998, 2014, 2007, 1995, 2014, 2008, 1994, 2021, 1986] [4.2, 7.2, 6.6, 6.0, 4.7, 6.5, 8.7, 6.7, 6.8]
[1998, 2014, 2007, 1995, 2014, 2008, 1994, 2021, 1986, 2014] [4.2, 7.2, 6.6, 6.0, 4.7, 6.5, 8.7, 6.7, 6.8, 6.0]
best fit line:
y = -11.80 + 0.01x
[2003] [8.0]
[2003, 2014] [8.0, 8.1]
[2003, 2014, 1996] [8.0, 8.1, 7.7]
[2003, 2014, 1996, 2010] [8.0, 8.1, 7.7, 7.6]
[2003, 2014, 1996, 2010, 2021] [8.0, 8.1, 7.7, 7.6, 5.2]
[2003, 2014, 1996, 2010, 2021, 1993] [8.0, 8.1, 7.7, 7.6, 5.2, 9.2]
[2003, 2014, 1996, 2010, 2021, 1993, 2000] [8.0, 8.1, 7.7, 7.6, 5.2, 9.2, 6.5]
[2003, 2014, 1996, 2010, 2021, 1993, 2000, 2010] [8.0, 8.1, 7.7, 7.6, 5.2, 9.2, 6.5, 6.7]
[2003, 2014, 1996, 2010, 2021, 1993, 2000, 2010, 2015] [8.0, 8.1, 7.7, 7.6, 5.2, 9.2, 6.5, 6.7, 7.2]
[2003, 2014, 1996, 2010, 2021, 1993, 2000, 2010, 2015, 2011] [8.0, 8.1, 7.7, 7.6, 5.2, 9.2, 6.5, 6.7, 7.2, 7.3]
best fit line:
y = -8.47 + 0.01x
[2013] [6.0]
[2013, 1994] [6.0, 5.2]
[2013, 1994, 2021] [6.0, 5.2, 6.4]
[2013, 1994, 2021, 2020] [6.0, 5.2, 6.4, 7.6]
[2013, 1994, 2021, 2020, 2007] [6.0, 5.2, 6.4, 7.6, 7.9]
[2013, 1994, 2021, 2020, 2007, 2006] [6.0, 5.2, 6.4, 7.6, 7.9, 6.9]
[2013, 1994, 2021, 2020, 2007, 2006, 1994] [6.0, 5.2, 6.4, 7.6, 7.9, 6.9, 7.1]
[2013, 1994, 2021, 2020, 2007, 2006, 1994, 2020] [6.0, 5.2, 6.4, 7.6, 7.9, 6.9, 7.1, 7.7]
[2013, 1994, 2021, 2020, 2007, 2006, 1994, 2020, 2012] [6.0, 5.2, 6.4, 7.6, 7.9, 6.9, 7.1, 7.7, 8.1]
[2013, 1994, 2021, 2020, 2007, 2006, 1994, 2020, 2012, 1957] [6.0, 5.2, 6.4, 7.6, 7.9, 6.9, 7.1, 7.7, 8.1, 6.5]
best fit line:
y = -14.26 + 0.01x
[1998] [4.2]
[1998, 2017] [4.2, 6.9]
[1998, 2017, 1967] [4.2, 6.9, 6.0]
[1998, 2017, 1967, 2016] [4.2, 6.9, 6.0, 7.0]
[1998, 2017, 1967, 2016, 2019] [4.2, 6.9, 6.0, 7.0, 9.2]
[1998, 2017, 1967, 2016, 2019, 2010] [4.2, 6.9, 6.0, 7.0, 9.2, 6.5]
[1998, 2017, 1967, 2016, 2019, 2010, 2014] [4.2, 6.9, 6.0, 7.0, 9.2, 6.5, 6.2]
[1998, 2017, 1967, 2016, 2019, 2010, 2014, 1998] [4.2, 6.9, 6.0, 7.0, 9.2, 6.5, 6.2, 4.3]
[1998, 2017, 1967, 2016, 2019, 2010, 2014, 1998, 2010] [4.2, 6.9, 6.0,

7.0, 9.2, 6.5, 6.2, 4.3, 7.4]
[1998, 2017, 1967, 2016, 2019, 2010, 2014, 1998, 2010, 2015] [4.2,
6.9, 6.0, 7.0, 9.2, 6.5, 6.2, 4.3, 7.4, 6.2]
best fit line:
y = -15.88 + 0.01x
[2005] [8.9]
[2005, 2018] [8.9, 6.6]
[2005, 2018, 2017] [8.9, 6.6, 6.9]
[2005, 2018, 2017, 1967] [8.9, 6.6, 6.9, 6.0]
[2005, 2018, 2017, 1967, 2013] [8.9, 6.6, 6.9, 6.0, 8.6]
[2005, 2018, 2017, 1967, 2013, 2021] [8.9, 6.6, 6.9, 6.0, 8.6, 9.1]
[2005, 2018, 2017, 1967, 2013, 2021, 2010] [8.9, 6.6, 6.9, 6.0, 8.6,
9.1, 5.0]
[2005, 2018, 2017, 1967, 2013, 2021, 2010, 2012] [8.9, 6.6, 6.9, 6.0,
8.6, 9.1, 5.0, 6.6]
[2005, 2018, 2017, 1967, 2013, 2021, 2010, 2012, 2008] [8.9, 6.6, 6.9,
6.0, 8.6, 9.1, 5.0, 6.6, 6.6]
[2005, 2018, 2017, 1967, 2013, 2021, 2010, 2012, 2008, 1974] [8.9,
6.6, 6.9, 6.0, 8.6, 9.1, 5.0, 6.6, 6.6, 8.0]
best fit line:
y = -17.17 + 0.01x
[2000] [7.3]
[2000, 2012] [7.3, 8.4]
[2000, 2012, 2022] [7.3, 8.4, 5.2]
[2000, 2012, 2022, 1989] [7.3, 8.4, 5.2, 7.2]
[2000, 2012, 2022, 1989, 2002] [7.3, 8.4, 5.2, 7.2, 6.9]
[2000, 2012, 2022, 1989, 2002, 1984] [7.3, 8.4, 5.2, 7.2, 6.9, 5.4]
[2000, 2012, 2022, 1989, 2002, 1984, 1996] [7.3, 8.4, 5.2, 7.2, 6.9,
5.4, 7.8]
[2000, 2012, 2022, 1989, 2002, 1984, 1996, 2018] [7.3, 8.4, 5.2, 7.2,
6.9, 5.4, 7.8, 7.2]
[2000, 2012, 2022, 1989, 2002, 1984, 1996, 2018, 2022] [7.3, 8.4, 5.2,
7.2, 6.9, 5.4, 7.8, 7.2, 7.5]
[2000, 2012, 2022, 1989, 2002, 1984, 1996, 2018, 2022, 2017] [7.3,
8.4, 5.2, 7.2, 6.9, 5.4, 7.8, 7.2, 7.5, 8.7]
best fit line:
y = -16.04 + 0.01x
[2021] [6.9]
[2021, 1993] [6.9, 7.1]
[2021, 1993, 2006] [6.9, 7.1, 7.1]
[2021, 1993, 2006, 2019] [6.9, 7.1, 7.1, 6.7]
[2021, 1993, 2006, 2019, 2021] [6.9, 7.1, 7.1, 6.7, 6.0]
[2021, 1993, 2006, 2019, 2021, 2009] [6.9, 7.1, 7.1, 6.7, 6.0, 5.8]
[2021, 1993, 2006, 2019, 2021, 2009, 2021] [6.9, 7.1, 7.1, 6.7, 6.0,
5.8, 5.4]
[2021, 1993, 2006, 2019, 2021, 2009, 2021, 2021] [6.9, 7.1, 7.1, 6.7,
6.0, 5.8, 5.4, 7.7]
[2021, 1993, 2006, 2019, 2021, 2009, 2021, 2021, 2016] [6.9, 7.1, 7.1,
6.7, 6.0, 5.8, 5.4, 7.7, 5.6]
[2021, 1993, 2006, 2019, 2021, 2009, 2021, 2021, 2016, 2021] [6.9,

7.1, 7.1, 6.7, 6.0, 5.8, 5.4, 7.7, 5.6, 7.7]
best fit line:
y = -23.33 + 0.02x
[1986] [6.9]
[1986, 1994] [6.9, 5.2]
[1986, 1994, 2021] [6.9, 5.2, 6.4]
[1986, 1994, 2021, 2004] [6.9, 5.2, 6.4, 7.5]
[1986, 1994, 2021, 2004, 2019] [6.9, 5.2, 6.4, 7.5, 7.9]
[1986, 1994, 2021, 2004, 2019, 1994] [6.9, 5.2, 6.4, 7.5, 7.9, 7.1]
[1986, 1994, 2021, 2004, 2019, 1994, 1989] [6.9, 5.2, 6.4, 7.5, 7.9,
7.1, 6.5]
[1986, 1994, 2021, 2004, 2019, 1994, 1989, 1999] [6.9, 5.2, 6.4, 7.5,
7.9, 7.1, 6.5, 7.4]
[1986, 1994, 2021, 2004, 2019, 1994, 1989, 1999, 1970] [6.9, 5.2, 6.4,
7.5, 7.9, 7.1, 6.5, 7.4, 7.2]
[1986, 1994, 2021, 2004, 2019, 1994, 1989, 1999, 1970, 2021] [6.9,
5.2, 6.4, 7.5, 7.9, 7.1, 6.5, 7.4, 7.2, 7.1]
best fit line:
y = -15.07 + 0.01x
[2008] [6.6]
[2008, 2004] [6.6, 8.3]
[2008, 2004, 2019] [6.6, 8.3, 6.3]
[2008, 2004, 2019, 2016] [6.6, 8.3, 6.3, 6.8]
[2008, 2004, 2019, 2016, 1969] [6.6, 8.3, 6.3, 6.8, 8.5]
[2008, 2004, 2019, 2016, 1969, 1935] [6.6, 8.3, 6.3, 6.8, 8.5, 5.8]
[2008, 2004, 2019, 2016, 1969, 1935, 2022] [6.6, 8.3, 6.3, 6.8, 8.5,
5.8, 9.1]
[2008, 2004, 2019, 2016, 1969, 1935, 2022, 1980] [6.6, 8.3, 6.3, 6.8,
8.5, 5.8, 9.1, 7.1]
[2008, 2004, 2019, 2016, 1969, 1935, 2022, 1980, 2004] [6.6, 8.3, 6.3,
6.8, 8.5, 5.8, 9.1, 7.1, 5.1]
[2008, 2004, 2019, 2016, 1969, 1935, 2022, 1980, 2004, 1985] [6.6,
8.3, 6.3, 6.8, 8.5, 5.8, 9.1, 7.1, 5.1, 6.3]
best fit line:
y = -19.32 + 0.01x
[1998] [4.2]
[1998, 2016] [4.2, 7.0]
[1998, 2016, 2021] [4.2, 7.0, 8.7]
[1998, 2016, 2021, 2016] [4.2, 7.0, 8.7, 7.6]
[1998, 2016, 2021, 2016, 1998] [4.2, 7.0, 8.7, 7.6, 4.3]
[1998, 2016, 2021, 2016, 1998, 2010] [4.2, 7.0, 8.7, 7.6, 4.3, 7.4]
[1998, 2016, 2021, 2016, 1998, 2010, 2013] [4.2, 7.0, 8.7, 7.6, 4.3,
7.4, 2.4]
[1998, 2016, 2021, 2016, 1998, 2010, 2013, 2019] [4.2, 7.0, 8.7, 7.6,
4.3, 7.4, 2.4, 7.8]
[1998, 2016, 2021, 2016, 1998, 2010, 2013, 2019, 2020] [4.2, 7.0, 8.7,
7.6, 4.3, 7.4, 2.4, 7.8, 2.8]
[1998, 2016, 2021, 2016, 1998, 2010, 2013, 2019, 2020, 2018] [4.2,
7.0, 8.7, 7.6, 4.3, 7.4, 2.4, 7.8, 2.8, 8.5]
best fit line:

y = -43.24 + 0.02x
[2014] [4.7]
[2014, 2010] [4.7, 6.5]
[2014, 2010, 2010] [4.7, 6.5, 8.4]
[2014, 2010, 2010, 1981] [4.7, 6.5, 8.4, 5.2]
[2014, 2010, 2010, 1981, 1964] [4.7, 6.5, 8.4, 5.2, 7.2]
[2014, 2010, 2010, 1981, 1964, 2007] [4.7, 6.5, 8.4, 5.2, 7.2, 8.2]
[2014, 2010, 2010, 1981, 1964, 2007, 1978] [4.7, 6.5, 8.4, 5.2, 7.2, 8.2, 6.1]
[2014, 2010, 2010, 1981, 1964, 2007, 1978, 1983] [4.7, 6.5, 8.4, 5.2, 7.2, 8.2, 6.1, 3.8]
[2014, 2010, 2010, 1981, 1964, 2007, 1978, 1983, 2017] [4.7, 6.5, 8.4, 5.2, 7.2, 8.2, 6.1, 3.8, 8.6]
[2014, 2010, 2010, 1981, 1964, 2007, 1978, 1983, 2017, 2011] [4.7, 6.5, 8.4, 5.2, 7.2, 8.2, 6.1, 3.8, 8.6, 7.8]
best fit line:
y = -14.25 + 0.01x
[2006] [7.5]
[2006, 2021] [7.5, 8.9]
[2006, 2021, 2014] [7.5, 8.9, 9.0]
[2006, 2021, 2014, 1982] [7.5, 8.9, 9.0, 5.8]
[2006, 2021, 2014, 1982, 2016] [7.5, 8.9, 9.0, 5.8, 8.0]
[2006, 2021, 2014, 1982, 2016, 2009] [7.5, 8.9, 9.0, 5.8, 8.0, 2.4]
[2006, 2021, 2014, 1982, 2016, 2009, 2015] [7.5, 8.9, 9.0, 5.8, 8.0, 2.4, 3.1]
[2006, 2021, 2014, 1982, 2016, 2009, 2015, 1964] [7.5, 8.9, 9.0, 5.8, 8.0, 2.4, 3.1, 6.4]
[2006, 2021, 2014, 1982, 2016, 2009, 2015, 1964, 2017] [7.5, 8.9, 9.0, 5.8, 8.0, 2.4, 3.1, 6.4, 7.5]
[2006, 2021, 2014, 1982, 2016, 2009, 2015, 1964, 2017, 1991] [7.5, 8.9, 9.0, 5.8, 8.0, 2.4, 3.1, 6.4, 7.5, 6.8]
best fit line:
y = -22.31 + 0.01x
[1995] [5.4]
[1995, 2003] [5.4, 3.8]
[1995, 2003, 2007] [5.4, 3.8, 5.5]
[1995, 2003, 2007, 2016] [5.4, 3.8, 5.5, 6.6]
[1995, 2003, 2007, 2016, 2021] [5.4, 3.8, 5.5, 6.6, 5.2]
[1995, 2003, 2007, 2016, 2021, 1975] [5.4, 3.8, 5.5, 6.6, 5.2, 6.4]
[1995, 2003, 2007, 2016, 2021, 1975, 2010] [5.4, 3.8, 5.5, 6.6, 5.2, 6.4, 6.1]
[1995, 2003, 2007, 2016, 2021, 1975, 2010, 2011] [5.4, 3.8, 5.5, 6.6, 5.2, 6.4, 6.1, 7.3]
[1995, 2003, 2007, 2016, 2021, 1975, 2010, 2011, 1996] [5.4, 3.8, 5.5, 6.6, 5.2, 6.4, 6.1, 7.3, 6.0]
[1995, 2003, 2007, 2016, 2021, 1975, 2010, 2011, 1996, 2017] [5.4, 3.8, 5.5, 6.6, 5.2, 6.4, 6.1, 7.3, 6.0, 4.3]
best fit line:
y = -16.63 + 0.01x
[2019] [5.2]

