In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

Main Question: Does the number of hours a streamer spends streaming predict the total number of followers gained on Twitch?
Hypothesis: Streamers who spend more hours streaming tend to gain more followers due to increased visibility and audience engagement

In [2]:
```
df = pd.read_csv("twitchdata.csv")
df.head()
```

Out[2]:

	Channel	Watch time(Minutes)	Stream time(minutes)	Peak viewers	Average viewers	Followers	Followers gained	Views gained	Partnered	Mature	Language
0	xQcOW	6196161750	215250	222720	27716	3246298	1734810	93036735	True	False	English
1	summit1g	6091677300	211845	310998	25610	5310163	1370184	89705964	True	False	English
2	Gaules	5644590915	515280	387315	10976	1767635	1023779	102611607	True	True	Portuguese
3	ESL_CSGO	3970318140	517740	300575	7714	3944850	703986	106546942	True	False	English
4	Tfue	3671000070	123660	285644	29602	8938903	2068424	78998587	True	False	English

In [3]:
```
df.tail()
```

Out[3]:

	Channel	Watch time(Minutes)	Stream time(minutes)	Peak viewers	Average viewers	Followers	Followers gained	Views gained	Partnered	Mature	Language
995	LITkillah	122524635	13560	21359	9104	601927	562691	2162107	True	False	Spanish
996	빅헤드 (bighead033)	122523705	153000	3940	793	213212	52289	4399897	True	False	Korean
997	마스카 (newmasca)	122452320	217410	6431	567	109068	-4942	3417970	True	False	Korean
998	AndyMilonakis	122311065	104745	10543	1153	547446	109111	3926918	True	False	English
999	Remx	122192850	99180	13788	1205	178553	59432	2049420	True	False	French

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 11 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
 0   Channel               1000 non-null   object
 1   Watch time(Minutes)   1000 non-null   int64 
 2   Stream time(minutes)  1000 non-null   int64 
 3   Peak viewers          1000 non-null   int64 
 4   Average viewers       1000 non-null   int64 
 5   Followers             1000 non-null   int64 
 6   Followers gained      1000 non-null   int64 
 7   Views gained          1000 non-null   int64 
 8   Partnered             1000 non-null   bool  
 9   Mature                1000 non-null   bool  
 10  Language              1000 non-null   object
dtypes: bool(2), int64(7), object(2)
memory usage: 72.4+ KB
```

Basic info about data:

Data has 1000 rows and 11 columns

most of the data is numeric (Watch Time, Stream Time, Peak Viewers, Average Viewers, Followers, Followers Gained, Views Gained)

With Categotical columns like Channel and Languages

And Booleen columns like Partnered, Mature

As per the data description - The time period of this data is 1 year (365 days)

In [5]:
```
df.isna().sum()
```

Out[5]:
```
Channel                 0
Watch time(Minutes)     0
Stream time(minutes)    0
Peak viewers            0
Average viewers         0
Followers               0
Followers gained        0
Views gained            0
Partnered               0
Mature                  0
Language                0
dtype: int64
```

The data has no na or missing values

Initial inspection of the data revels that the column "Channel" has names in different languates with the translated name in brackets. For the sake of consistency lets keep all the names in English

We will also need consistent conlumn names

And the time columns need to be in days instead of mins for better readability and understanding

Another thing to note is that this data is biased and does not capture the entire population correctly as these are only the "TOP" 1000 streamers, and streamers with lower stream times may not be captured in this data completely here. So if our model were to be trained on this data, it may not be able to predit correctly for all the values.

In [6]:
```
# Renaming Columns
df2 = df.rename(columns = {"Watch time(Minutes)": "Watch time (mins)", "Stream time(minutes)": "Stream time (mi
df2.head()
```

Out[6]:

	Channel	Watch time (mins)	Stream time (mins)	Peak viewers	Average viewers	Followers	Followers gained	Views gained	Partnered	Mature	Language
0	xQcOW	6196161750	215250	222720	27716	3246298	1734810	93036735	True	False	English
1	summit1g	6091677300	211845	310998	25610	5310163	1370184	89705964	True	False	English
2	Gaules	5644590915	515280	387315	10976	1767635	1023779	102611607	True	True	Portuguese
3	ESL_CSGO	3970318140	517740	300575	7714	3944850	703986	106546942	True	False	English
4	Tfue	3671000070	123660	285644	29602	8938903	2068424	78998587	True	False	English

In [7]:
```
# Converting Stream Time in mins to days
df2['Watch time (days)'] = ((df2['Watch time (mins)']/60)/24).round().astype(int)
df2['Stream time (days)'] = ((df2['Stream time (mins)']/60)/24).round().astype(int)
df2.head()
```

Out[7]:

	Channel	Watch time (mins)	Stream time (mins)	Peak viewers	Average viewers	Followers	Followers gained	Views gained	Partnered	Mature	Language	Watch time (days)
0	xQcOW	6196161750	215250	222720	27716	3246298	1734810	93036735	True	False	English	4302890
1	summit1g	6091677300	211845	310998	25610	5310163	1370184	89705964	True	False	English	4230331
2	Gaules	5644590915	515280	387315	10976	1767635	1023779	102611607	True	True	Portuguese	3919855
3	ESL_CSGO	3970318140	517740	300575	7714	3944850	703986	106546942	True	False	English	2757165
4	Tfue	3671000070	123660	285644	29602	8938903	2068424	78998587	True	False	English	2549306

In [8]:
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 13 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Channel             1000 non-null   object
 1   Watch time (mins)   1000 non-null   int64
 2   Stream time (mins)  1000 non-null   int64
 3   Peak viewers        1000 non-null   int64
 4   Average viewers     1000 non-null   int64
 5   Followers           1000 non-null   int64
 6   Followers gained    1000 non-null   int64
 7   Views gained        1000 non-null   int64
 8   Partnered           1000 non-null   bool
 9   Mature              1000 non-null   bool
 10  Language            1000 non-null   object
 11  Watch time (days)   1000 non-null   int64
 12  Stream time (days)  1000 non-null   int64
dtypes: bool(2), int64(9), object(2)
memory usage: 88.0+ KB
```

