



Spotify®

What makes a playlist successful?

George John Jordan Thomas Aquinas Hayward, Optimist

Today's Discussion

Topic Outline

Survey Results

Strategy

Data Preparation

Exploratory Data Analysis & Visualization

Modeling

Conclusion

Next Steps

Thank You!



Spotify®

Survey Results



A word cloud visualization centered around music and playlists. The most prominent words are "playlist", "music", "songs", and "artists". Other significant words include "play", "listen", "time", "mood", "used", "etc", "right", "old", "new", "mix", "vibes", "house", "want", "like", "Spotify", and "matches". The words are arranged in a circular pattern, with smaller words forming the outer ring and larger words in the center. The font color is a dark shade of green, and the background is black.



WHAT MAKES A PLAYLIST GOOD?



Mary

"When it fits a mood; when I don't feel like skipping tracks; when it's a collab; when it has a mix of old and new songs of similar genre."



Ben

"It has some songs I know and love, but also helps me discover new songs. It matches my mood."



Ray

"When the whole playlist matches the exact vibe you're looking for perfectly (pre-game, 90s, gym flow, etc.), but also has a wide range of artists and song types."

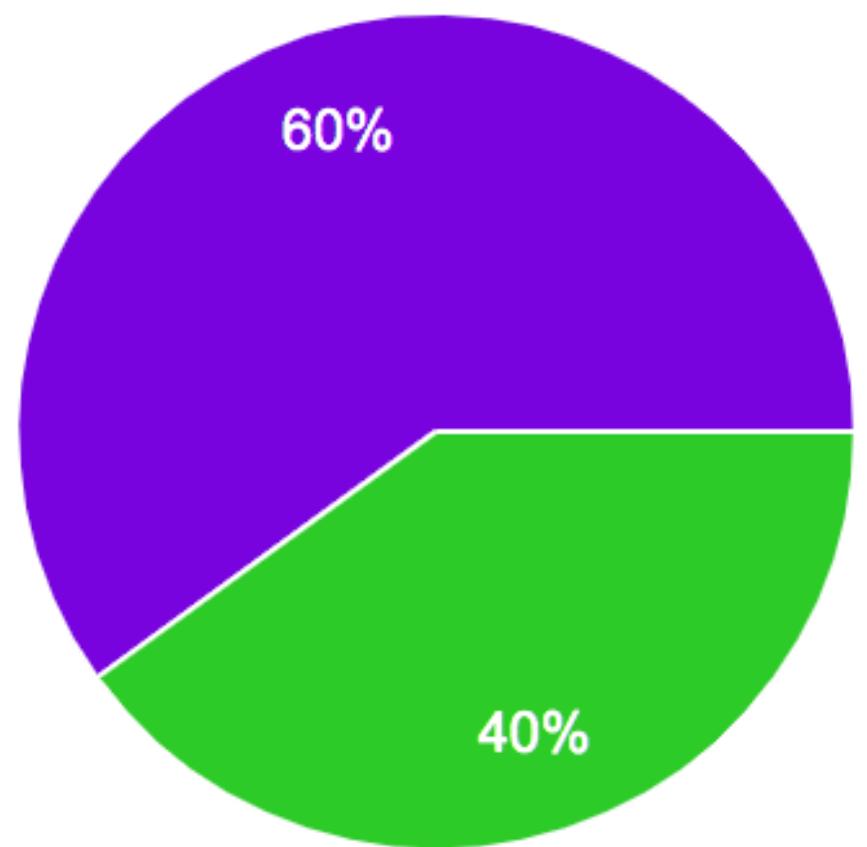


SURVEY BREAKDOWN

Gender

20 responses

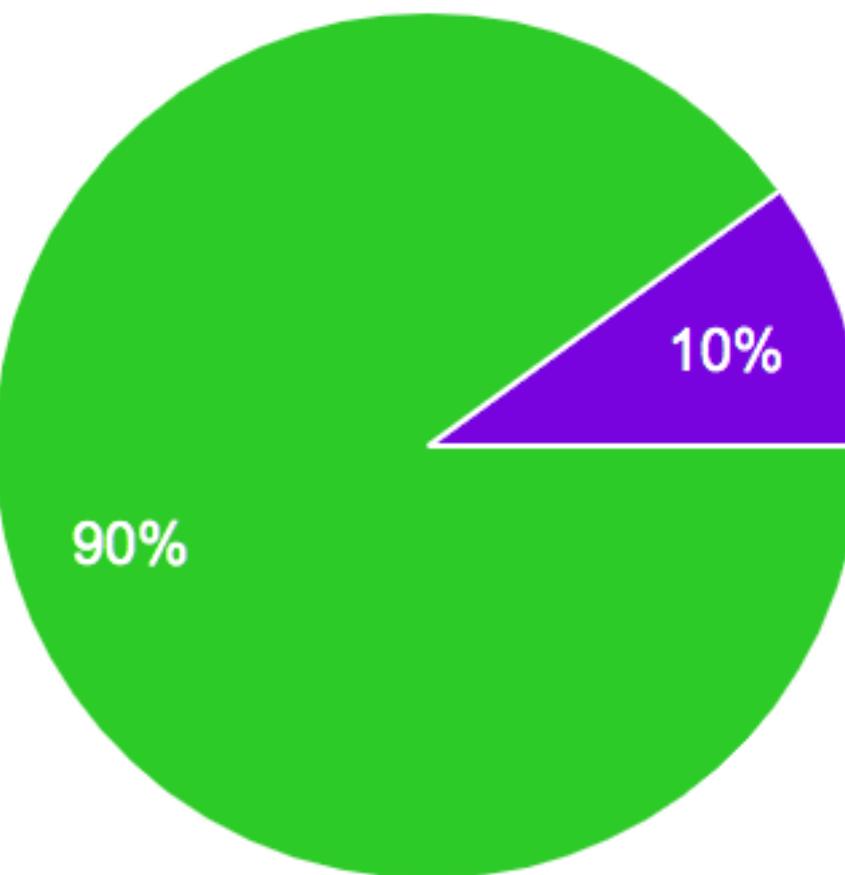
- Female
- Male
- Prefer not to say



Are you a Spotify Premium subscriber?