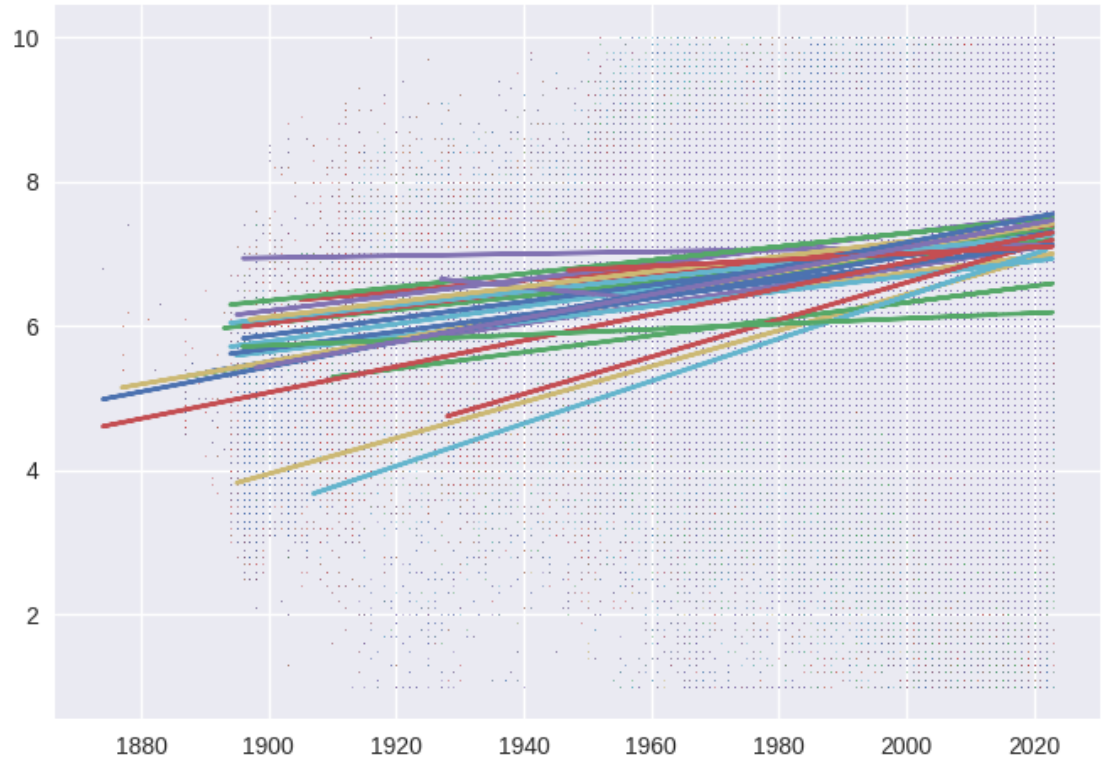
[2019, 2015] [5.2, 7.5]
[2019, 2015, 2016] [5.2, 7.5, 5.8]
[2019, 2015, 2016, 2009] [5.2, 7.5, 5.8, 7.4]
[2019, 2015, 2016, 2009, 2020] [5.2, 7.5, 5.8, 7.4, 6.2]
[2019, 2015, 2016, 2009, 2020, 2019] [5.2, 7.5, 5.8, 7.4, 6.2, 9.0]
[2019, 2015, 2016, 2009, 2020, 2019, 2010] [5.2, 7.5, 5.8, 7.4, 6.2, 9.0, 5.0]
[2019, 2015, 2016, 2009, 2020, 2019, 2010, 2020] [5.2, 7.5, 5.8, 7.4, 6.2, 9.0, 5.0, 4.4]
[2019, 2015, 2016, 2009, 2020, 2019, 2010, 2020, 2017] [5.2, 7.5, 5.8, 7.4, 6.2, 9.0, 5.0, 4.4, 4.5]
[2019, 2015, 2016, 2009, 2020, 2019, 2010, 2020, 2017, 2015] [5.2, 7.5, 5.8, 7.4, 6.2, 9.0, 5.0, 4.4, 4.5, 7.2]
best fit line:
y = -44.80 + 0.03x
[2008] [6.5]
[2008, 2017] [6.5, 8.4]
[2008, 2017, 1998] [6.5, 8.4, 6.6]
[2008, 2017, 1998, 1986] [6.5, 8.4, 6.6, 6.9]
[2008, 2017, 1998, 1986, 2015] [6.5, 8.4, 6.6, 6.9, 7.5]
[2008, 2017, 1998, 1986, 2015, 1990] [6.5, 8.4, 6.6, 6.9, 7.5, 8.8]
[2008, 2017, 1998, 1986, 2015, 1990, 1991] [6.5, 8.4, 6.6, 6.9, 7.5, 8.8, 7.2]
[2008, 2017, 1998, 1986, 2015, 1990, 1991, 1939] [6.5, 8.4, 6.6, 6.9, 7.5, 8.8, 7.2, 6.3]
[2008, 2017, 1998, 1986, 2015, 1990, 1991, 1939, 2017] [6.5, 8.4, 6.6, 6.9, 7.5, 8.8, 7.2, 6.3, 7.5]
[2008, 2017, 1998, 1986, 2015, 1990, 1991, 1939, 2017, 2013] [6.5, 8.4, 6.6, 6.9, 7.5, 8.8, 7.2, 6.3, 7.5, 8.4]
best fit line:
y = 4.28 + 0.00x
[2006] [7.5]
[2006, 1986] [7.5, 6.8]
[2006, 1986, 1982] [7.5, 6.8, 5.8]
[2006, 1986, 1982, 1986] [7.5, 6.8, 5.8, 7.1]
[2006, 1986, 1982, 1986, 2016] [7.5, 6.8, 5.8, 7.1, 8.0]
[2006, 1986, 1982, 1986, 2016, 2017] [7.5, 6.8, 5.8, 7.1, 8.0, 7.5]
[2006, 1986, 1982, 1986, 2016, 2017, 2011] [7.5, 6.8, 5.8, 7.1, 8.0, 7.5, 7.7]
[2006, 1986, 1982, 1986, 2016, 2017, 2011, 1991] [7.5, 6.8, 5.8, 7.1, 8.0, 7.5, 7.7, 6.8]
[2006, 1986, 1982, 1986, 2016, 2017, 2011, 1991, 2017] [7.5, 6.8, 5.8, 7.1, 8.0, 7.5, 7.7, 6.8, 7.2]
[2006, 1986, 1982, 1986, 2016, 2017, 2011, 1991, 2017, 2002] [7.5, 6.8, 5.8, 7.1, 8.0, 7.5, 7.7, 6.8, 7.2, 5.9]
best fit line:
y = -24.08 + 0.02x
[2021] [7.0]
[2021, 1995] [7.0, 5.4]
[2021, 1995, 2013] [7.0, 5.4, 6.0]

[2021, 1995, 2013, 2021] [7.0, 5.4, 6.0, 6.9]
[2021, 1995, 2013, 2021, 2003] [7.0, 5.4, 6.0, 6.9, 8.0]
[2021, 1995, 2013, 2021, 2003, 1933] [7.0, 5.4, 6.0, 6.9, 8.0, 5.3]
[2021, 1995, 2013, 2021, 2003, 1933, 2006] [7.0, 5.4, 6.0, 6.9, 8.0, 5.3, 5.7]
[2021, 1995, 2013, 2021, 2003, 1933, 2006, 1957] [7.0, 5.4, 6.0, 6.9, 8.0, 5.3, 5.7, 6.6]
[2021, 1995, 2013, 2021, 2003, 1933, 2006, 1957, 2014] [7.0, 5.4, 6.0, 6.9, 8.0, 5.3, 5.7, 6.6, 8.1]
[2021, 1995, 2013, 2021, 2003, 1933, 2006, 1957, 2014, 2014] [7.0, 5.4, 6.0, 6.9, 8.0, 5.3, 5.7, 6.6, 8.1, 9.0]
best fit line:
y = -11.87 + 0.01x
[2022] [6.9]
[2022, 2007] [6.9, 5.5]
[2022, 2007, 2017] [6.9, 5.5, 8.4]
[2022, 2007, 2017, 1964] [6.9, 5.5, 8.4, 6.4]
[2022, 2007, 2017, 1964, 1995] [6.9, 5.5, 8.4, 6.4, 6.7]
[2022, 2007, 2017, 1964, 1995, 2007] [6.9, 5.5, 8.4, 6.4, 6.7, 8.6]
[2022, 2007, 2017, 1964, 1995, 2007, 2013] [6.9, 5.5, 8.4, 6.4, 6.7, 8.6, 5.7]
[2022, 2007, 2017, 1964, 1995, 2007, 2013, 2016] [6.9, 5.5, 8.4, 6.4, 6.7, 8.6, 5.7, 5.8]
[2022, 2007, 2017, 1964, 1995, 2007, 2013, 2016, 2010] [6.9, 5.5, 8.4, 6.4, 6.7, 8.6, 5.7, 5.8, 7.6]
[2022, 2007, 2017, 1964, 1995, 2007, 2013, 2016, 2010, 2007] [6.9, 5.5, 8.4, 6.4, 6.7, 8.6, 5.7, 5.8, 7.6, 7.9]
best fit line:
y = -16.56 + 0.01x
[1957] [6.6]
[1957, 1962] [6.6, 7.9]
[1957, 1962, 1955] [6.6, 7.9, 6.7]
[1957, 1962, 1955, 1953] [6.6, 7.9, 6.7, 7.7]
[1957, 1962, 1955, 1953, 1960] [6.6, 7.9, 6.7, 7.7, 6.5]
[1957, 1962, 1955, 1953, 1960, 1969] [6.6, 7.9, 6.7, 7.7, 6.5, 5.9]
[1957, 1962, 1955, 1953, 1960, 1969, 1932] [6.6, 7.9, 6.7, 7.7, 6.5, 5.9, 6.1]
[1957, 1962, 1955, 1953, 1960, 1969, 1932, 1968] [6.6, 7.9, 6.7, 7.7, 6.5, 5.9, 6.1, 6.9]
[1957, 1962, 1955, 1953, 1960, 1969, 1932, 1968, 1972] [6.6, 7.9, 6.7, 7.7, 6.5, 5.9, 6.1, 6.9, 8.1]
[1957, 1962, 1955, 1953, 1960, 1969, 1932, 1968, 1972, 1959] [6.6, 7.9, 6.7, 7.7, 6.5, 5.9, 6.1, 6.9, 8.1, 7.7]
best fit line:
y = -11.40 + 0.01x
[2013] [6.0]
[2013, 2007] [6.0, 6.6]
[2013, 2007, 1995] [6.0, 6.6, 6.0]
[2013, 2007, 1995, 2021] [6.0, 6.6, 6.0, 8.9]
[2013, 2007, 1995, 2021, 2008] [6.0, 6.6, 6.0, 8.9, 7.9]

[2013, 2007, 1995, 2021, 2008, 2021] [6.0, 6.6, 6.0, 8.9, 7.9, 6.7]
[2013, 2007, 1995, 2021, 2008, 2021, 2011] [6.0, 6.6, 6.0, 8.9, 7.9, 6.7, 7.7]
[2013, 2007, 1995, 2021, 2008, 2021, 2011, 1962] [6.0, 6.6, 6.0, 8.9, 7.9, 6.7, 7.7, 7.3]
[2013, 2007, 1995, 2021, 2008, 2021, 2011, 1962, 1976] [6.0, 6.6, 6.0, 8.9, 7.9, 6.7, 7.7, 7.3, 9.8]
[2013, 2007, 1995, 2021, 2008, 2021, 2011, 1962, 1976, 2009] [6.0, 6.6, 6.0, 8.9, 7.9, 6.7, 7.7, 7.3, 9.8, 9.3]
best fit line:
y = -29.18 + 0.02x
[1951] [7.2]
[1951, 1950] [7.2, 6.5]
[1951, 1950, 1949] [7.2, 6.5, 7.1]
[1951, 1950, 1949, 1931] [7.2, 6.5, 7.1, 6.8]
[1951, 1950, 1949, 1931, 1942] [7.2, 6.5, 7.1, 6.8, 6.1]
[1951, 1950, 1949, 1931, 1942, 1946] [7.2, 6.5, 7.1, 6.8, 6.1, 7.4]
[1951, 1950, 1949, 1931, 1942, 1946, 1945] [7.2, 6.5, 7.1, 6.8, 6.1, 7.4, 6.3]
[1951, 1950, 1949, 1931, 1942, 1946, 1945, 1949] [7.2, 6.5, 7.1, 6.8, 6.1, 7.4, 6.3, 6.6]
[1951, 1950, 1949, 1931, 1942, 1946, 1945, 1949, 1947] [7.2, 6.5, 7.1, 6.8, 6.1, 7.4, 6.3, 6.6, 6.4]
[1951, 1950, 1949, 1931, 1942, 1946, 1945, 1949, 1947, 1928] [7.2, 6.5, 7.1, 6.8, 6.1, 7.4, 6.3, 6.6, 6.4, 7.5]
best fit line:
y = 23.14 + -0.01x
[2017] [6.9]
[2017, 1964] [6.9, 6.7]
[2017, 1964, 2003] [6.9, 6.7, 6.7]
[2017, 1964, 2003, 1977] [6.9, 6.7, 6.7, 6.0]
[2017, 1964, 2003, 1977, 1970] [6.9, 6.7, 6.7, 6.0, 7.6]
[2017, 1964, 2003, 1977, 1970, 1970] [6.9, 6.7, 6.7, 6.0, 7.6, 5.6]
[2017, 1964, 2003, 1977, 1970, 1970, 1973] [6.9, 6.7, 6.7, 6.0, 7.6, 5.6, 7.6]
[2017, 1964, 2003, 1977, 1970, 1970, 1973, 1995] [6.9, 6.7, 6.7, 6.0, 7.6, 5.6, 7.6, 8.0]
[2017, 1964, 2003, 1977, 1970, 1970, 1973, 1995, 2016] [6.9, 6.7, 6.7, 6.0, 7.6, 5.6, 7.6, 8.0, 7.9]
[2017, 1964, 2003, 1977, 1970, 1970, 1973, 1995, 2016, 2007] [6.9, 6.7, 6.7, 6.0, 7.6, 5.6, 7.6, 8.0, 7.9, 3.5]
best fit line:
y = -13.80 + 0.01x
[1989] [6.2]
[1989, 1999] [6.2, 7.1]
[1989, 1999, 2013] [6.2, 7.1, 7.4]
[1989, 1999, 2013, 2010] [6.2, 7.1, 7.4, 7.7]
[1989, 1999, 2013, 2010, 1996] [6.2, 7.1, 7.4, 7.7, 8.0]
[1989, 1999, 2013, 2010, 1996, 1983] [6.2, 7.1, 7.4, 7.7, 8.0, 7.2]
[1989, 1999, 2013, 2010, 1996, 1983, 2012] [6.2, 7.1, 7.4, 7.7, 8.0,