In [9]:
```
df2.describe().round().astype(int)
```

Out[9]:

	Watch time (mins)	Stream time (mins)	Peak viewers	Average viewers	Followers	Followers gained	Views gained	Watch time (days)	Stream time (days)
count	1000	1000	1000	1000	1000	1000	1000	1000	1000
mean	418427930	120515	37065	4781	570054	205519	11668166	290575	84
std	549635514	85376	60314	8454	804413	339914	24905722	381691	59
min	122192850	3465	496	235	3660	-15772	175788	84856	2
25%	163189894	73759	9114	1458	170546	43758	3880602	113327	51
50%	234990788	108240	16676	2425	318063	98352	6456324	163188	75
75%	433739918	141844	37570	4786	624332	236131	12196762	301208	98
max	6196161750	521445	639375	147643	8938903	3966525	670137548	4302890	362

There are some negative values for followers gained which might mean that those streams actually lost the that many followers.

In [10]:
```
df2[(df2["Followers gained"]<0)]
```

Out[10]:

	Channel	Watch time (mins)	Stream time (mins)	Peak viewers	Average viewers	Followers	Followers gained	Views gained	Partnered	Mature	Language	Watch time (days)
499	Amaz	235255500	84525	12652	2764	915023	-5405	4237993	True	False	English	163372
656	TSM_TheOddOne	181908120	188445	4363	913	864087	-15772	6370949	True	False	English	126325
997	마스카 (newmasca)	122452320	217410	6431	567	109068	-4942	3417970	True	False	Korean	85036

Making sense of the summary stats

We are interested in learning about hrs spent streaming and the followers

The mean streaming days is really, really low compared to expectations — 84 days. Assuming that a streamer spends 8 hours daily streaming, this corresponds to 252 days out of 365.