20 responses

- Yes
- No

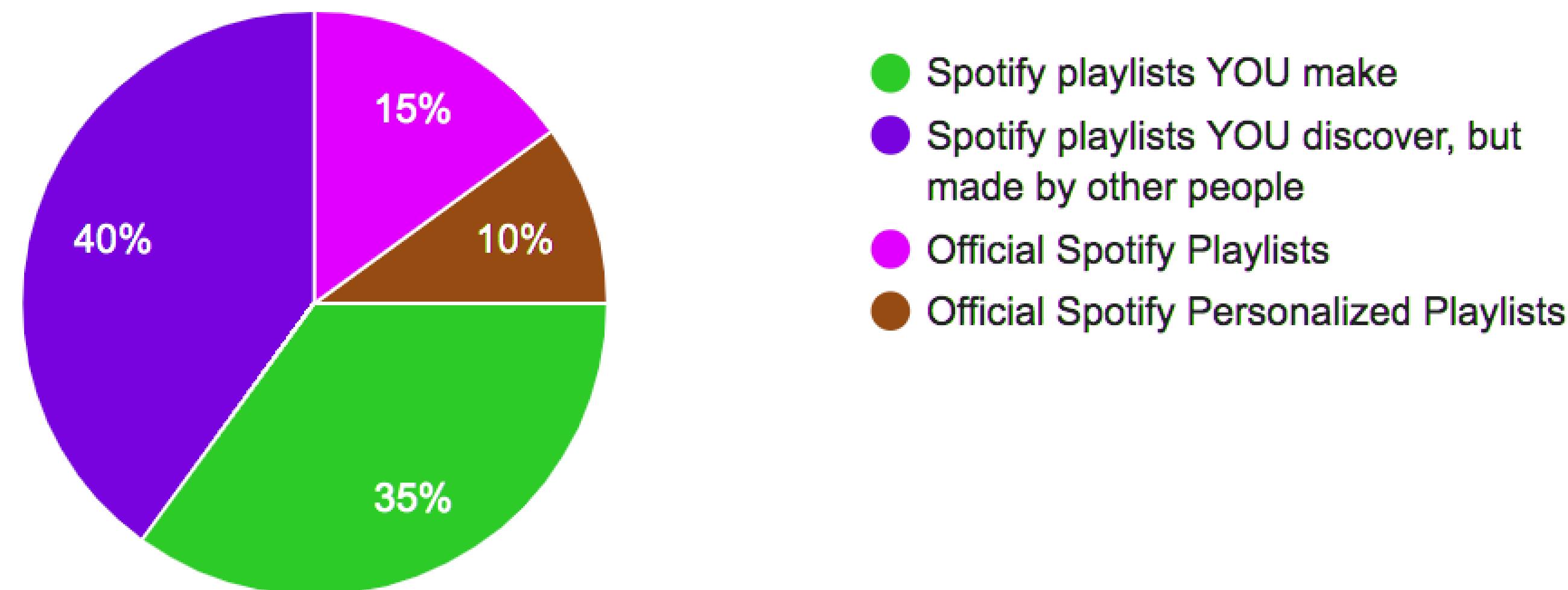




YOUR FAVORITE PART OF SPOTIFY?

If you had to choose, which of these is most important to your Spotify experience?

20 responses





Spotify®

Strategy



Game Plan



Phase 1

Open the data.

Phase 2

Look at all the columns. Categorical? Continuous?

Phase 3

Decide the dependent variable. What does success mean? What is the success metric?

Phase 4

Engineer features; visualize data; look for relationships; model.

Phase 5

Insights.

403,366 playlists

403,366 playlists

403,366 playlists

403,366 playlists

LOTS OF NUMBERS TO CRUNCH

• • •



Song Metadata & Playlists																
Spotify ID	User	Country	Length (ms)	Explicit	Popularity	Key	Mode	Artist	Title	Album	Genre	Danceability	Loudness (dB)	Tempo (BPM)	Energy	Valence
spotify:user:7310382bc047820dc5d324d2772a03	playlist:2hf0f325qqt4111vrksjz	1 US 455 2 341 381 332 321	["slaylist", "looking"] Pop Rap R&B	Defiant	Excited	Energizing	1 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:35eb3d66eb58ae6e4fd35bab46e5542e	playlist:1vhA70enKQTWZOPPTij1gX	0 US 444 0 93 114 129 110	["christian", "best"] Religious Rock	35eb3d66eb58ae6e4fd35bab46e5542e	18	13	1 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:2ac357e58edb6da0f92ff7f693680104	playlist:5ofVnH33nxzyTZ68WZ9nUU	0 US 1726 0 632 699 933 930	["lista", "mami"] Latin Pop Dance & House	2ac357e58edb6da0f92ff7f693680104	9	9	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:7dcbc955cef93e082bc3905cad5c8a1	playlist:65EFm0jcCYfyeca1H3sVUH	0 US 154 0 75 75 328 99	["new"] Dance & House Electronica	7dcbc955cef93e082bc3905cad5c8a1	68	64	2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:2b4069fcfac91052eb60c8ed0eded2fd	playlist:4FnHzHjy0NFkbg798UhKgp	0 US 19 0 18 18 15 1	["soda", "pop", "summer"] Pop Dance & House	2b4069fcfac91052eb60c8ed0eded2fd	0	0	0 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:105fe547c61aa8f9d3831dbbf0103992	playlist:15s2Hjp1gGrWbQcmMRnZS5	0 US 79 0 2 4 11 6	["night", "sax", "music"] Jazz Rock	105fe547c61aa8f9d3831dbbf0103992	0	0	0 1 3 2 1 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:29ee8e37b23c6209d07e4186f79e62f3	playlist:3RRpMk6upNjCyF7aGip8lT	0 US 268 0 7 11 16 0	["ottmar", "liebert", "collection"] Traditional	29ee8e37b23c6209d07e4186f79e62f3	0	0	0 0 2 1 0 0 0 0 0 0 0 0 0 0 0 0									
spotify:user:400e06c9a9898115013a895ab04f92ca	playlist:5fyuKwejya1d7xohj4hM2F	0 US 128 0 46 46 235 234	["music"] Pop Indie Rock	400e06c9a9898115013a895ab04f92ca	0	0	0 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:6d01fc6bf88c2a2854b59267c4f8f88d	playlist:4tt2LqgrKDzdz7KNqbLdqS	0 US 183 0 152 161 72 50	[] Electronica Indie Rock	6d01fc6bf88c2a2854b59267c4f8f88d	0	0	0 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2									
spotify:user:b93fb3c8c8dd61f6aee58db5ef6114d	playlist:5QnmGbt6bM50AW097LQY57	0 US 330 1 258 270 74 1	["zoned"] Latin Dance & House	b93fb3c8c8dd61f6aee58db5ef6114d	8	4	1 2 5 4 3 3 3 3 3 3 3 3 3 3 3 3									
spotify:user:be98b3e1dc68d33b1d19f99b2c49d2c8	playlist:7mgw19ra8MT7ryUosbJ0Kx	0 US 9 0 1 1 20 1	["empire", "sun", "walking", "dream"] Electronica	be98b3e1dc68d33b1d19f99b2c49d2c8	5	5	1 2 4 5 1 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:268f3524d209485667e3ab43ac27c29e	playlist:6jQYogkmD5VwE04fNM0W7w	0 US 70 0 44 51 67 52	["country", "music"] Country & Folk	268f3524d209485667e3ab43ac27c29e	0	0	0 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:63aaea00df3ca978e92761292a2eaa90	playlist:5XH0WG5KE0ZxtATzewUaWg	0 US 66 0 1 5 78 1	["steve", "pettit", "evangelistic", "team", "high", "price"]	63aaea00df3ca978e92761292a2eaa90	0	0	0 2 3 3 1 1 1 1 1 1 1 1 1 1 1 1									
Empowering Sophisticated	Religious	-	-	Romantic	-	-	-	-	-	-	-	-	-	-	-	
spotify:user:4386d0317126baad62457b43bff3e500	playlist:3TR37MZYLGp7tp5aEiUH0h	0 US 501 15 133 141 56 48	[] Indie Rock Pop Alternative	4386d0317126baad62457b43bff3e500	15	3	1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:64672b35946481aa3953fa492273718f	playlist:0KIQsi7z3Ghu1SAjPhJcvm	1 US 521 2 245 79 78 15	["ragga", "jungle"] Dance & House Reggae	64672b35946481aa3953fa492273718f	1	1	1 2 4 4 2 2 2 2 2 2 2 2 2 2 2 2									
spotify:user:6234eea77f077d721bc8375bc2fe505e	playlist:4bPfPAonxfMaPBPRluDo4V	0 US 76 0 47 60 157 68	["viejitas", "bonitas"] Latin Jazz	6234eea77f077d721bc8375bc2fe505e	0	0	0 2 5 3 3 3 3 3 3 3 3 3 3 3 3 3									
spotify:user:9f472ecbc51af7e92e3efd23ac1d11ce	playlist:6Pmu2P102HWmg8zCB1BPZ9	0 US 31 0 2 7 17 8	["van", "halen"] Rock	9f472ecbc51af7e92e3efd23ac1d11ce	0	0	0 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:78a6655af797a286b32d6fb67584ba00	playlist:6DICLM6DYotMGeUmJqIulf	0 US 503 7 353 386 20 1	["top", "songs"] Rap R&B	78a6655af797a286b32d6fb67584ba00	0	0	0 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2									
spotify:user:631e3296e38597c66cd488becd7	playlist:1FQCF1LBHSWPuYOCXdNKAoH	0 US 78 0 59 67 25 11	["abby"] Indie Rock	631e3296e38597c66cd488becd7	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0									
spotify:user:920995a70302b6ae67c65065cf855764	playlist:4J0FXDR0DXVujDQ3YakyJc	0 US 14 0 1 1 208 68	["miles", "davis", "sketches", "spain"] Jazz	920995a70302b6ae67c65065cf855764	0	0	0 3 6 6 0 0 0 0 0 0 0 0 0 0 0 0									
spotify:user:61b3c743795a4f3e0f6c8896b2cdf55	playlist:14Q0ucvnz6BDLcT7WsvI77	0 US 387 3 86 101 181 165	["mah", "faves"] Indie Rock	61b3c743795a4f3e0f6c8896b2cdf55	5	3	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:0c39bfe9cd637f54001b91455ddc9727	playlist:4KXYr7LK0eY6C4Pjz6h4pp	0 US 58 0 23 39 126 109	["jesus"] Religious	0c39bfe9cd637f54001b91455ddc9727	0	0	0 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:4ed73df8f60ddf6f9fd3f7feb1c412d3	playlist:1QW8VQ49ibbdwBEKYQbnBp	0 US 73 0 59 68 4	["classic", "rock"] Rock	4ed73df8f60ddf6f9fd3f7feb1c412d3	12	6	1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2									
spotify:user:43ec110e5340b84f51bfe1b1c6238dc9	playlist:5QdYjbky6Rclfn2ZDH1McJ	1 US 14 0 1 1 212 19	["danger", "doom", "mouse", "mask"] Rap	43ec110e5340b84f51bfe1b1c6238dc9	12	10	1 12 14 14 14 14 14 14 14 14 14 14 14 14									
spotify:user:4caaecbffb7be674fa8446923b55c8a3	playlist:7LPYpGmu75IJw6c0DgQ04g	0 US 40 1 39 36 23 0	["list"] Dance & House	4caaecbffb7be674fa8446923b55c8a3	0	0	0 5 8 8 8 8 8 8 8 8 8 8 8 8 8 8									
spotify:user:270ebca4873a87ddc99e7aeda8a47344	playlist:1YzPf50CqHS0BAL92EPC6c	0 US 9 0 1 1 17 0	["oliver", "swain", "big", "machine"] Country & Folk	270ebca4873a87ddc99e7aeda8a47344	0	0	0 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0									
spotify:user:283fdcc361db98115ea4632404c1563a	playlist:2kJnXLwMu10vimiKZSxeQdG	0 US 108 0 104 104 44 9	["mix"] Dance & House	283fdcc361db98115ea4632404c1563a	0	0	0 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:b50af0f4de752c5b0988829cd3a929b7	playlist:3WdZ8U6Tkrr3rqfQncxQPN	0 US 108 0 104 104 44 9	Latin Pop	b50af0f4de752c5b0988829cd3a929b7	3	1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1									