7.2, 6.0]
[1989, 1999, 2013, 2010, 1996, 1983, 2012, 1982] [6.2, 7.1, 7.4, 7.7, 8.0, 7.2, 6.0, 5.6]
[1989, 1999, 2013, 2010, 1996, 1983, 2012, 1982, 2015] [6.2, 7.1, 7.4, 7.7, 8.0, 7.2, 6.0, 5.6, 4.3]
[1989, 1999, 2013, 2010, 1996, 1983, 2012, 1982, 2015, 1988] [6.2, 7.1, 7.4, 7.7, 8.0, 7.2, 6.0, 5.6, 4.3, 5.9]
best fit line:
y = -52.38 + 0.03x
[2013] [6.5]
[2013, 2001] [6.5, 7.3]
[2013, 2001, 2013] [6.5, 7.3, 7.0]
[2013, 2001, 2013, 2018] [6.5, 7.3, 7.0, 7.8]
[2013, 2001, 2013, 2018, 2018] [6.5, 7.3, 7.0, 7.8, 6.6]
[2013, 2001, 2013, 2018, 2018, 2009] [6.5, 7.3, 7.0, 7.8, 6.6, 6.2]
[2013, 2001, 2013, 2018, 2018, 2009, 2020] [6.5, 7.3, 7.0, 7.8, 6.6, 6.2, 6.3]
[2013, 2001, 2013, 2018, 2018, 2009, 2020, 1994] [6.5, 7.3, 7.0, 7.8, 6.6, 6.2, 6.3, 5.2]
[2013, 2001, 2013, 2018, 2018, 2009, 2020, 1994, 2021] [6.5, 7.3, 7.0, 7.8, 6.6, 6.2, 6.3, 5.2, 6.4]
[2013, 2001, 2013, 2018, 2018, 2009, 2020, 1994, 2021, 2020] [6.5, 7.3, 7.0, 7.8, 6.6, 6.2, 6.3, 5.2, 6.4, 7.6]
best fit line:
y = -27.32 + 0.02x
[2014] [4.7]
[2014, 2008] [4.7, 7.9]
[2014, 2008, 2021] [4.7, 7.9, 6.7]
[2014, 2008, 2021, 2015] [4.7, 7.9, 6.7, 3.1]
[2014, 2008, 2021, 2015, 1976] [4.7, 7.9, 6.7, 3.1, 9.8]
[2014, 2008, 2021, 2015, 1976, 1991] [4.7, 7.9, 6.7, 3.1, 9.8, 4.7]
[2014, 2008, 2021, 2015, 1976, 1991, 2016] [4.7, 7.9, 6.7, 3.1, 9.8, 4.7, 7.3]
[2014, 2008, 2021, 2015, 1976, 1991, 2016, 1989] [4.7, 7.9, 6.7, 3.1, 9.8, 4.7, 7.3, 5.9]
[2014, 2008, 2021, 2015, 1976, 1991, 2016, 1989, 2015] [4.7, 7.9, 6.7, 3.1, 9.8, 4.7, 7.3, 5.9, 5.2]
[2014, 2008, 2021, 2015, 1976, 1991, 2016, 1989, 2015, 1989] [4.7, 7.9, 6.7, 3.1, 9.8, 4.7, 7.3, 5.9, 5.2, 7.2]
best fit line:
y = -1.38 + 0.00x
[2021] [4.7]
[2021, 2015] [4.7, 7.5]
[2021, 2015, 2013] [4.7, 7.5, 6.5]
[2021, 2015, 2013, 2016] [4.7, 7.5, 6.5, 5.8]
[2021, 2015, 2013, 2016, 2019] [4.7, 7.5, 6.5, 5.8, 9.0]
[2021, 2015, 2013, 2016, 2019, 2010] [4.7, 7.5, 6.5, 5.8, 9.0, 5.0]
[2021, 2015, 2013, 2016, 2019, 2010, 2018] [4.7, 7.5, 6.5, 5.8, 9.0, 5.0, 9.0]
[2021, 2015, 2013, 2016, 2019, 2010, 2018, 2005] [4.7, 7.5, 6.5, 5.8,

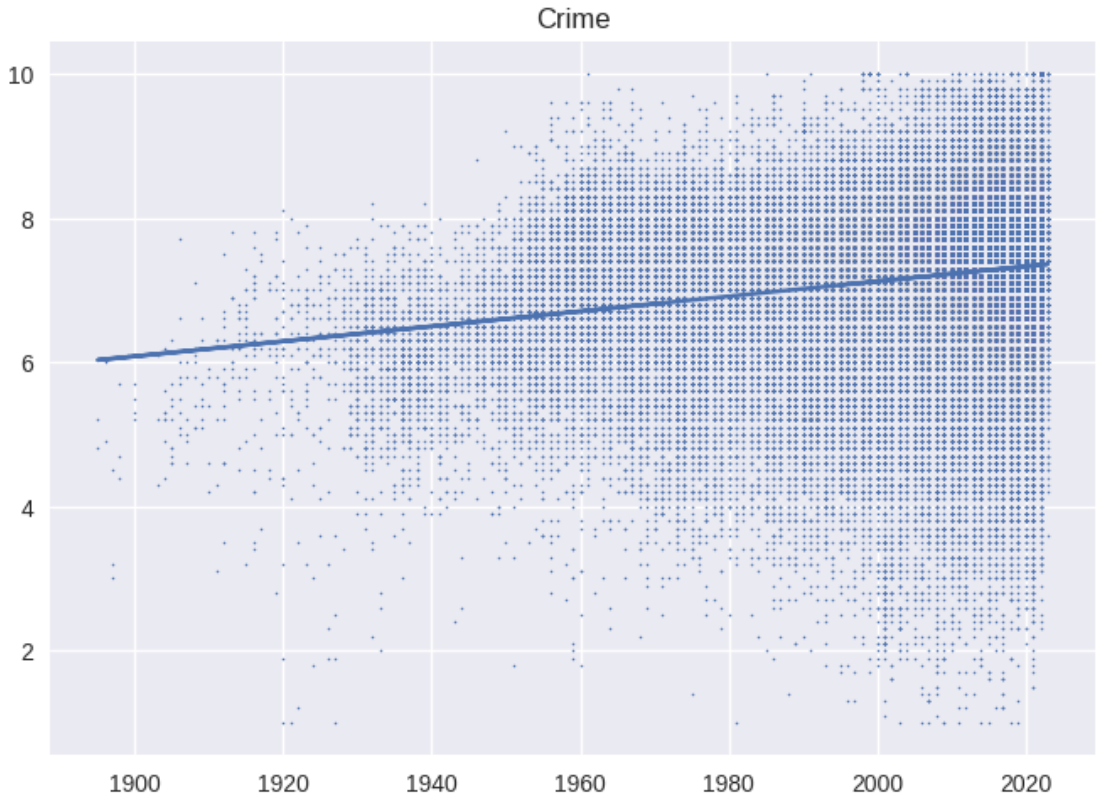
```
9.0, 5.0, 9.0, 4.8]
[2021, 2015, 2013, 2016, 2019, 2010, 2018, 2005, 2022] [4.7, 7.5, 6.5,
5.8, 9.0, 5.0, 9.0, 4.8, 7.1]
[2021, 2015, 2013, 2016, 2019, 2010, 2018, 2005, 2022, 2022] [4.7,
7.5, 6.5, 5.8, 9.0, 5.0, 9.0, 4.8, 7.1, 8.6]
best fit line:
y = -1.64 + 0.00x
[2006] [7.5]
[2006, 2021] [7.5, 8.9]
[2006, 2021, 1986] [7.5, 8.9, 6.8]
[2006, 2021, 1986, 1982] [7.5, 8.9, 6.8, 5.8]
[2006, 2021, 1986, 1982, 1986] [7.5, 8.9, 6.8, 5.8, 7.1]
[2006, 2021, 1986, 1982, 1986, 2016] [7.5, 8.9, 6.8, 5.8, 7.1, 8.0]
[2006, 2021, 1986, 1982, 1986, 2016, 1964] [7.5, 8.9, 6.8, 5.8, 7.1,
8.0, 6.4]
[2006, 2021, 1986, 1982, 1986, 2016, 1964, 2017] [7.5, 8.9, 6.8, 5.8,
7.1, 8.0, 6.4, 7.5]
[2006, 2021, 1986, 1982, 1986, 2016, 1964, 2017, 1991] [7.5, 8.9, 6.8,
5.8, 7.1, 8.0, 6.4, 7.5, 6.8]
[2006, 2021, 1986, 1982, 1986, 2016, 1964, 2017, 1991, 2017] [7.5,
8.9, 6.8, 5.8, 7.1, 8.0, 6.4, 7.5, 6.8, 7.2]
best fit line:
y = -25.62 + 0.02x
```



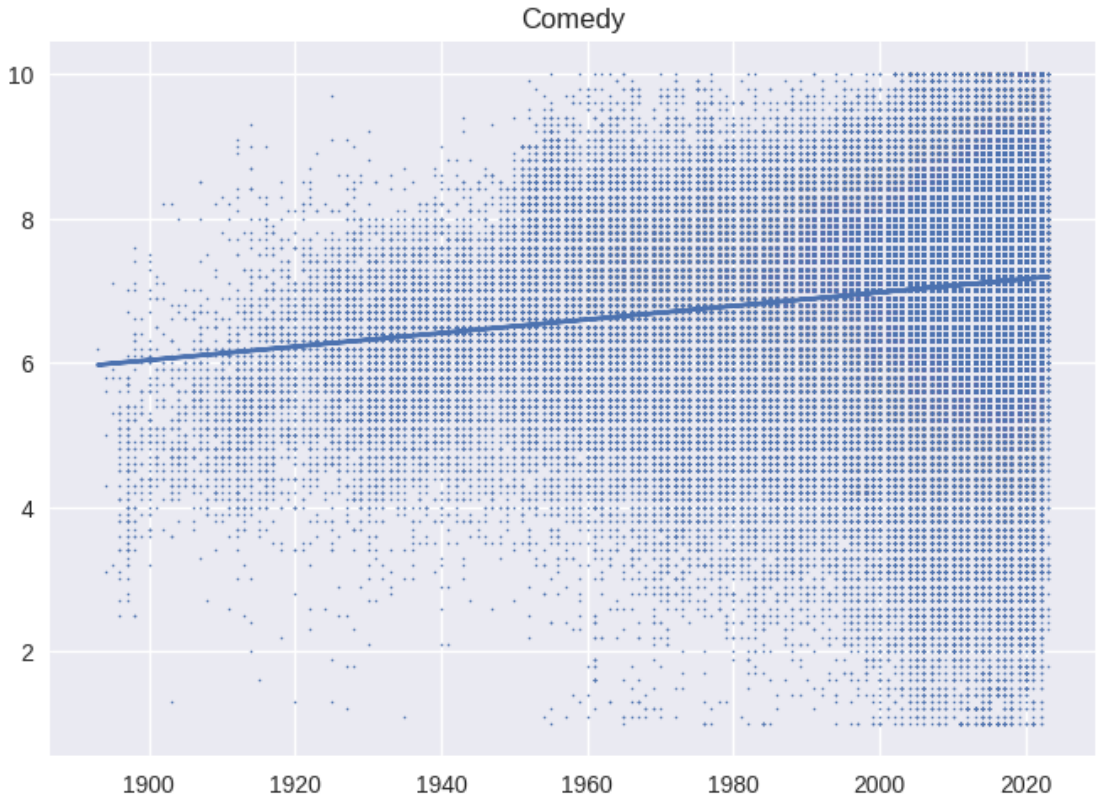

```python
import matplotlib.pyplot as plt
bestfits = {}
for genre in genre_split:
```

```
  year = []
  rating = []

  for index, row in genre_split[genre].iterrows():
    if row['startYear'] != '\\N':
      year.append(int(row['startYear']))
      rating.append(float(row['averageRating']))

  plt.title(genre)
  plt.scatter(year, rating, s=.6)

  a, b = best_fit(year, rating)
  bestfits[genre] = [a,b]
  yfit = [a + b * yeari for yeari in year]
  plt.plot(year, yfit)
  plt.show()
  plt.clf()
```

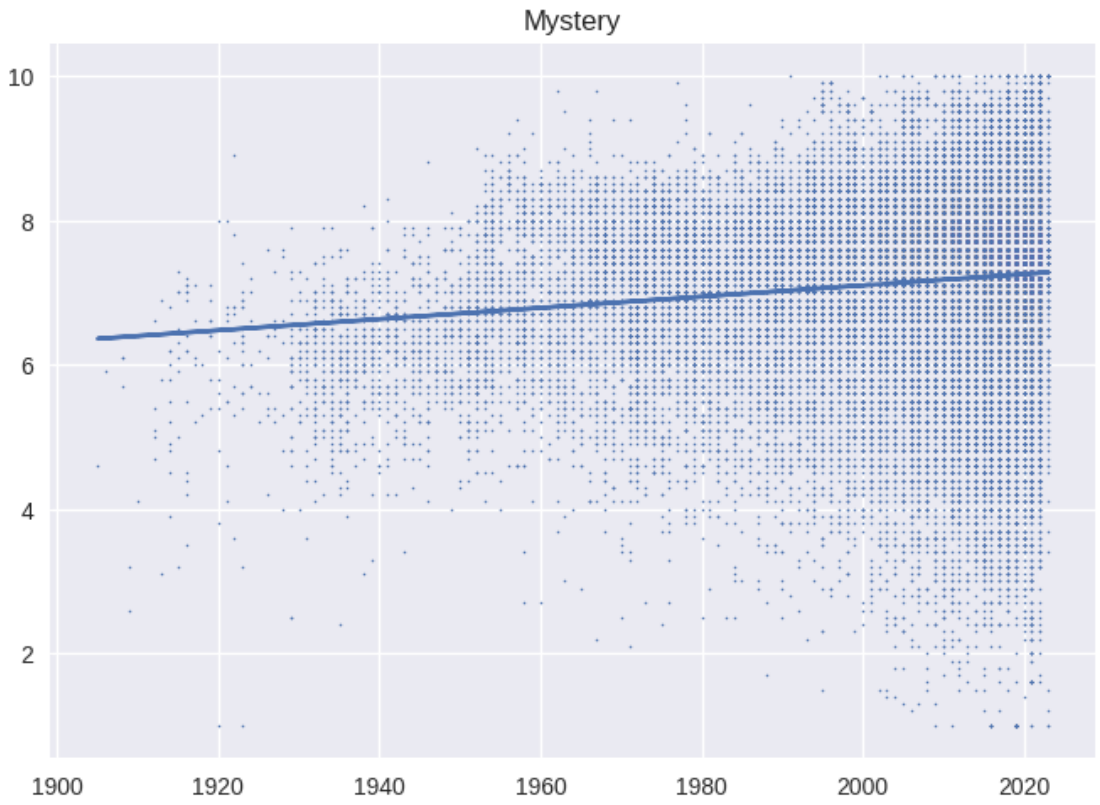
```
best fit line:
y = -13.67 + 0.01x
```