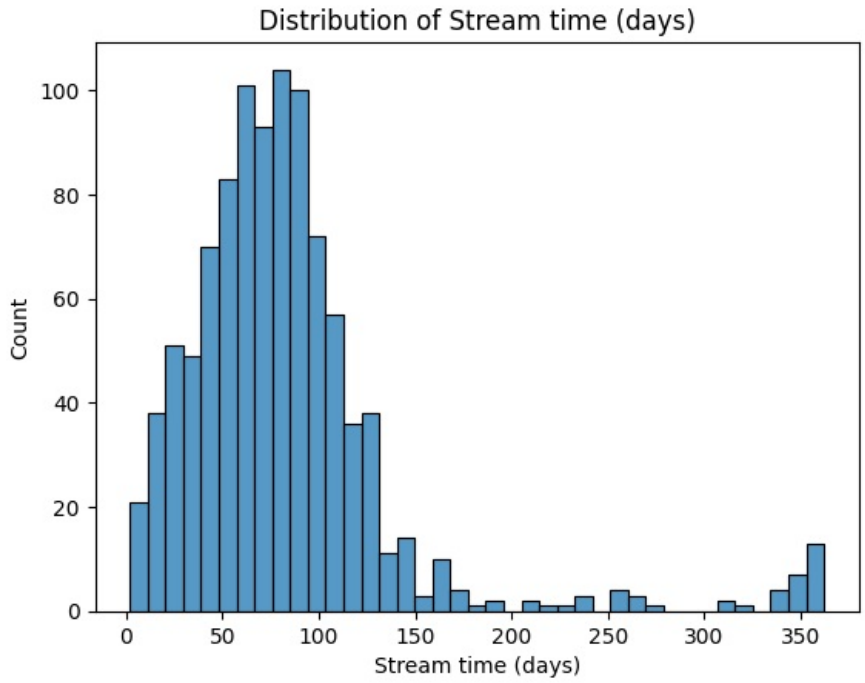
The minimum stream time is 2 days, and the maximum stream time is 362 days, which is greater than the number of days in a year. Given that a person cannot stream continuously for such a long duration, this and similar observations are likely incorrect. Even if it were assumed that the streamer left their stream on continuously for the entire year (365 days), it would still not exceed that limit.

Therefore, either the data collected covers more than one year, meaning the mentioned time period is inaccurate, or there is a data collection error.

I have reached out to the dataset author to clarify this. The wait time is 7 days, after which a follow-up will be sent. If no response is received within this timeframe, these observations will be considered data collection errors and removed from the dataset.

In [11]:
```
sns.histplot(data = df2, x = "Stream time (days)")
plt.title("Distribution of Stream time (days)")
```

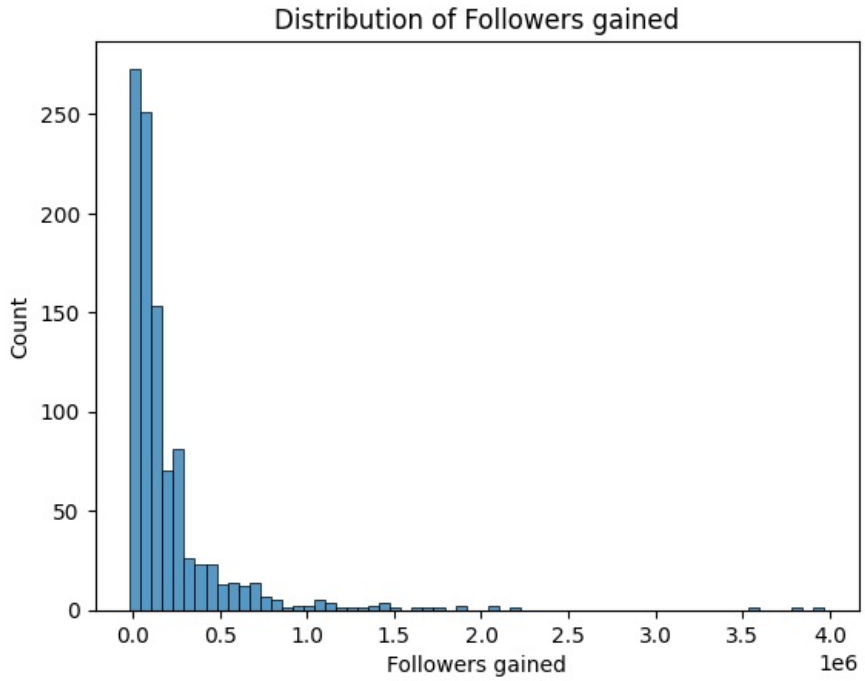
Out[11]:
```
Text(0.5, 1.0, 'Distribution of Stream time (days)')
```