Song Data										Song Emotions											
Song ID		Song Name		Song Length		Song BPM		Song Key		Song Mode		Song Type		Song Mood		Song Energy		Song Danceability		Song Popularity	
spotify:user:c04367d85e2d85003de9be2c1b5cf00b:playlist:3YwC4p5Ed528zy5PchBvlf	0	US	85	0	72	69	125	113	["classical", "classics"]	Classical	0	0	0	1	2	1	1	2	1	Stirring	
spotify:user:63f30f0682a27dba7751c53c937caa90:playlist:5V0iulVs1nqhBXqkkcHKnP	1	US	43	1	39	39	96	9	["breakers"]	Indie Rock	Alternative	Rock	Yearning	Brooding	Melancholy	3	3	2	2	3	
spotify:user:41069cc8999f842d4914757e9309ab03:playlist:2g3ZnnC4518wZirsbjBI3	0	US	39	0	31	33	303	65	["smooth", "sexy", "salsa"]	Latin Pop	Jazz	Lively	Fiery	Romantic	1	3	1	4	13	13	3
spotify:user:d052537038ec6890906d2b011f38c4f5:playlist:51JWE5aaydEr5tRgsqeZWc	1	US	54	7	43	34	127	55	["love", "keeps", "lifting"]	Rap R&B	Indie Rock	Defiant	Sensual Cool	2	1	1	2	2	1	3	
spotify:user:a9af1290c53595cabba4d6713eea5df5:playlist:5dG7celtJ0a9HVGpXM0Z81	0	US	84	0	9	13	68	45	["folk"]	Rock Country & Folk	Empowering	Gritty	Yearning	1	2	1	1	1	1	2	
spotify:user:a3593d01625fed5ecfeeb42bd902688a:playlist:00CxDCtLKKH2XeafWNkMfI	0	US	60	0	38	28	153	146	["tigers", "norte"]	Latin -	Lively	Cool	Romantic	1	0	0	1	2	1	1	
spotify:user:260fa315fd42a5b4ea6c96a5e6ee4f21:playlist:0BYZbQB6sgVG7ZrUniY88D	0	US	1800	0	128	199	93	87	["study", "muzik", "haha"]	Indie Rock	Alternative	Rock	Brooding	Yearning	0	0	0	1	2	0	2
Melancholy																					
spotify:user:5757e5517ad867a31aac5b79d77bade1:playlist:1eJpIzrLUJthTHBoJvxjkm	0	US	208	0	122	120	46	33	["country", "desert"]	Country & Folk Rock	Pop	Upbeat	Yearning	Empowering	7	2	2	4	7	2	12
spotify:user:9488f2da65ae4194fb146a23ee9f9ad6:playlist:1YX2n0T6teAghj1I0mDA2J	0	US	623	60	302	333	407	393	["collection", "dope", "music"]	Rap Indie Rock	Alternative	Defiant	Excited Cool	2	1	2	2	2	1	3	
spotify:user:6d5a04eca32e0804f338627d09ddc027:playlist:46lBRWLJP0u4ZVHGC673Yp	0	US	242	1	160	180	89	63	["summer", "forever"]	Alternative Indie Rock	Electronica	Defiant	Excited Urgent	5	1	1	3	5	2	6	
spotify:user:ce72c4d76e5431d00a5113f68a69c4b2:playlist:5mXFvxGP8LFCspfnB2jedF	0	US	14	0	14	14	65	39	[]	Pop Electronica	R&B	Brooding	Defiant	Excited	1	2	1	1	1	1	2
spotify:user:bbe955cac1a86d81eb6290e85def7e3c:playlist:1XlJJah6ti5ifQaPgfo2ju	0	US	232	11	194	188	12	2	["party"]	Dance & House Electronica	Rap	Excited	Aggressive	Energizing	0	0	0	0	1	1	2
spotify:user:45022317c2957f16e05228c15ca8e921:playlist:2Qnh29VuJ8r4oefNWCv9mo	1	US	90	0	71	77	1041	934	["cry", "heart"]	Pop Rock	Indie Rock	Yearning	Empowering	Sophisticated	47	46	1	2	6	6	4
spotify:user:f12d6f489c35c5510ed13dd0d4ca1540:playlist:2k3IZE08sh50uaZ9yq0ui4	0	US	47	0	45	44	30	0	["twerk"]	Rap Pop	Electronica	Defiant	Energizing	Aggressive	0	0	0	1	3	2	0
spotify:user:ce53e4b60cce53320e37fa90177872d8:playlist:69VW6s7WEx4vwxiwp4GMo	0	US	13	0	13	13	69	65	["fall"]	Rock Alternative	Country & Folk	Rowdy	Defiant	Lively	0	4	1	1	2	0	0
spotify:user:2a5c0e4d31ccb88833eff22593cc3978:playlist:46lB52sNACmGjJUnYo3ZuG	1	US	37	0	28	29	231	0	["young", "rich", "niggas"]	Rap R&B	-	Defiant	Cool	Sensual	26	1	1	6	26	2	67
spotify:user:c76a4b8b36a58dee5cb69bc75e8e03e4:playlist:7js1EPQLyFRWbBtXS2YQgf	0	US	34	0	15	15	34	1	["betta", "move"]	Electronica	Dance & House	Indie Rock	Excited	Energizing	1	0	0	0	2	2	1
spotify:user:4328009308385bfc597106d1d4c32f31:playlist:1AMli4cjFOujs2tEG0lIo1	0	US	16	0	13	14	122	83	["happy"]	Indie Rock	Alternative	Rock	Excited	Yearning	1	0	0	1	2	1	2
spotify:user:b5bd00f02b23cf2f02b3f893a168a2d0:playlist:3GJBT01naXy4om2wJQesiM	0	US	153	0	96	117	226	217	["birthday", "party"]	Rock Metal	Alternative	Rowdy	Excited	Urgent	0	0	0	2	3	4	1
spotify:user:28ddb607228dc0e88eb33b5c145a709:playlist:66l4CeQpiKyrRXTBvC4Ev9	0	US	14	0	13	13	47	22	["new", "turkish"]	Pop Rock	Traditional	Sophisticated	Brooding	Upbeat	0	0	0	2	5	3	11
spotify:user:2db8d7519f4c9a1b6b35f70ae70fe4fa:playlist:2BABAfZCTL2L1DxYRTDEng	0	US	15	0	15	15	32	27	["yup"]	Pop Indie Rock	Rock	Empowering	Excited	Easygoing	1	0	0	1	2	1	2
spotify:user:d5dcfbe3a4c0112e9dfe36ce2edf94f7:playlist:07YYyV2y31iIrxy0DncLA	0	US	28	0	28	28	7	4	["bitch", "whiskey"]	Rap Pop	Dance & House	Aggressive	Defiant	Cool	0	1	1	1	1	0	3
spotify:user:2c8a1dce25265e13dedb901e3e74757e:playlist:2s5ciqy0S54YvlRhUtN6jE	0	US	600	20	2	246	164	144	["email", "music"]	Latin Pop	Rap	Energizing	Defiant	Excited	2	15	7	1	1	2	6
spotify:user:9eaf2af811849f4857817233af14ce6f:playlist:6VfGgG4sQ8wBNVJT6fng7W	1	US	242	0	55	62	956	876	["silent", "sleep"]	Alternative	Country & Folk	New Age	Peaceful	Romantic	5	1	1	2	4	5	9
spotify:user:0e0cb01eb1fe758753fb966d5aaaf8d7d:playlist:7KumgwGcvKv9pvJtc9deeN	0	US	252	0	182	198	156	84	["fairytales"]	Pop Alternative	Rock	Yearning	Empowering	Upbeat	0	2	1	1	2	4	14
spotify:user:426e90e2bface02fc702036184af3fa:playlist:0ECIakr8EittCMEjTSSK0A	0	US	339	0	92	85	224	221	["girls", "songs"]	Pop R&B	Children's	Empowering	Excited	Yearning	1	15	10	1	1	2	4
spotify:user:bbc9b8176199e39f2b8fb72a36254a60:playlist:7173BpBFDZNbcBvf6ZtMls	0	US	100	0	20	20	65	3	["ctf", "september"]	Indie Rock	Pop	New Age	Urgent	Empowering	1	0	0	1	2	1	2