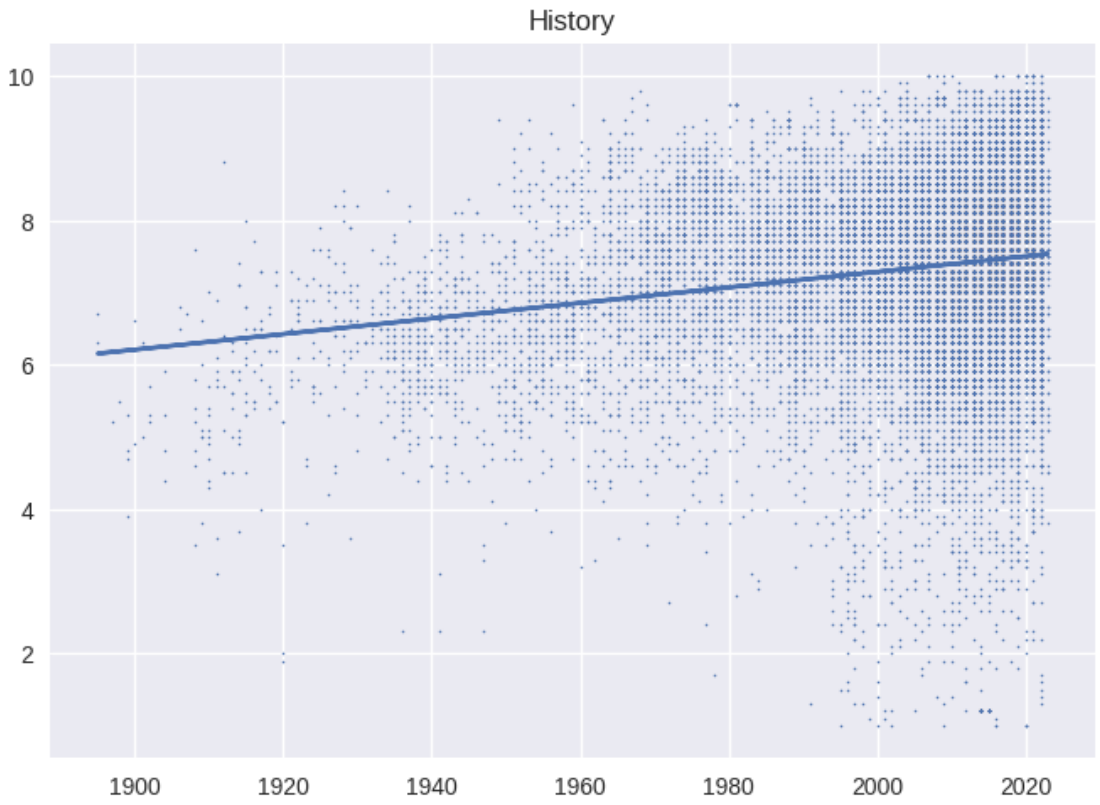

```
best fit line:
y = -11.80 + 0.01x
```



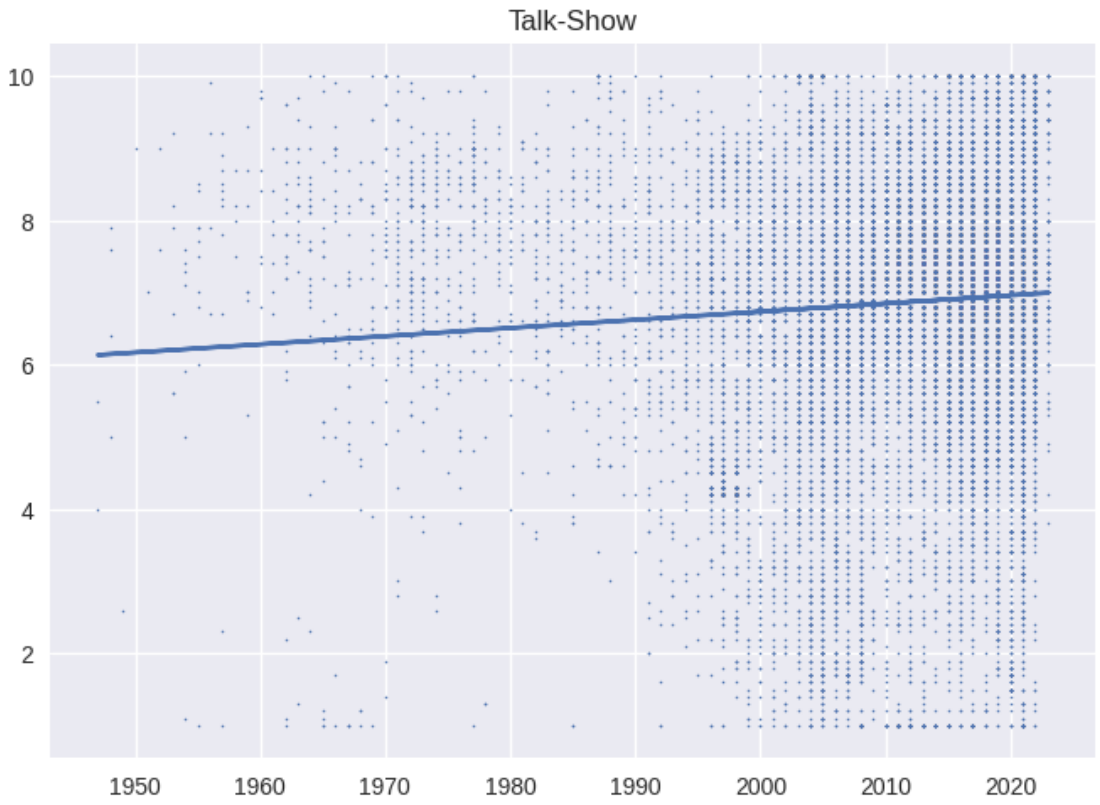
best fit line:
y = -8.47 + 0.01x



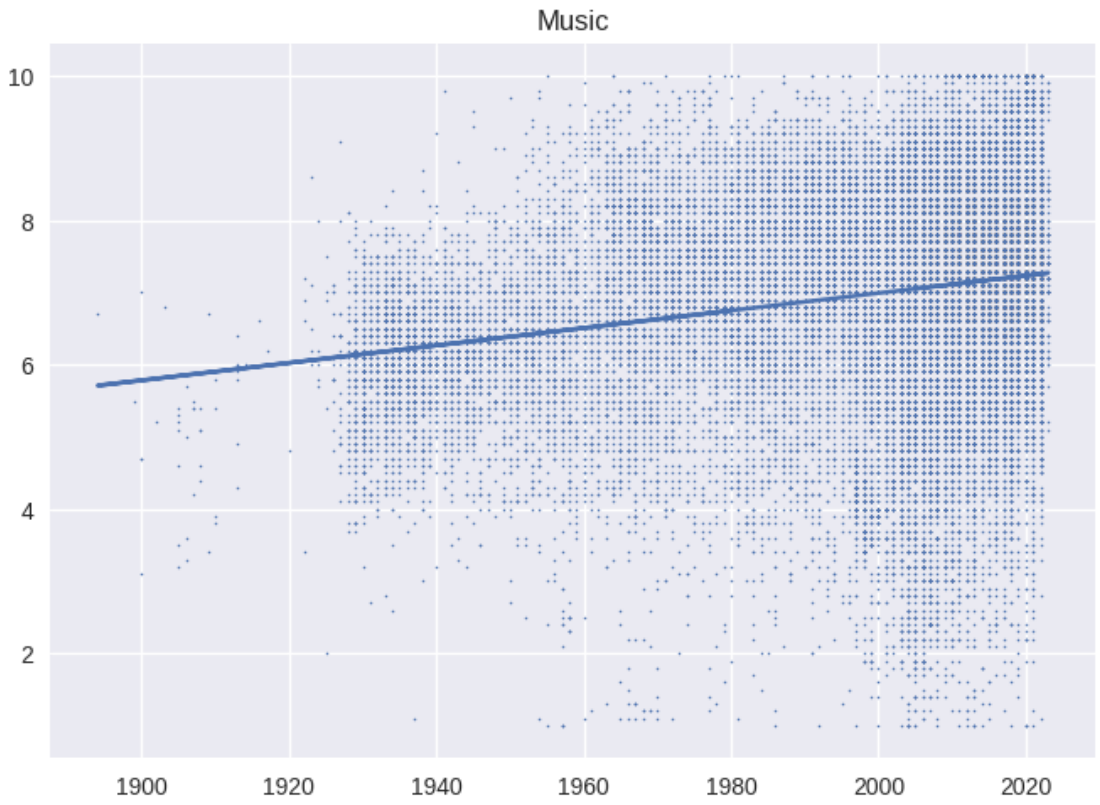
best fit line:
y = -14.26 + 0.01x



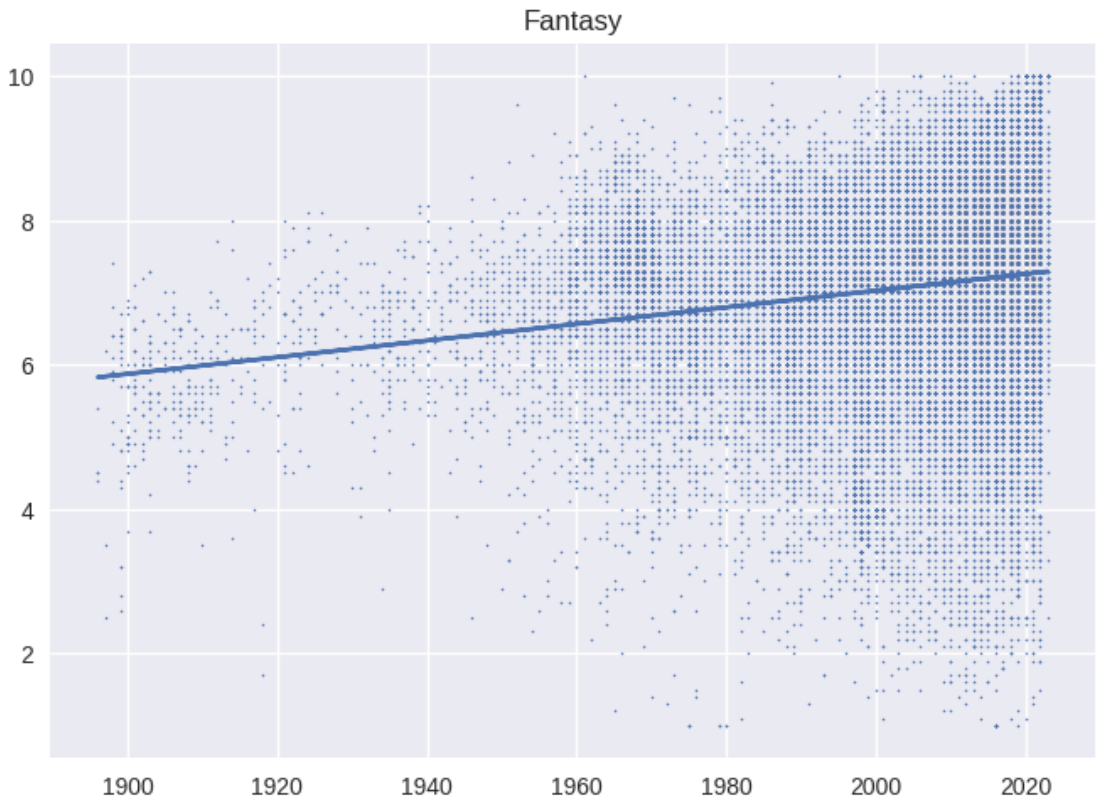

```
best fit line:
y = -15.88 + 0.01x
```