This histplot shows that the data is right skewed, with most of the streamer streaming between 2 to 140 days

In [12]:
```
sns.histplot(data = df2, x = "Followers gained")
plt.title("Distribution of Followers gained")
```

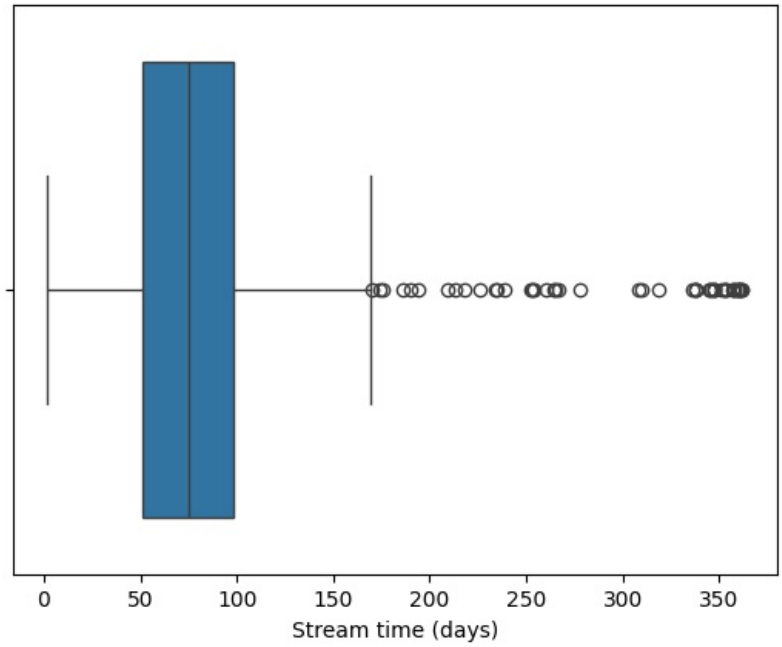
Out[12]:
```
Text(0.5, 1.0, 'Distribution of Followers gained')
```



Most of the streams gained between 0 to 50k follower, with some streamers gaining between 350k and 400k followers

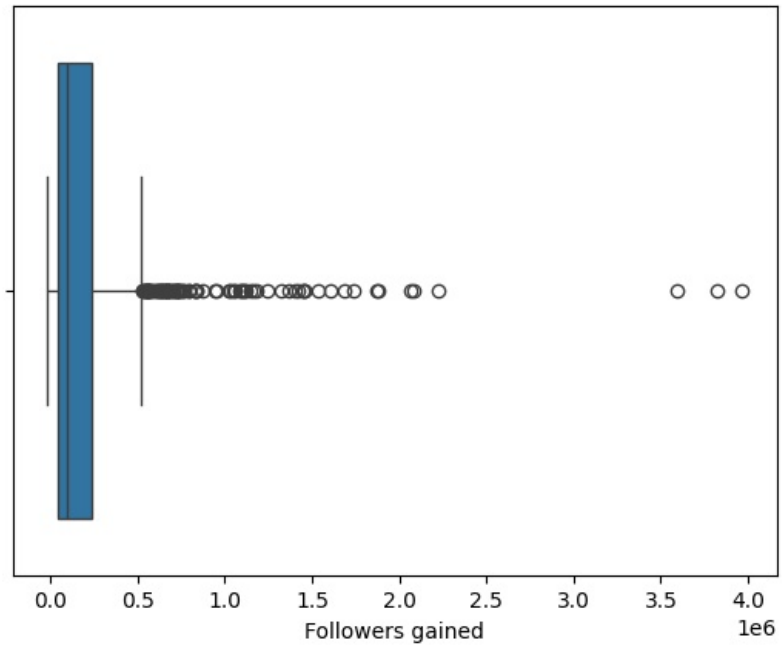
In [13]:
```
sns.boxplot(data = df2, x = "Stream time (days)")
```