Some Features in the Data

- **STREAMS, STREAM30S, MONTHLY OWNER STREAM30S**

About user engagement within the playlist.

- **TRACKS, LOCAL TRACKS*, ARTISTS, ALBUMS**

Quantitative descriptions of the playlist.

*Local Tracks indicate a known premium subscription.

- **DAUS, WAUS, MAUS, USERS, SKIPPERS**

About user engagement on the platform, and sometimes within the playlist.

- **MOODS, TOKENS, GENRES**

Qualitative descriptions of the playlist. Tokens come from the user. Moods & Genres come from Spotify.



Possible Success Metrics

- **MONTHLY OWNER STREAM30S**

Number of streams over 30 seconds by playlist owner this month

- **MAUS**

Number of Monthly Active Users, i.e. users with a stream over 30 seconds from playlist in past month

- **MONTHLY STREAM30S**

Number of streams over 30 seconds this month

- **STREAMS**

Number of streams from playlist today



Success Metric for User Playlist

SUMMARY USER AND FINANCIAL METRICS

USERS (M)	Q1 2018	Q4 2018	Q1 2019	% Change	
				Y/Y	Q/Q
Total Monthly Active Users ("MAUs")	173	207	217	26%	5%
Premium Subscribers	75	96	100	32%	4%
Ad-Supported MAUs	102	116	123	21%	6%

What is success?

When a playlist gets used! The more, the better!

Considerations

There are two types of playlists in the dataset: user-created and Spotify official.

For this analysis, we'll focus on user-created playlists.

Success Metric

Number of streams over 30 seconds by playlist owner this month



Engineered Features

● NATURAL LANGUAGE SENTIMENT SCORING

Token Sentiment Score, Mood Sentiment Score (all 3 moods averaged), Token Sentiment Absolute Value Score (intensity proxy)

● PERSONAL AGGREGATIONS

Token count (effort proxy), Genre count (variation in playlist), Owner playlist count

● RATIOS

Tracks per Artist (variation), Tracks per Album, Local-to-Total Tracks Ratio (discovery), Stream 30s-to-Streams (quality of time), Stream 30s-to-Monthly Streams (day-to-month ratio), Monthly Owner-to-Total Streams (intimacy)

● BINARIES

Owner has multiple playlists, Owner known to be Premium subscriber



Spotify®

Data Preparation



GETTING THE DATA READY

Outliers

- Spotify official playlists must be taken out of the data. There are 399 of them.
- For linear regression, consider data within 3σ .

Dummy Variables

- Computers can't make sense of categorical variables, so they must be made binary.
- Avoid dummy variable trap.

Sentiment Analysis

- Python has very powerful natural language processing libraries, such as Vader Sentiment. It can score a word's positivity.