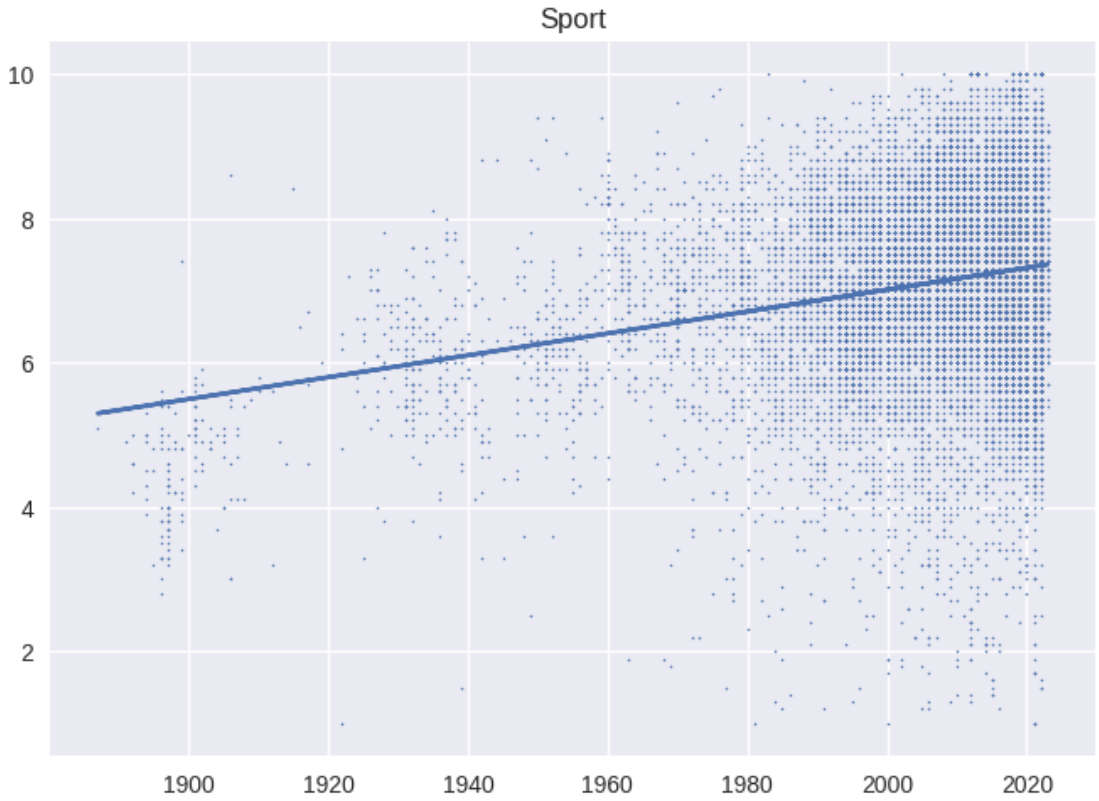
best fit line:
y = -17.17 + 0.01x



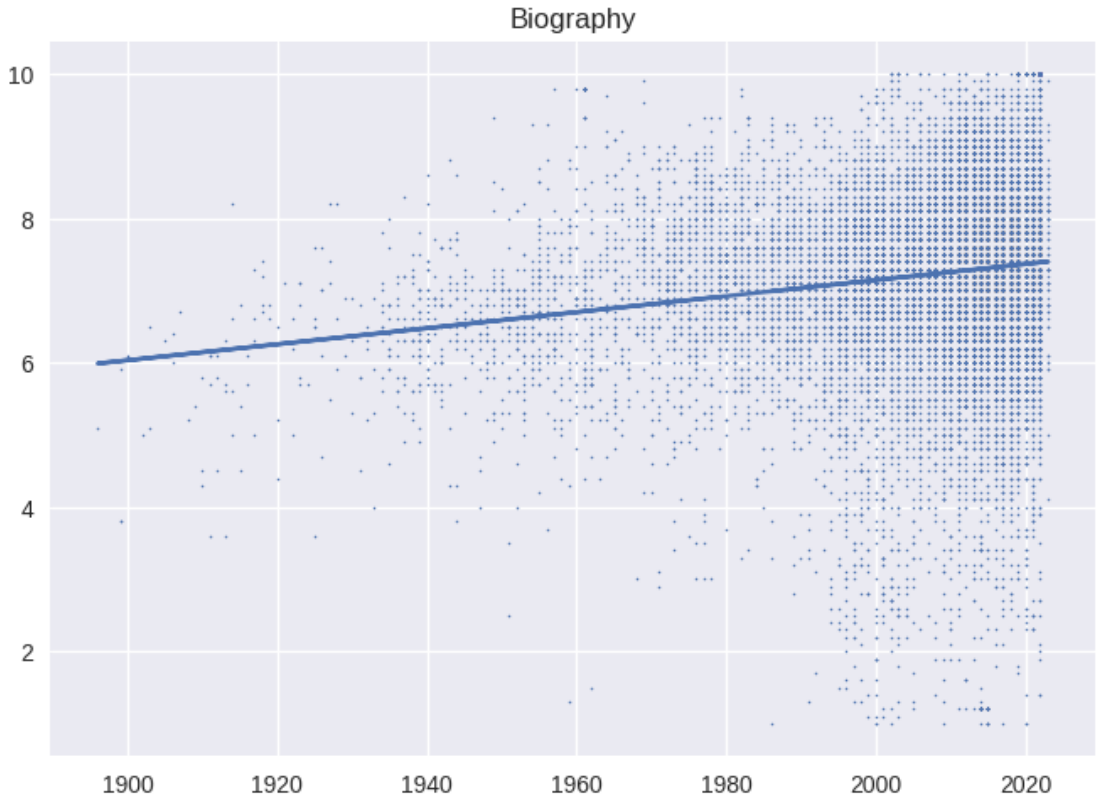
best fit line:
y = -16.04 + 0.01x



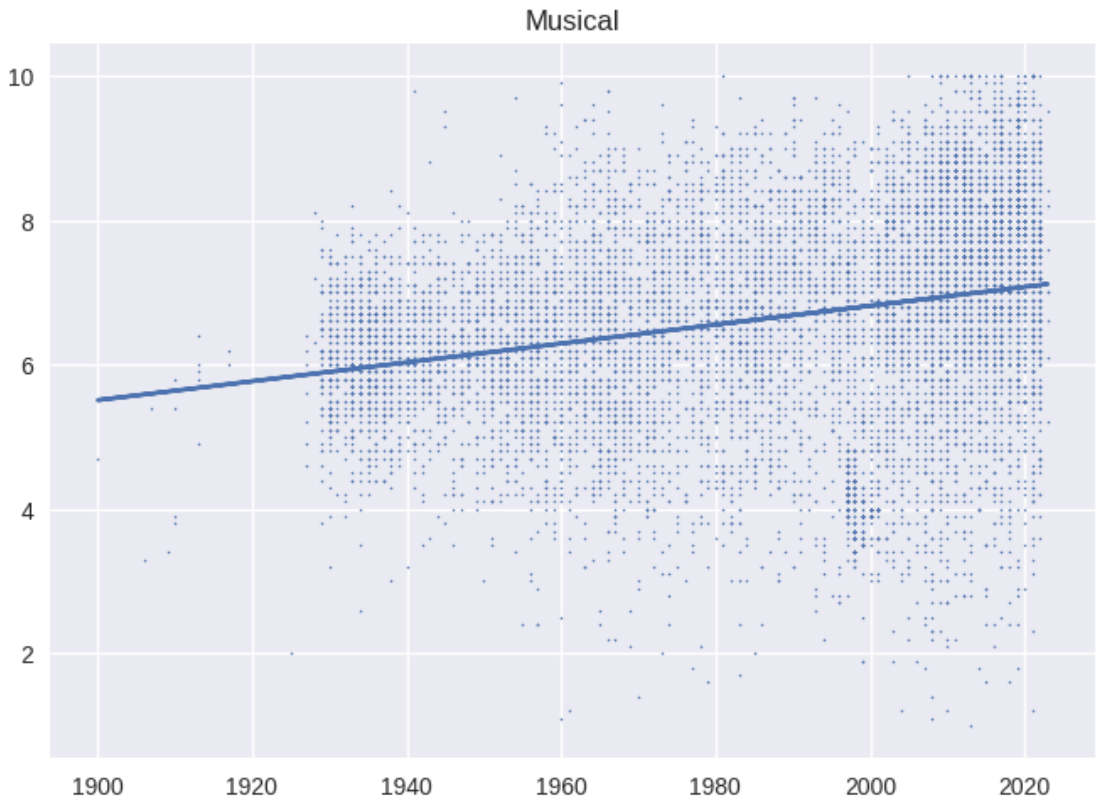
best fit line:
y = -23.33 + 0.02x



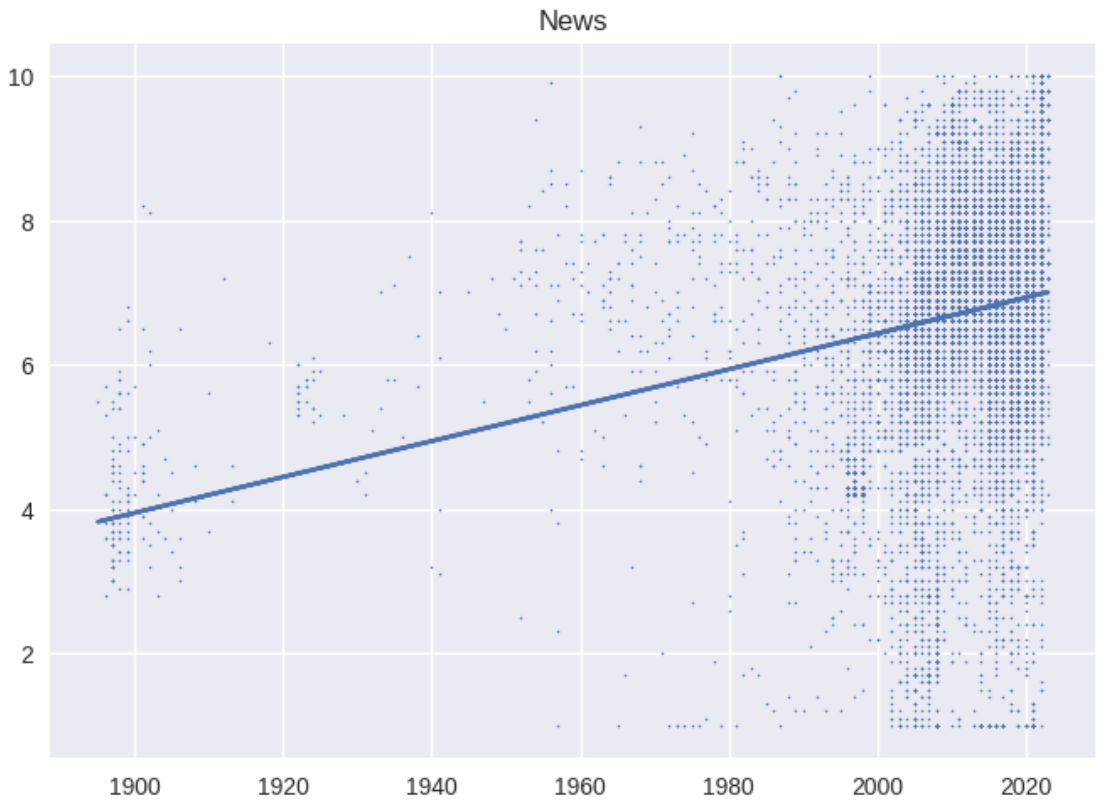
best fit line:
y = -15.07 + 0.01x



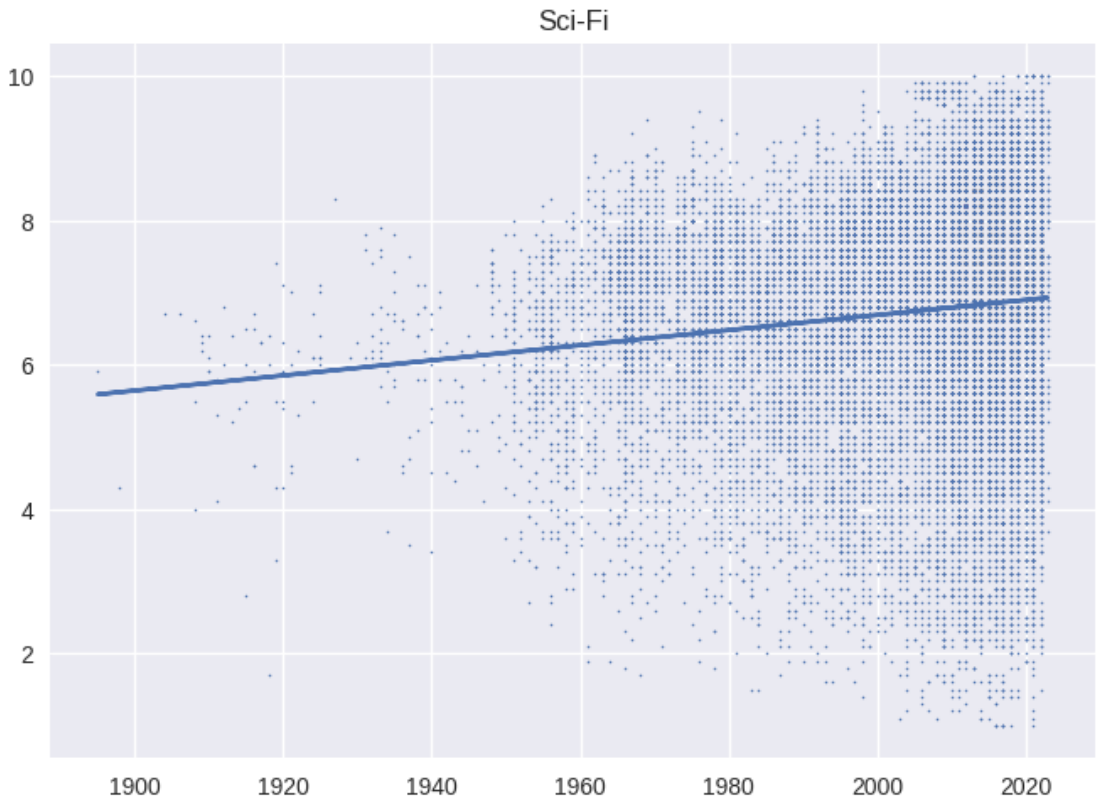
best fit line:
y = -19.32 + 0.01x



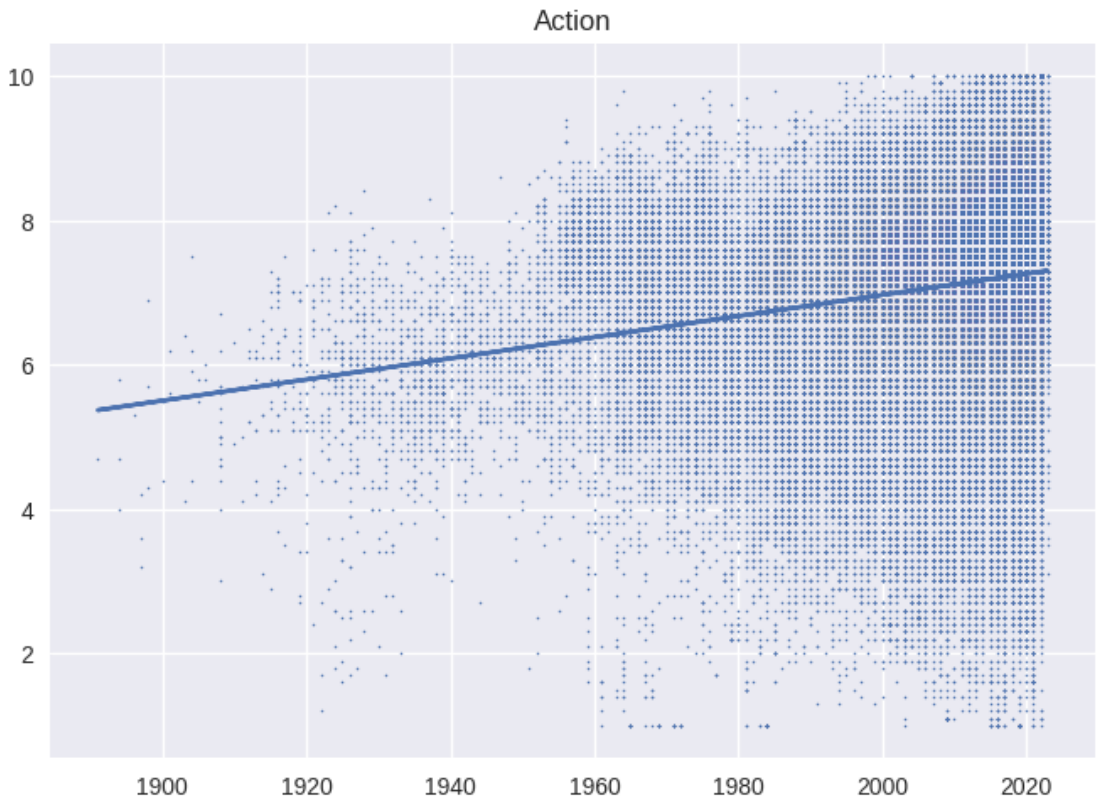
best fit line:
y = -43.24 + 0.02x



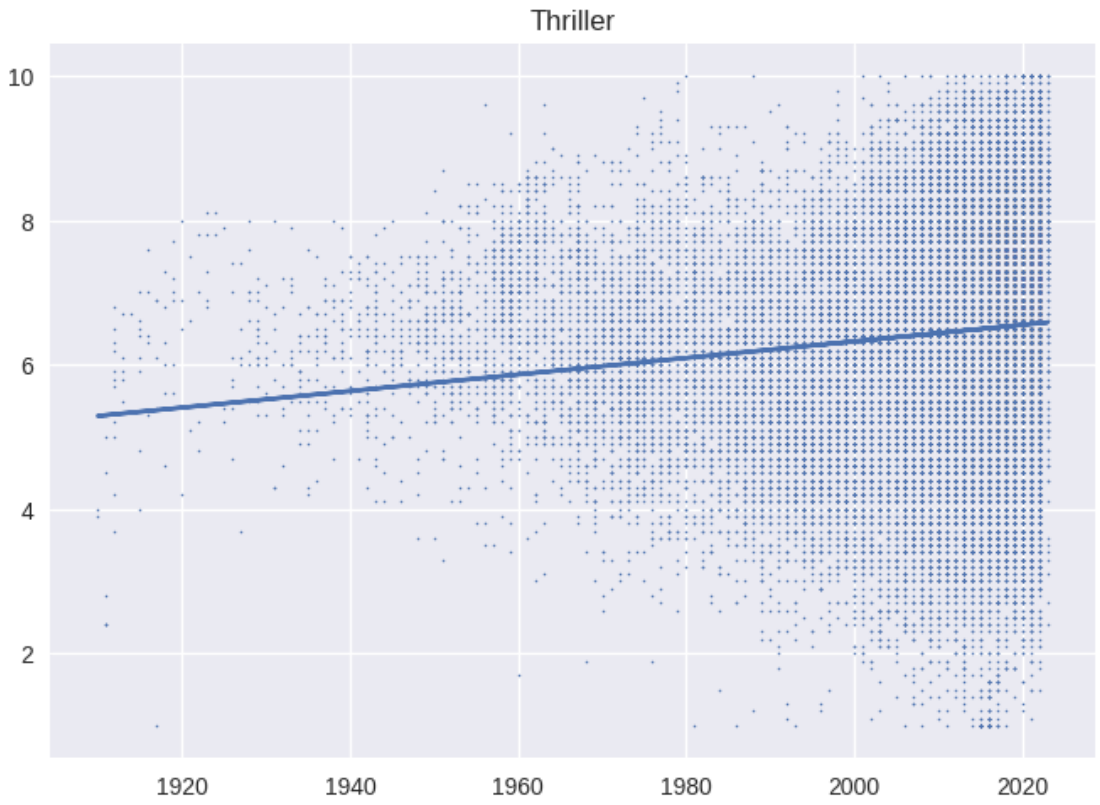

```
best fit line:
y = -14.25 + 0.01x
```