Out[13]:
```
<Axes: xlabel='Stream time (days)'>
```



In [14]:
```
sns.boxplot(data = df2, x = "Followers gained")
```

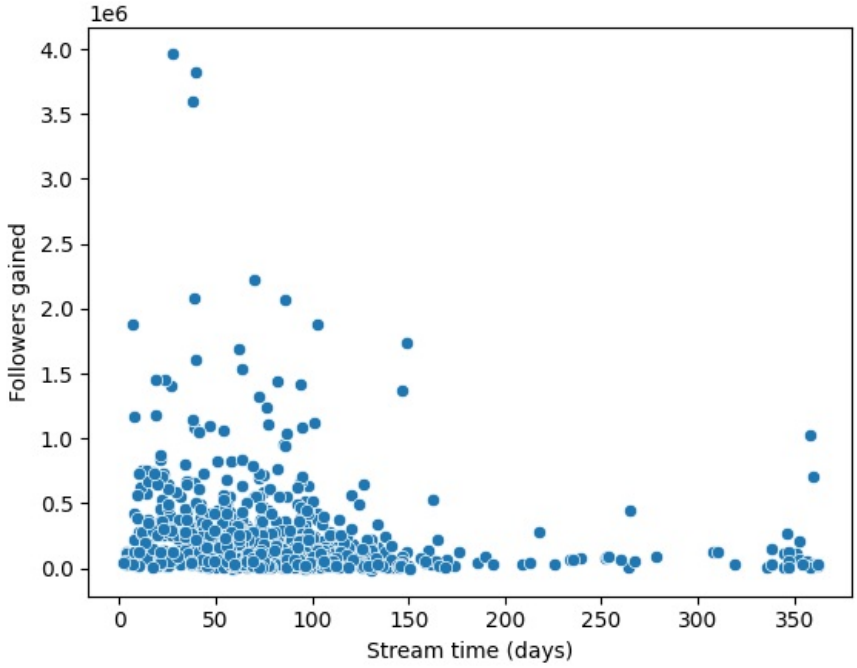
Out[14]:
```
<Axes: xlabel='Followers gained'>
```



Further investigating reveals potential outlier for both of these variables.

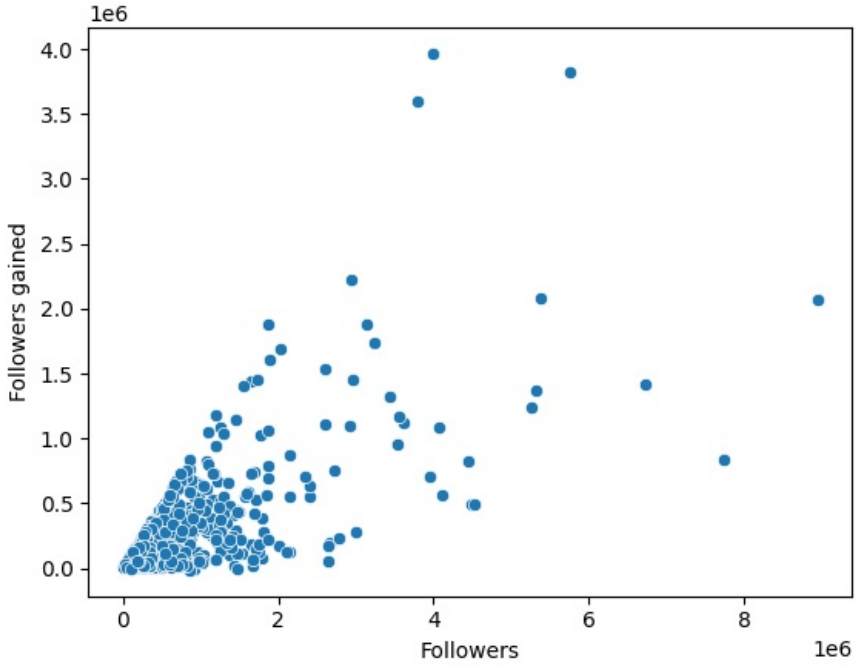
In [15]:
```
sns.scatterplot(data=df2, y="Followers gained", x="Stream time (days)")
```