Look for Nulls

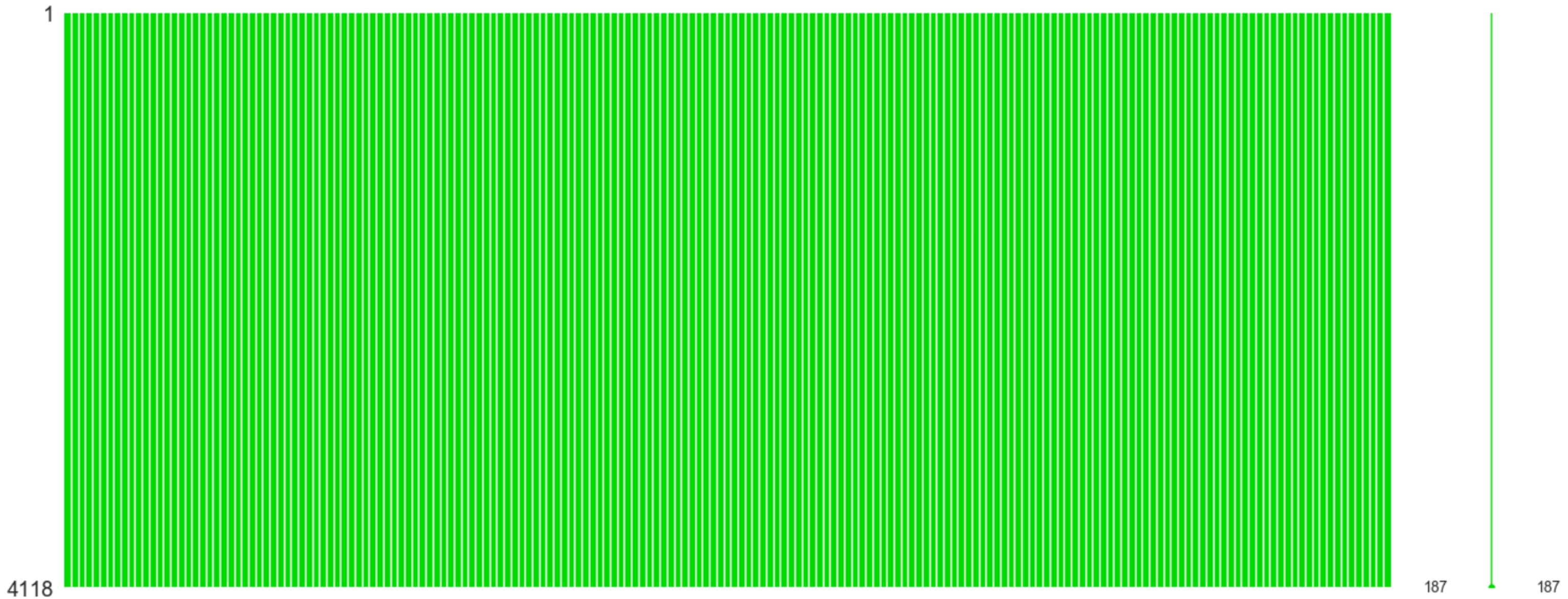
- Most data analysis does not do well with missing values. Consider eliminating the nulls, interpolating, or mean-filling.



ABSOLUTELY NO NULLS!

```
In [ ]: #checking for completeness  
msno.matrix(spotify_lite_numeric, color = (.0,.85,.0))
```

```
Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1adb7470>
```



WORD SENTIMENT ANALYSIS

	Lead Mood	Mood Sentiment Score
0	Aggressive	-0.1531
1	Brooding	0.0258
2	Cool	0.3182
3	Defiant	-0.2263
4	Easygoing	0.3182
5	Empowering	0.0000
6	Energizing	0.4588
7	Excited	0.3400
8	Fiery	-0.3400
9	Gritty	0.0000
10	Lively	0.4404



Reasonable minds can disagree on how the Vader Sentiment engine scores a given word, but, as we shall see, it becomes very powerful when analyzing word aggregations.



MOST NEGATIVE TOKEN COMBOS

	Playlist ID	Tokens	Token Sentiment Score
0	1awu4P3xtUYkkpkTn2H6fl	["boss", "ass", "bitch", "bitch", "bitch", "bi...]	-0.9623
1	1v8iwl5ehBWHgTfpJHYu9f	["killer", "killed", "killer", "killed"]	-0.9618
2	5rnoJcL2ssx56EgOHlKxEs	["killer", "killed", "killer", "killed"]	-0.9618
3	5uFFoFd935fHpsjNyS0gjo	["punk", "punk", "goes", "black", "death", "he..."]	-0.9413
4	3eZxZLBMuncTetclqdFu17	["yung", "based", "boy", "shit", "feel", "homi..."]	-0.9403
5	561AfqMPTVuCcBLHf7GqQB	["fuck", "bad", "bad", "bitch"]	-0.9360
6	3ABdz7AF9UGsf0p6ls0RTQ	["dope", "ass", "trick", "ass", "sucka", "ass"...]	-0.9360
7	3XYRrF0FdmbNYJzVZIB9ml	["hell", "yeah", "bitch", "go", "hard", "hell"...]	-0.9217
8	7eSo6eg25wtRnMQeYmuMWr	["depressing", "ass", "depressed", "shit"]	-0.9186
9	6eXrG7eVailErqsyWyY3rU	["shit", "kill", "shit"]	-0.9169
10	4jyKtjc4TUEoY9tjCCg6sD	["kill", "people", "burn", "shit", "fuck", "sc..."]	-0.9153