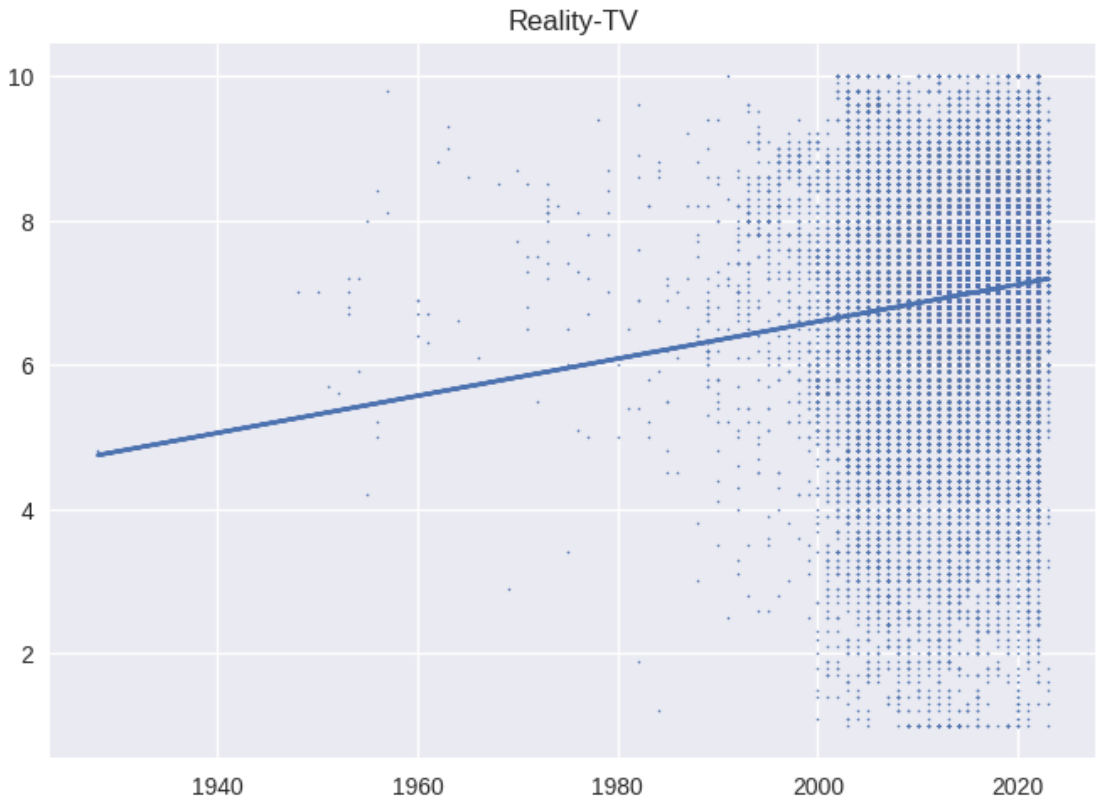
best fit line:
y = -22.31 + 0.01x



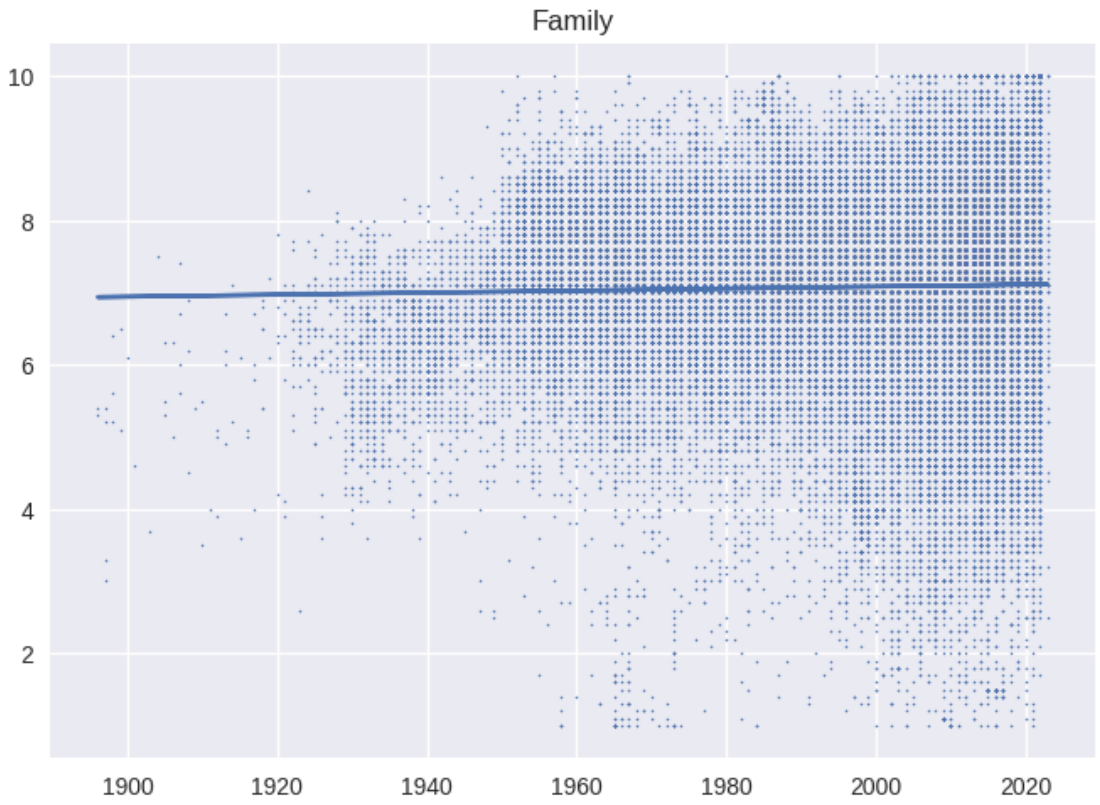
best fit line:
y = -16.63 + 0.01x



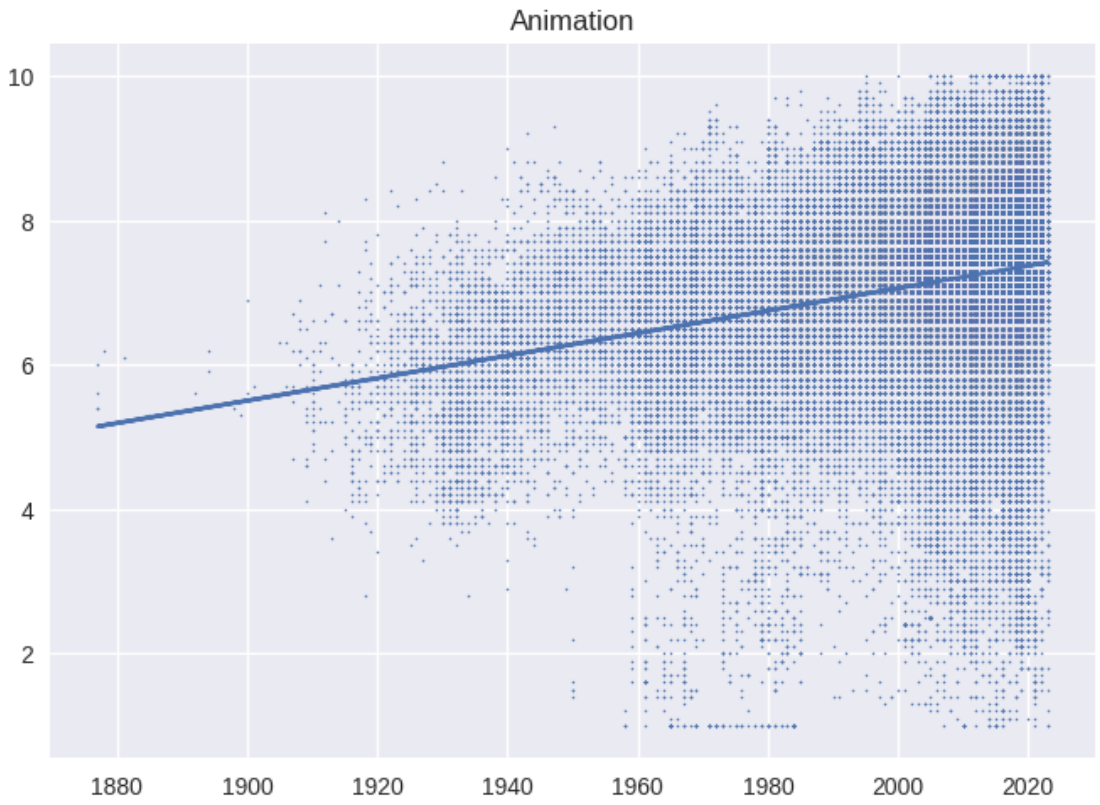

```
best fit line:
y = -44.80 + 0.03x
```



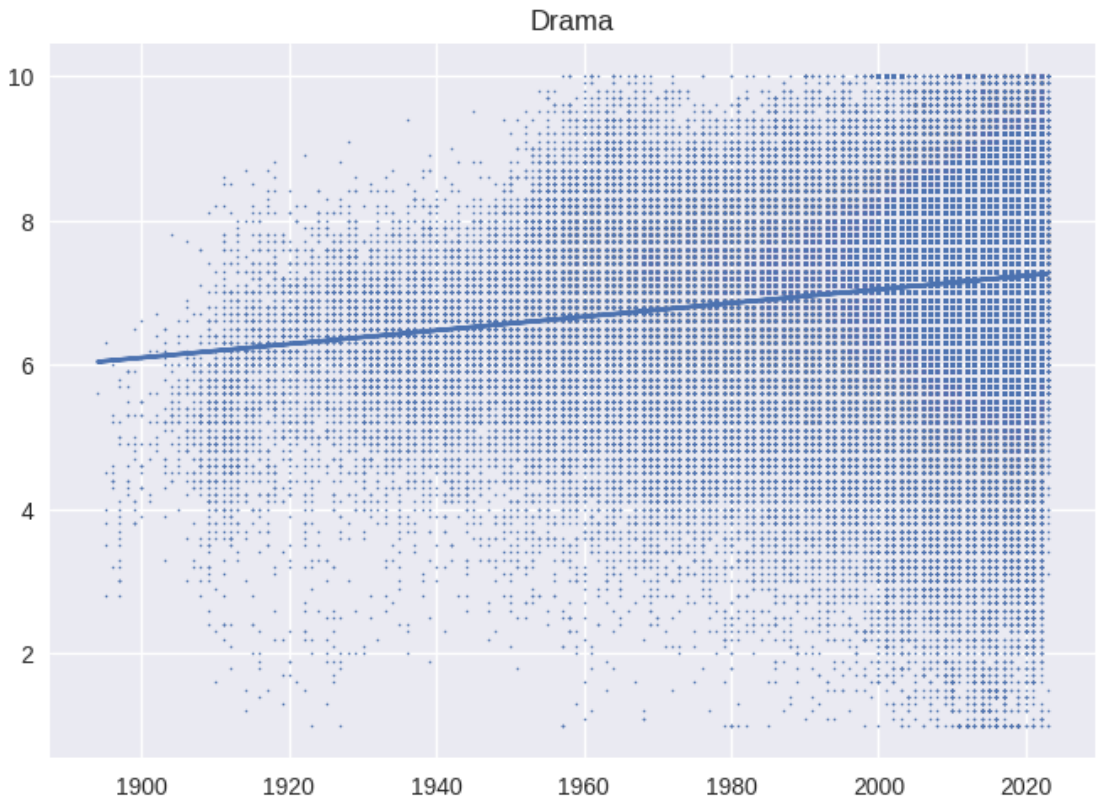

```
best fit line:
y = 4.28 + 0.00x
```