Out[15]:
```
<Axes: xlabel='Stream time (days)', ylabel='Followers gained'>
```



In [20]:
```
sns.scatterplot(data=df2, y="Followers gained", x="Followers")
```

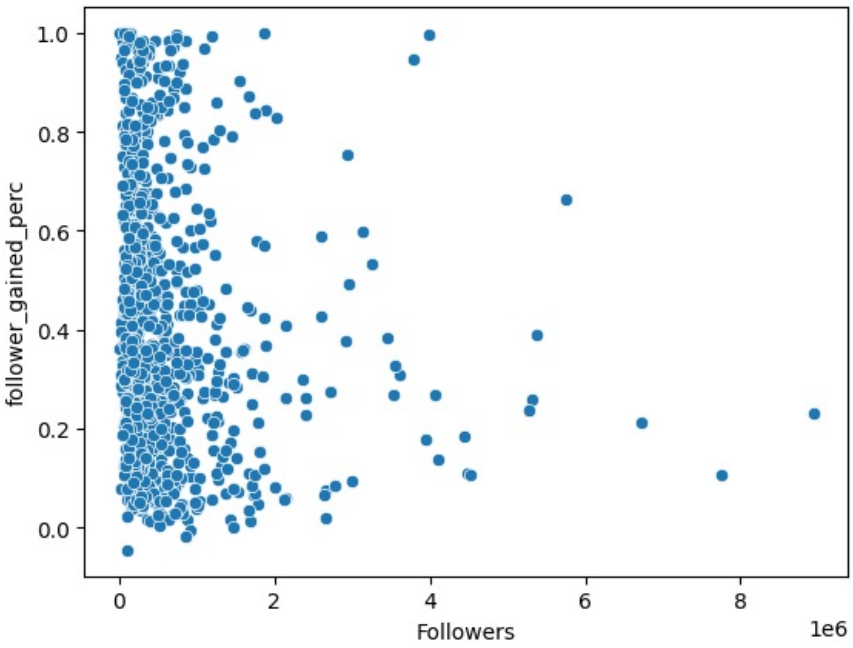
Out[20]:
```
<Axes: xlabel='Followers', ylabel='Followers gained'>
```



In [40]:
```
# df2['follower_gained_perc'] = df2['Followers gained']/(df2['Followers'] + 1)
sns.scatterplot(data=df2, y="follower_gained_perc", x="Followers")

df3 = df2[(df2['Followers'] > 1500000) & (df2['Stream time (days)'] < 200) ]
# sns.scatterplot(data=df3, y="follower_gained_perc", x="Stream time (days)")

df3[(df3['follower_gained_perc']>0.9) & (df3['Stream time (days)'] < 20)]
```

Out[40]:

	Channel	Watch time (mins)	Stream time (mins)	Peak viewers	Average viewers	Followers	Followers gained	Views gained	Partnered	Mature	Language	Watch time (days)	Stre ti (da
669	SLAKUN10	179262330	9555	48358	18906	1874932	1874846	5835029	True	False	Spanish	124488	



In [16]:
```
Q1 = df2["Stream time (days)"].quantile(0.25)
Q3 = df2["Stream time (days)"].quantile(0.75)
IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

df_clean = df2[(df2["Stream time (days)"] >= lower_bound) & (df2["Stream time (days)"] <= upper_bound)]
```