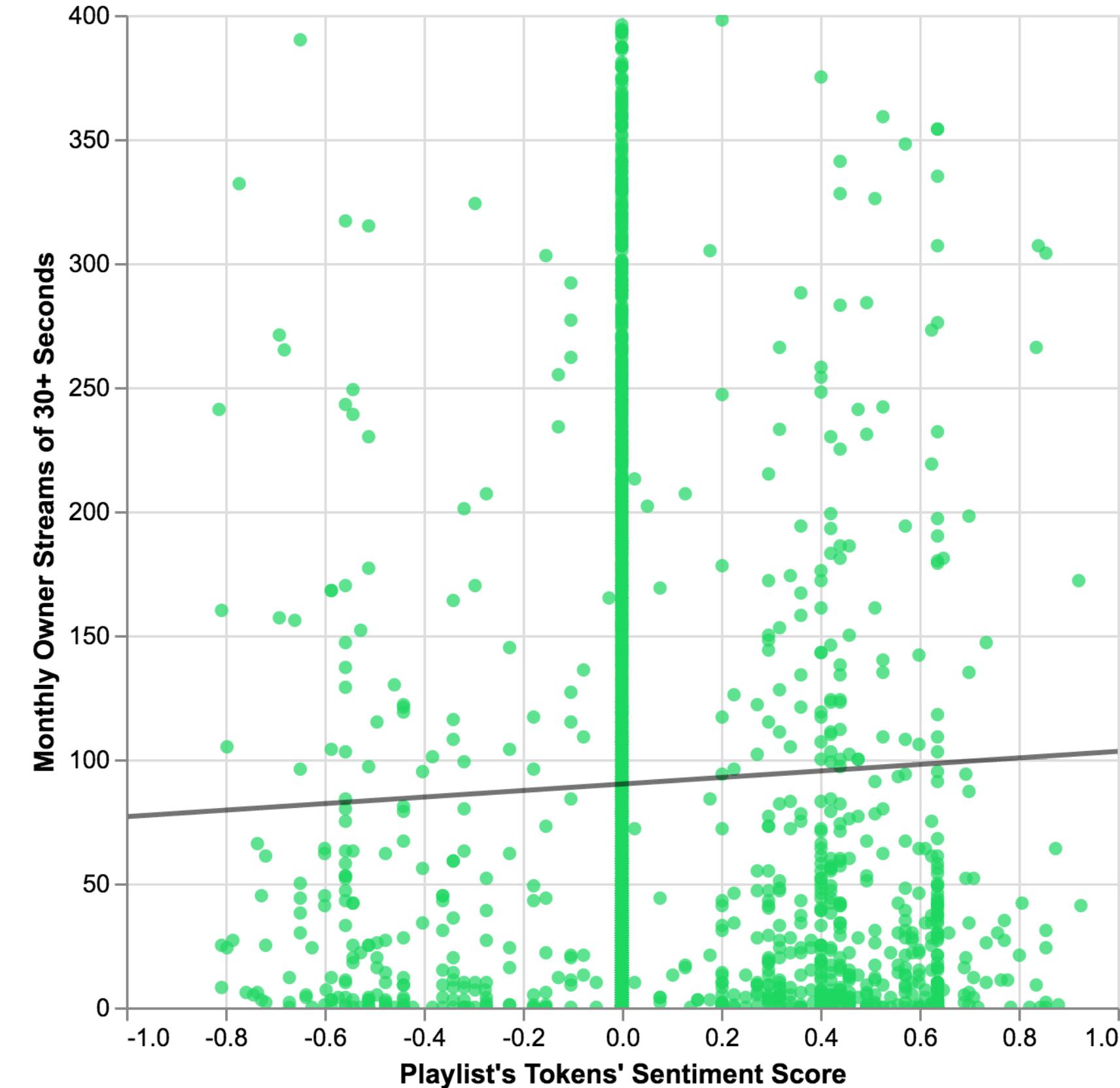
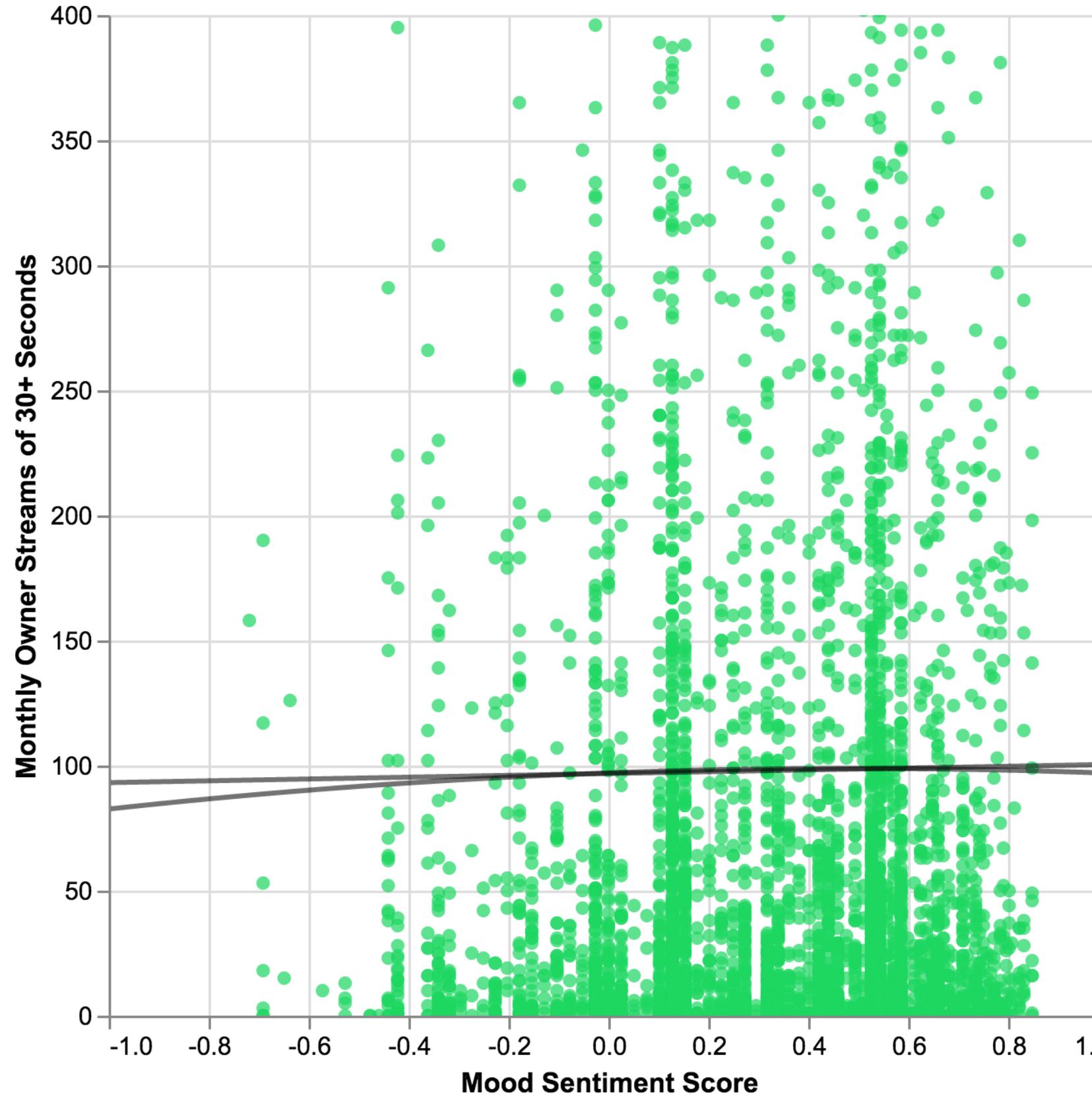


MOST POSITIVE TOKEN COMBOS

	Playlist ID	Tokens	Token Sentiment Score
0	5AWXjgKgf6dM5LB5xrmVxh	["relaxing", "piano", "greatest", "best", "lov...	0.9899
1	6NojvHGiwAWiKkXdLsENks	["relaxing", "piano", "greatest", "best", "lov...	0.9899
2	2nGaln1PPag20CxAlCeQ1G	["greatest", "best", "loved", "hymns", "spirit...	0.9856
3	1TpnPILooYYP0ZI63yYHcG	["love", "songs", "love", "songs", "deep", "lo...	0.9803
4	2vkImB9cNciyIP7aLv8fKr	["super", "happy", "funtime", "awesome", "nice...	0.9758
5	4M6n6PoCwhlG1mLMGBdHI3	["love", "songs", "piano", "songs", "romance",...	0.9682
6	7haamLtX1X711w2cc7kYt9	["hope", "anne", "muckley", "love", "like", "l...	0.9678
7	3pD9Sn1NOWHA7yYWHEDVCX	["edm", "rehab", "top", "electronic", "dance",...	0.9509
8	74WXhw6lAXJpC5SYHJqrAb	["great", "worship", "songs", "kids", "praise"...	0.9451
9	4IR6coi2zbiRrO51HRT19N	["great", "worship", "songs", "kids", "praise"...	0.9451
10	3PpbClzYP6w0xKY7qZ8IEI	["glee", "cast", "love", "like", "love", "song...	0.9442