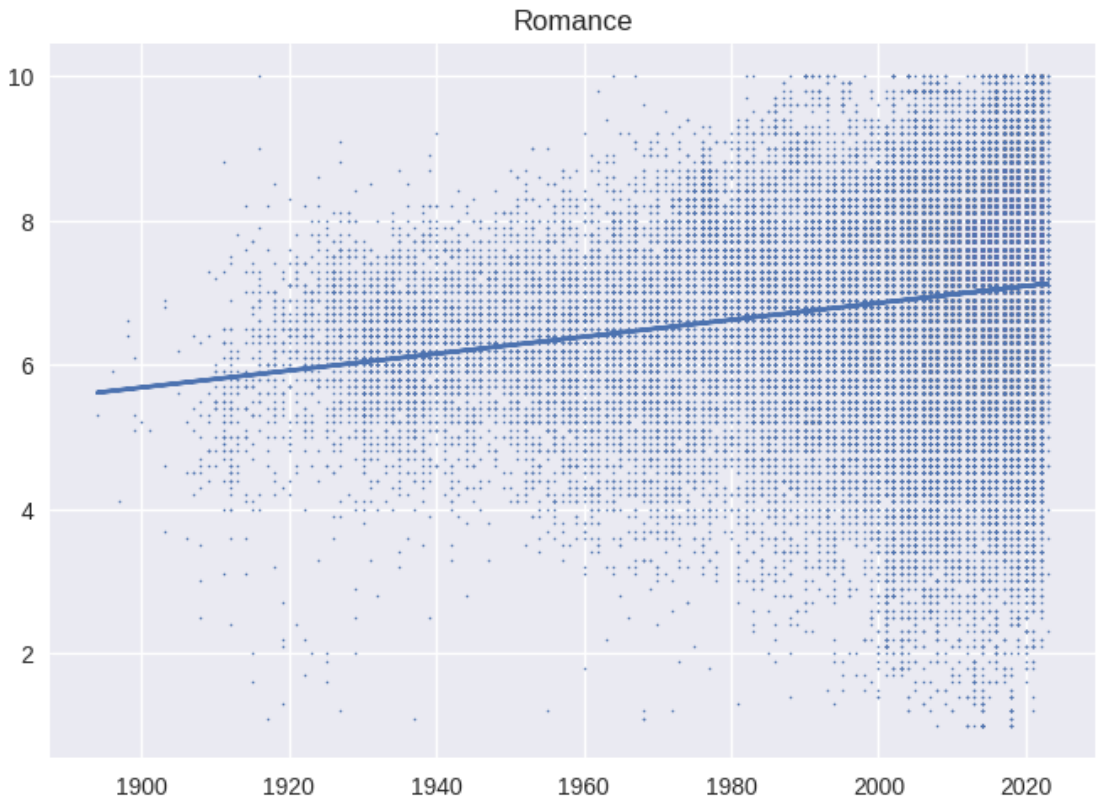
best fit line:
y = -24.08 + 0.02x



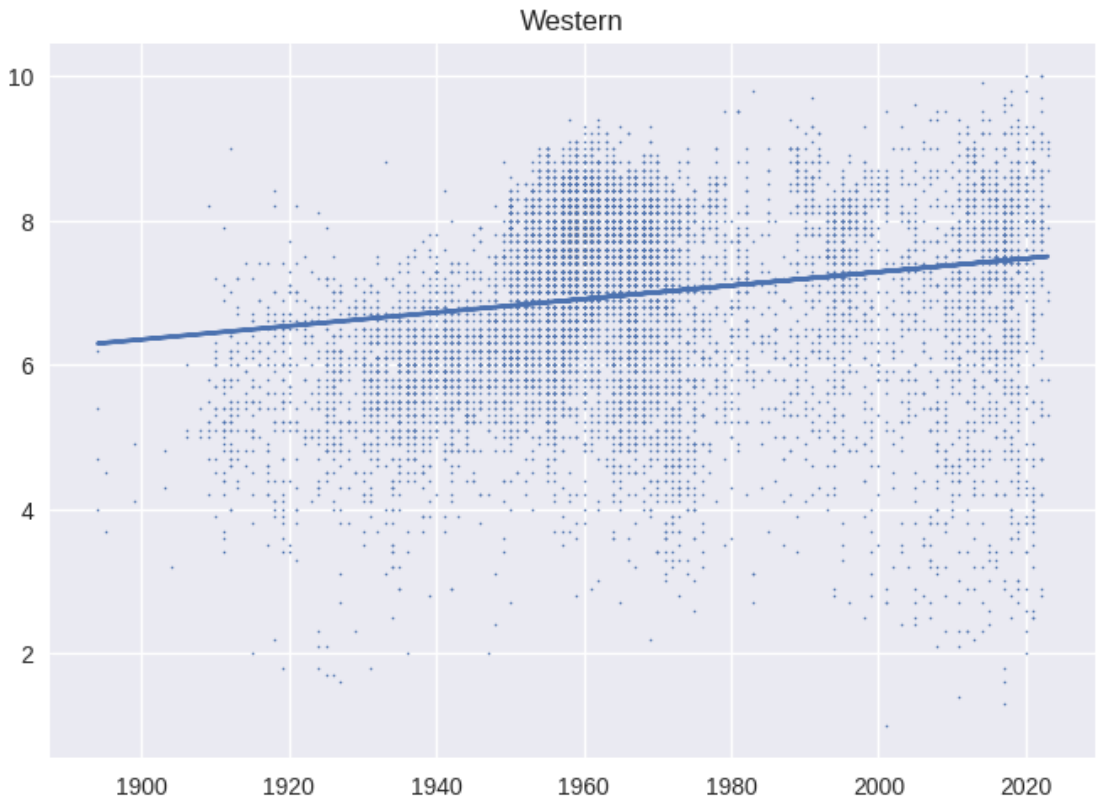

```
best fit line:
y = -11.87 + 0.01x
```



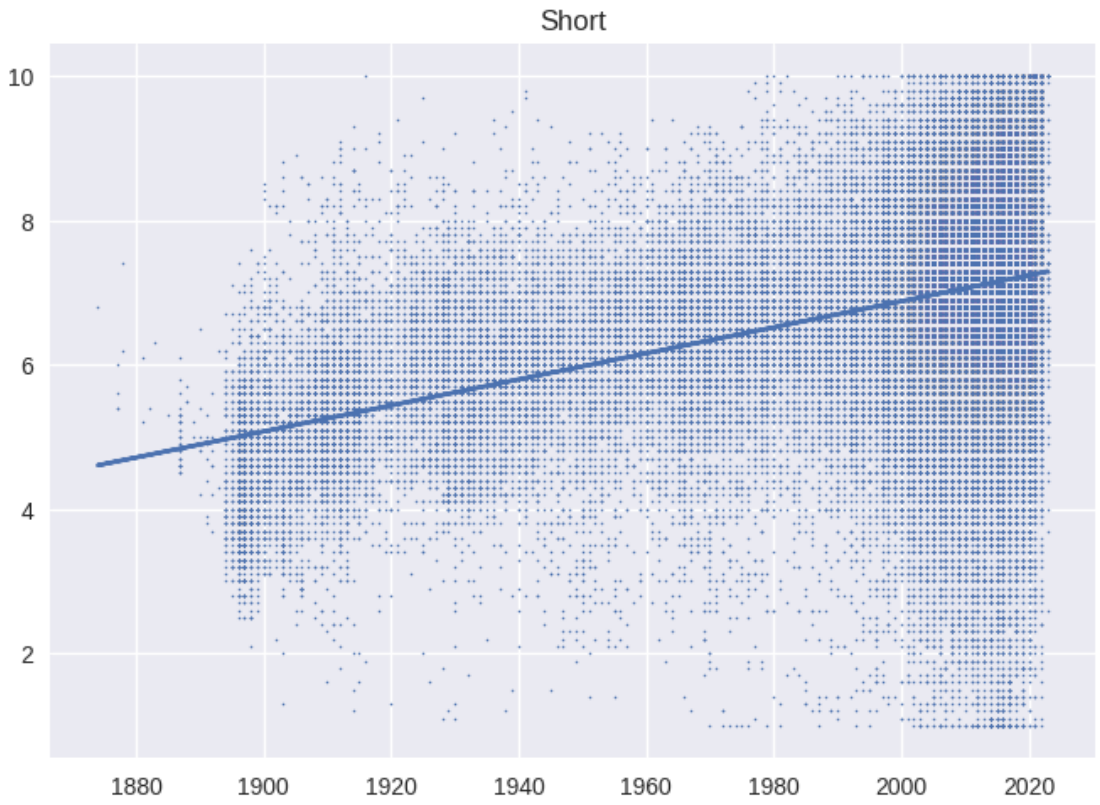
best fit line:
y = -16.56 + 0.01x



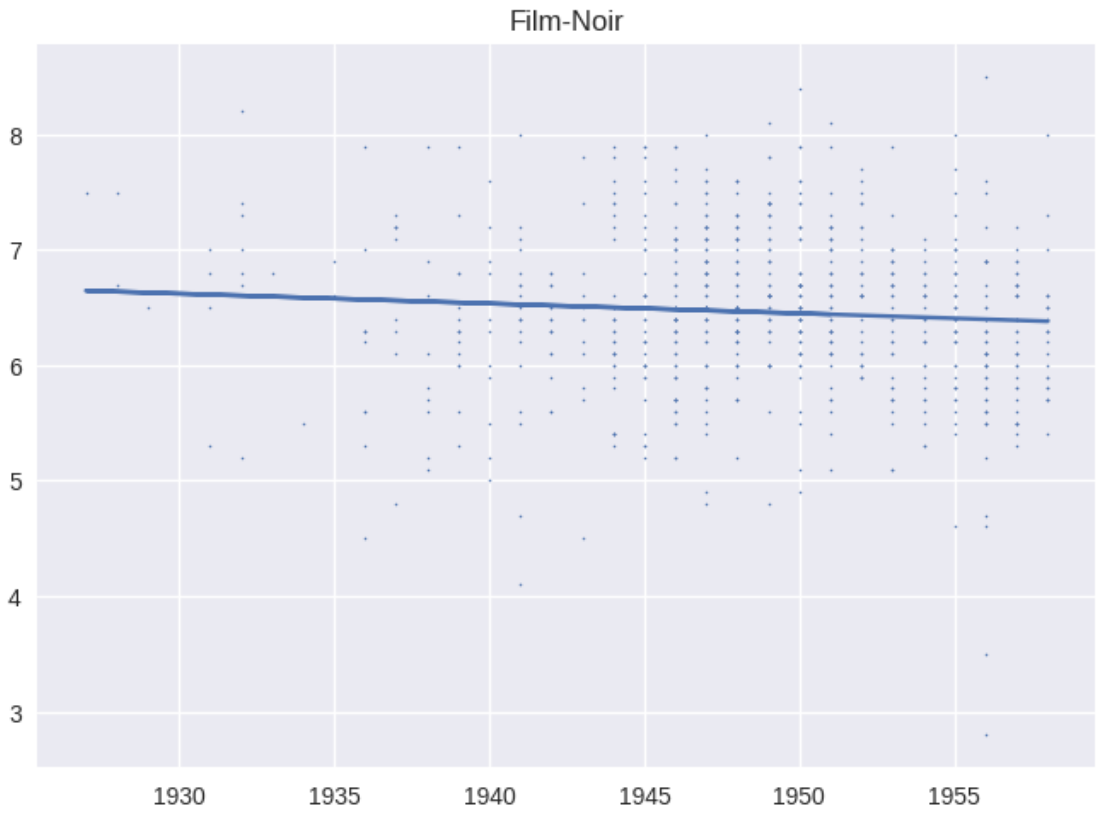

```
best fit line:
y = -11.40 + 0.01x
```



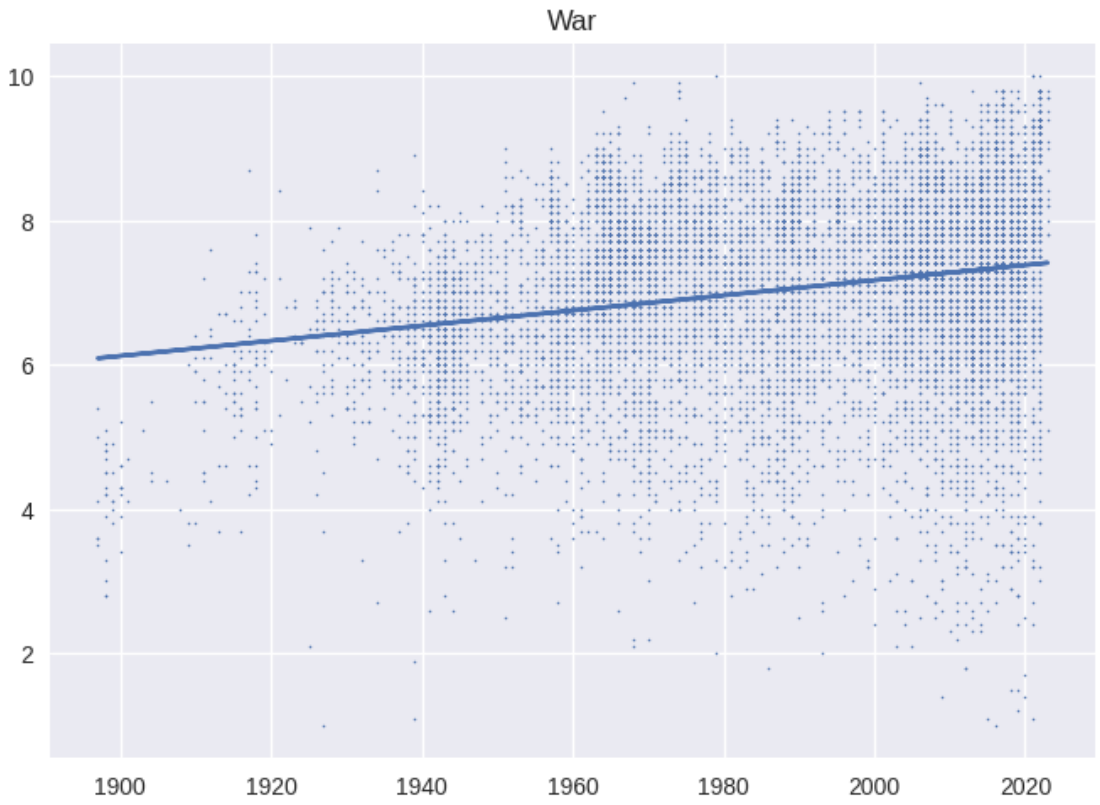
best fit line:
y = -29.18 + 0.02x



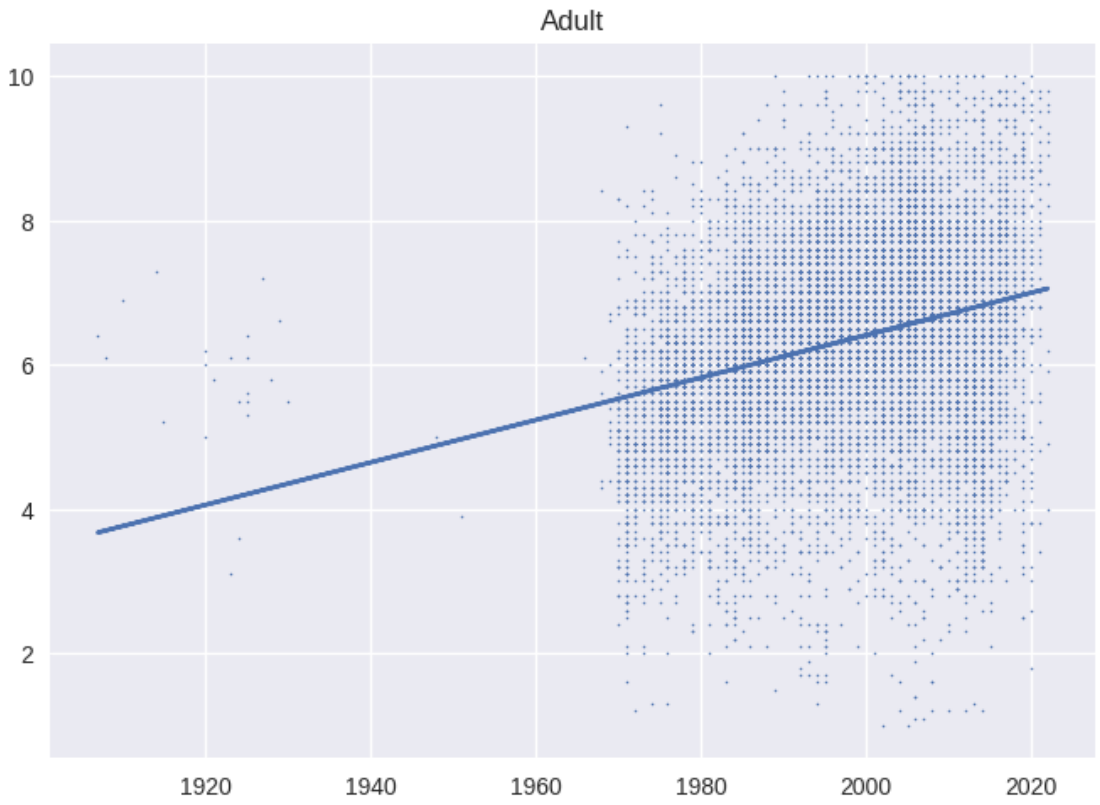
best fit line:
y = 23.14 + -0.01x



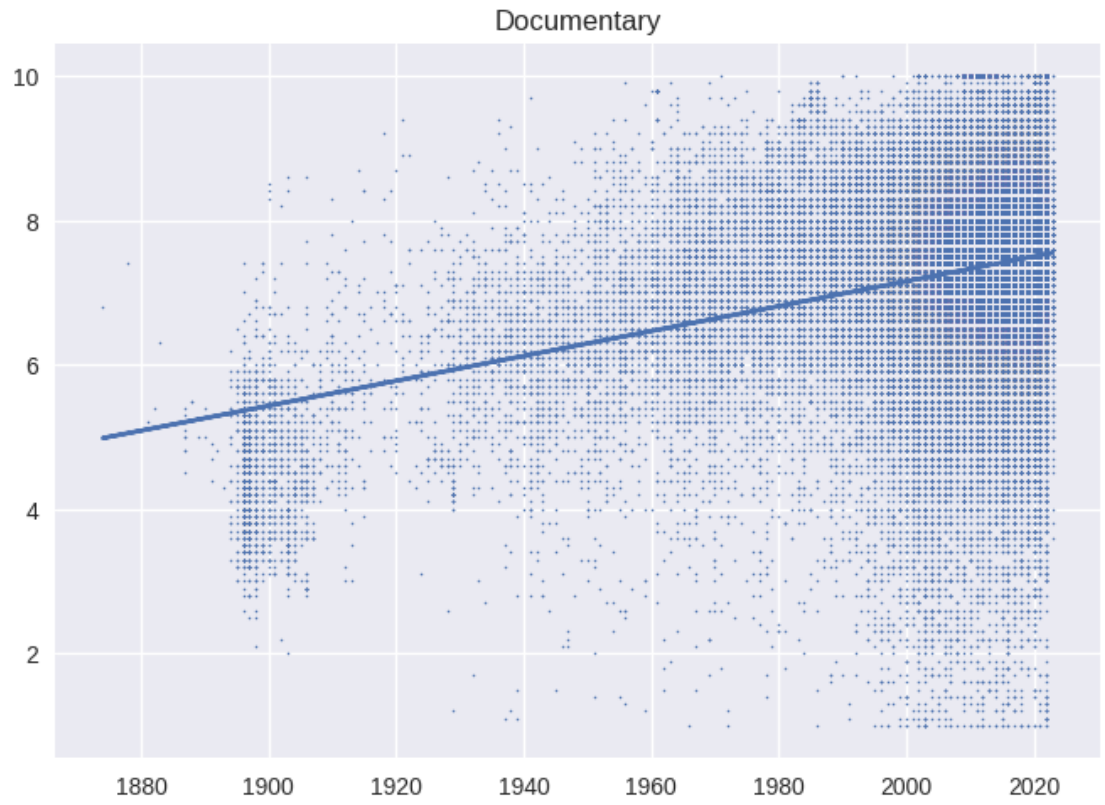

```
best fit line:
y = -13.80 + 0.01x
```