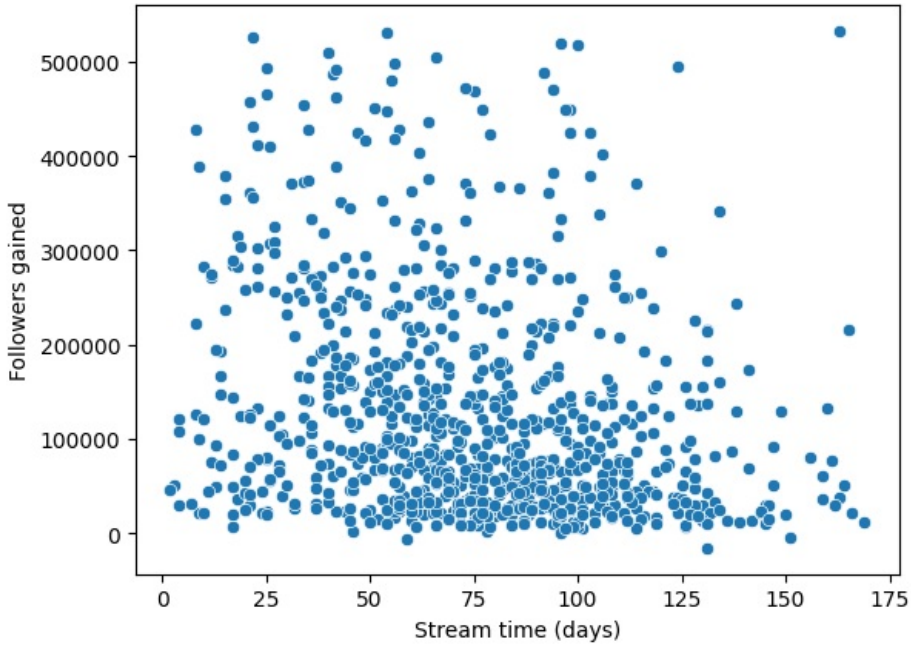
In [17]:
```
Q1 = df_clean["Followers gained"].quantile(0.25)
Q3 = df_clean["Followers gained"].quantile(0.75)
IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

df_cleaned = df_clean[(df_clean["Followers gained"] >= lower_bound) & (df_clean["Followers gained"] <= upper_bou
```

In [18]:
```
df_cleaned.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 867 entries, 7 to 999
Data columns (total 13 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Channel             867 non-null    object
 1   Watch time (mins)   867 non-null    int64
 2   Stream time (mins)  867 non-null    int64
 3   Peak viewers        867 non-null    int64
 4   Average viewers     867 non-null    int64
 5   Followers           867 non-null    int64
 6   Followers gained    867 non-null    int64
 7   Views gained        867 non-null    int64
 8   Partnered           867 non-null    bool
 9   Mature              867 non-null    bool
 10  Language            867 non-null    object
 11  Watch time (days)   867 non-null    int64
 12  Stream time (days)  867 non-null    int64
dtypes: bool(2), int64(9), object(2)
memory usage: 83.0+ KB
```

In [ ]:
```
sns.scatterplot(data=df_cleaned, y="", x="Stream time (days)")
```

Out[ ]:
```
<Axes: xlabel='Stream time (days)', ylabel='Followers gained'>
```



In [45]:
```
df_cleaned['follower_gained_perc'] = df_cleaned['Followers gained']/(df_cleaned['Followers'] + 1)
df3 = df_cleaned[(df_cleaned['Followers'] > 1500000) & (df_cleaned['Stream time (days)'] < 200) ]
# sns.scatterplot(data=df3, y="follower_gained_perc", x="Stream time (days)")
sns.scatterplot(data=df_cleaned, y="follower_gained_perc", x="Stream time (days)")
```

Out[45]:
```
<Axes: xlabel='Stream time (days)', ylabel='follower_gained_perc'>
```