SCORES & STREAMS - NO CLEAR TREND



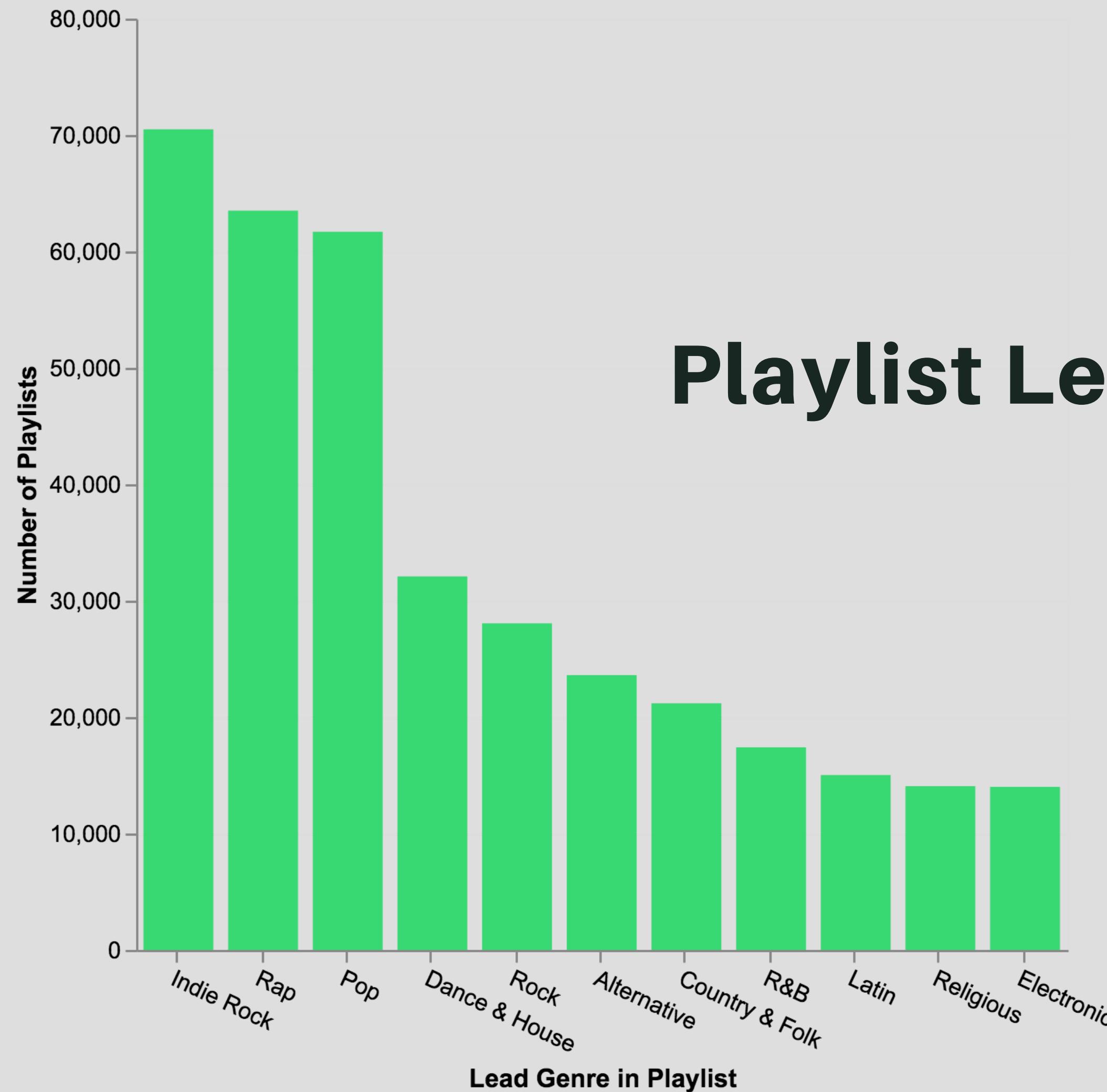


Spotify®

Exploratory Data Analysis & Visualization

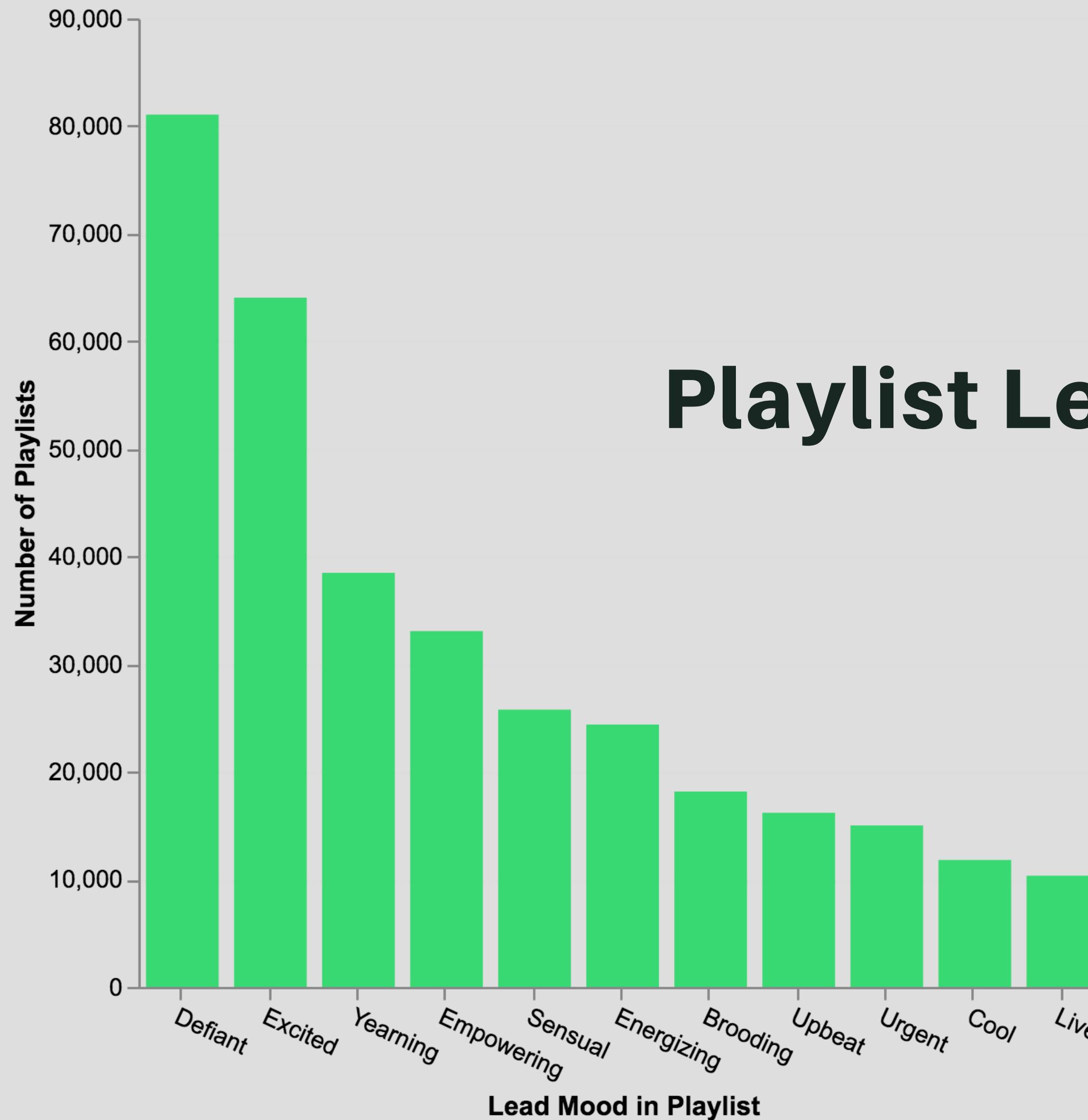


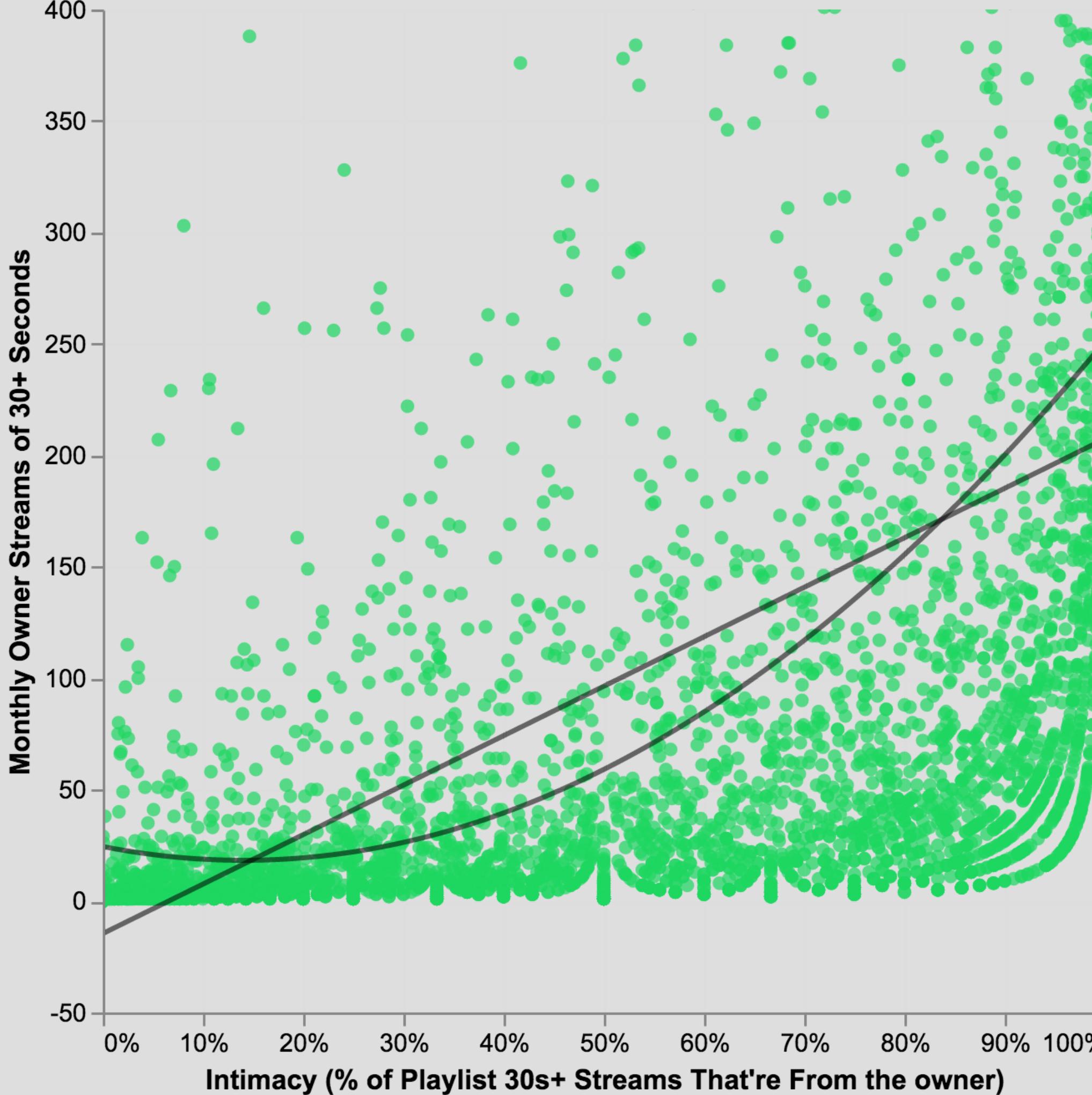