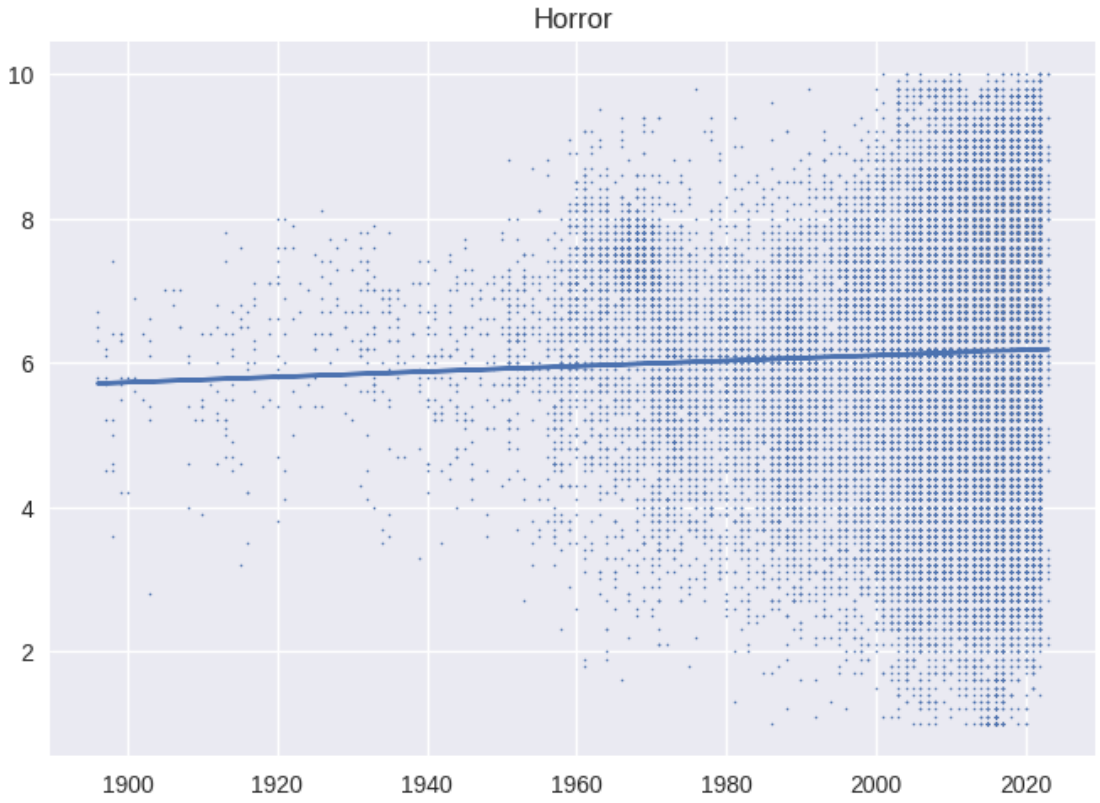
best fit line:
y = -52.38 + 0.03x



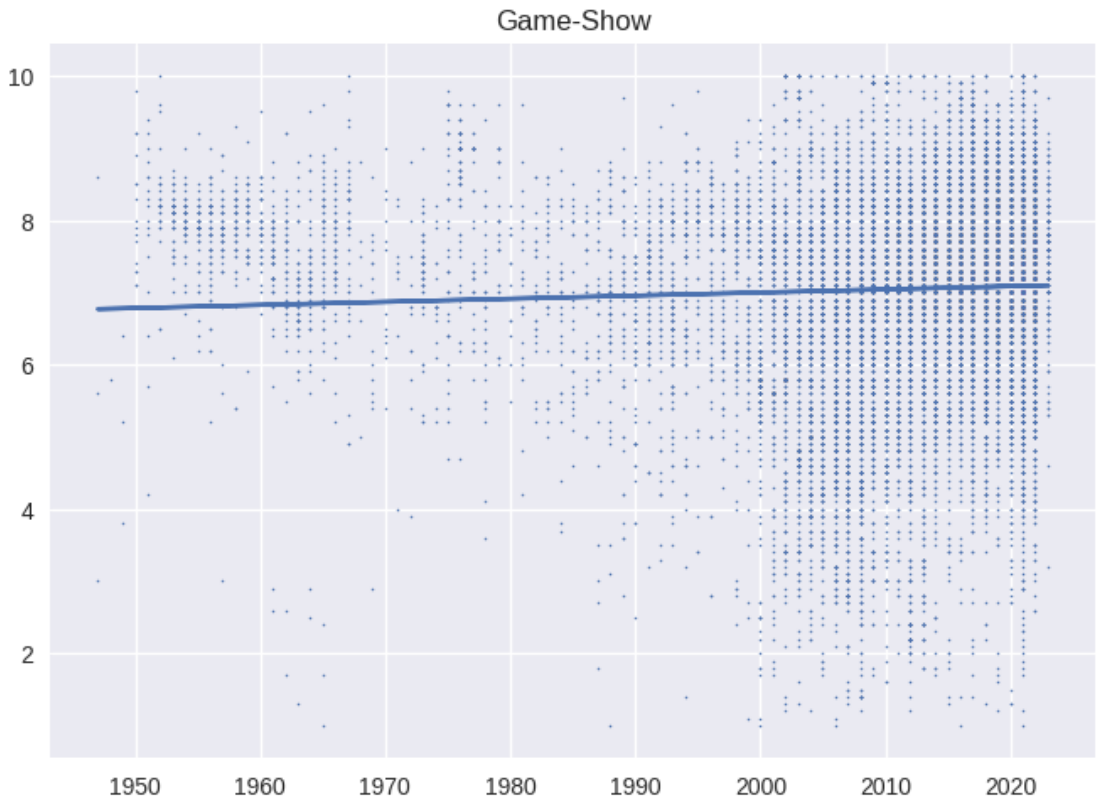
best fit line:
y = -27.32 + 0.02x



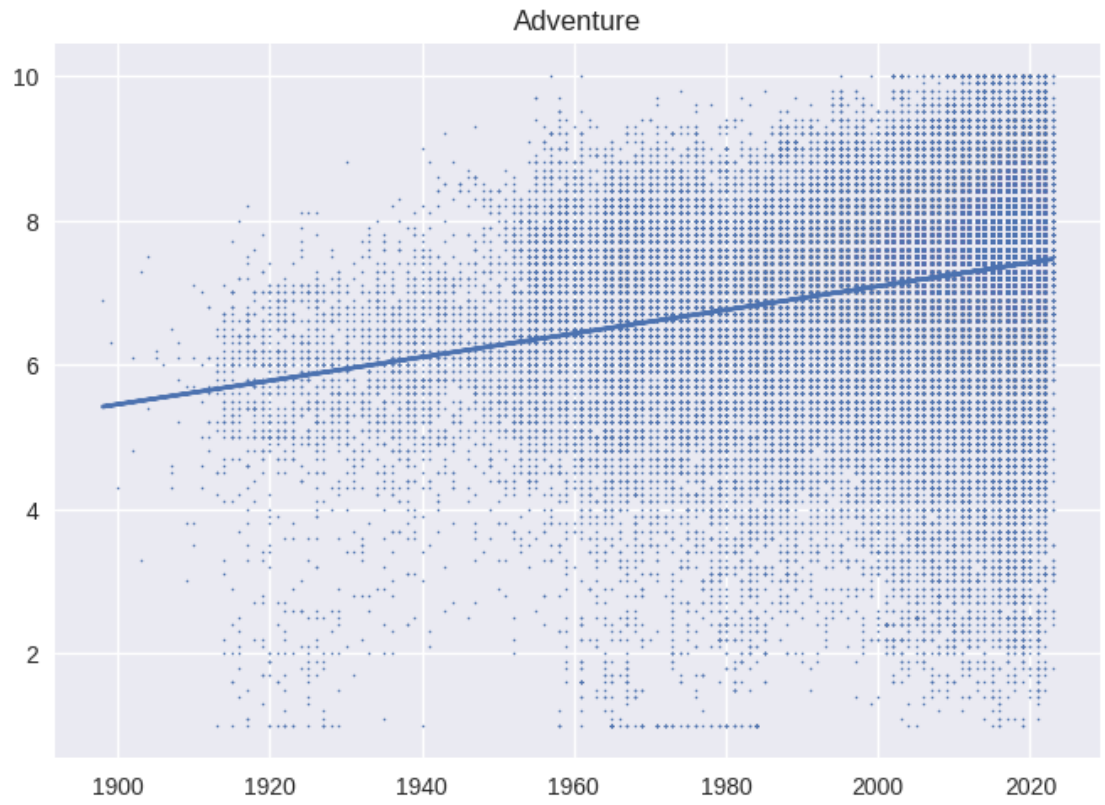

```
best fit line:
y = -1.38 + 0.00x
```




```
best fit line:
y = -1.64 + 0.00x
```



best fit line:
y = -25.62 + 0.02x




```
<Figure size 800x550 with 0 Axes>
```

```
print(bestfits)
highestb = 0
highestbg = ""
highesta = 0
highestag = ""

lowestb = 0
lowestbg = ""
lowesta = 0
lowestag = ""

for genre in bestfits:
  if bestfits[genre][0] > highestb:
    highestb = bestfits[genre][0]
    highestbg = genre
  if bestfits[genre][1] > highesta:
    highesta = bestfits[genre][1]
    highestag = genre
  if bestfits[genre][0] < lowestb:
    lowestb = bestfits[genre][0]
    lowestbg = genre
  if bestfits[genre][1] < lowesta:
    lowesta = bestfits[genre][1]
```

```
    lowestag = genre

print(f"Highest starting value: {highestbg} with {highestb}\nMost
improvement over the years: {highestag} with {highesta}")
print(f"Lowest starting value: {lowestbg} with {lowestb}\nMost
degraded over the years: {lowestag} with {lowesta}")
```

```
{'Crime': [-13.671131278847193, 0.010395788689438055], 'Comedy': [-
11.795263229268372, 0.009838963677214 97], 'Mystery': [-
8.466216847828097, 0.007781740083392541], 'History': [-
14.26194920677155, 0.010773194898180736], 'Talk-Show': [-
15.88283760667808, 0.011307874820479095], 'Music': [-17.1673809843522,
0.01207869555941277], 'Fantasy': [-16.04128647052476,
0.011532544104470056], 'Sport': [-23.33018962057568,
0.015170432738985837], 'Biography': [-15.070878855792985,
0.011106796646604895], 'Musical': [-19.32058360953547,
0.013067473863163316], 'News': [-43.23757993147416,
0.024833859310266243], 'Sci-Fi': [-14.246102994280857,
0.010466334309625723], 'Action': [-22.312620581447334,
0.014638582852197775], 'Thriller': [-16.62725084935351,
0.011474376965610 87], 'Reality-TV': [-44.803702557104444,
0.025699811711743206], 'Family': [4.282853321573924,
0.0013987242237185034], 'Animation': [-24.080781062233218,
0.015569486341862723], 'Drama': [-11.871671560238065,
0.009456469018614395], 'Romance': [-16.55990523238937,
0.011706751384420212], 'Western': [-11.402267414815412,
0.009342341883102066], 'Short': [-29.181550214166915,
0.01802888092125245], 'Film-Noir': [23.14437657145985, -
0.008561425435171945], 'War': [-13.802720984943392,
0.010485570230823362], 'Adult': [-52.37935764784966,
0.029393270694227626], 'Documentary': [-27.323167878926267,
0.017238287709639758], 'Horror': [-1.3837196689972426,
0.0037421571538692896], 'Game-Show': [-1.6358118914238595,
0.004317070650578428], 'Adventure': [-25.621622480216825,
0.016353280429719237]}
Highest starting value: Film-Noir with 23.14437657145985
Most improvement over the years: Adult with 0.029393270694227626
Lowest starting value: Adult with -52.37935764784966
Most degraded over the years: Film-Noir with -0.008561425435171945
```

lets see how the numbers match up

```
difference = {}
for genre in bestfits:
  a = testbestfits[genre][0] - testbestfits[genre][0]
  b = testbestfits[genre][1] - testbestfits[genre][1]
  difference[genre] = [a,b]

for genre in difference:
  print(f"{genre}: {difference[genre][0]} {difference[genre][1]}")
```

```
Crime: 0.0 0.0
Comedy: 0.0 0.0
Mystery: 0.0 0.0
History: 0.0 0.0
Talk-Show: 0.0 0.0
Music: 0.0 0.0
Fantasy: 0.0 0.0
Sport: 0.0 0.0
Biography: 0.0 0.0
Musical: 0.0 0.0
News: 0.0 0.0
Sci-Fi: 0.0 0.0
Action: 0.0 0.0
Thriller: 0.0 0.0
Reality-TV: 0.0 0.0
Family: 0.0 0.0
Animation: 0.0 0.0
Drama: 0.0 0.0
Romance: 0.0 0.0
Western: 0.0 0.0
Short: 0.0 0.0
Film-Noir: 0.0 0.0
War: 0.0 0.0
Adult: 0.0 0.0
Documentary: 0.0 0.0
Horror: 0.0 0.0
Game-Show: 0.0 0.0
Adventure: 0.0 0.0
```

Our results seem to match up between testing and training data. This is most likely due to rounding and our large number of data points.