Playlist Lead Genres





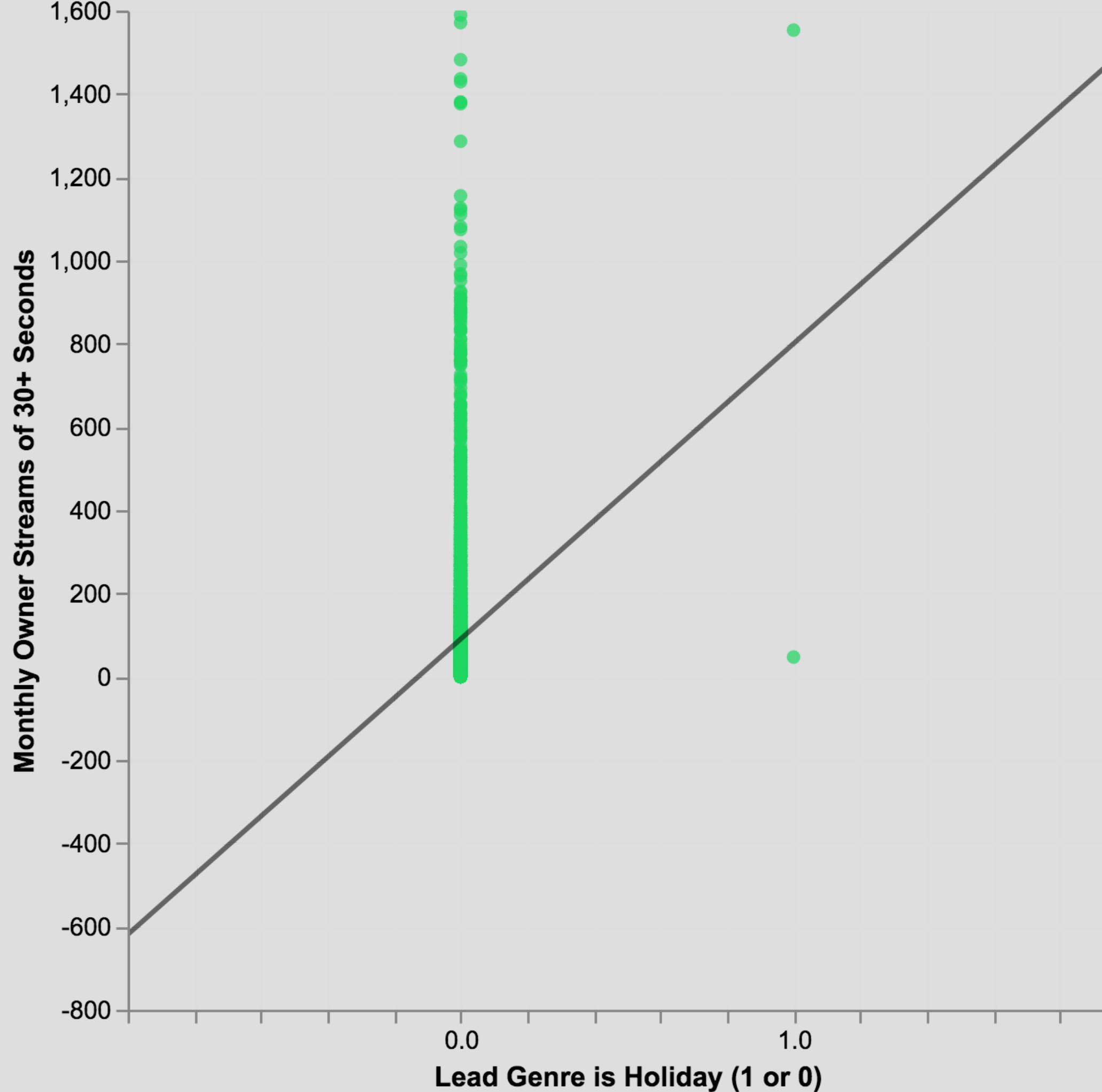
Playlist Lead Moods





Intimacy

I contend that the proportion of a playlist's monthly 30+ second streams that come from YOU demonstrates the playlist's intimacy.

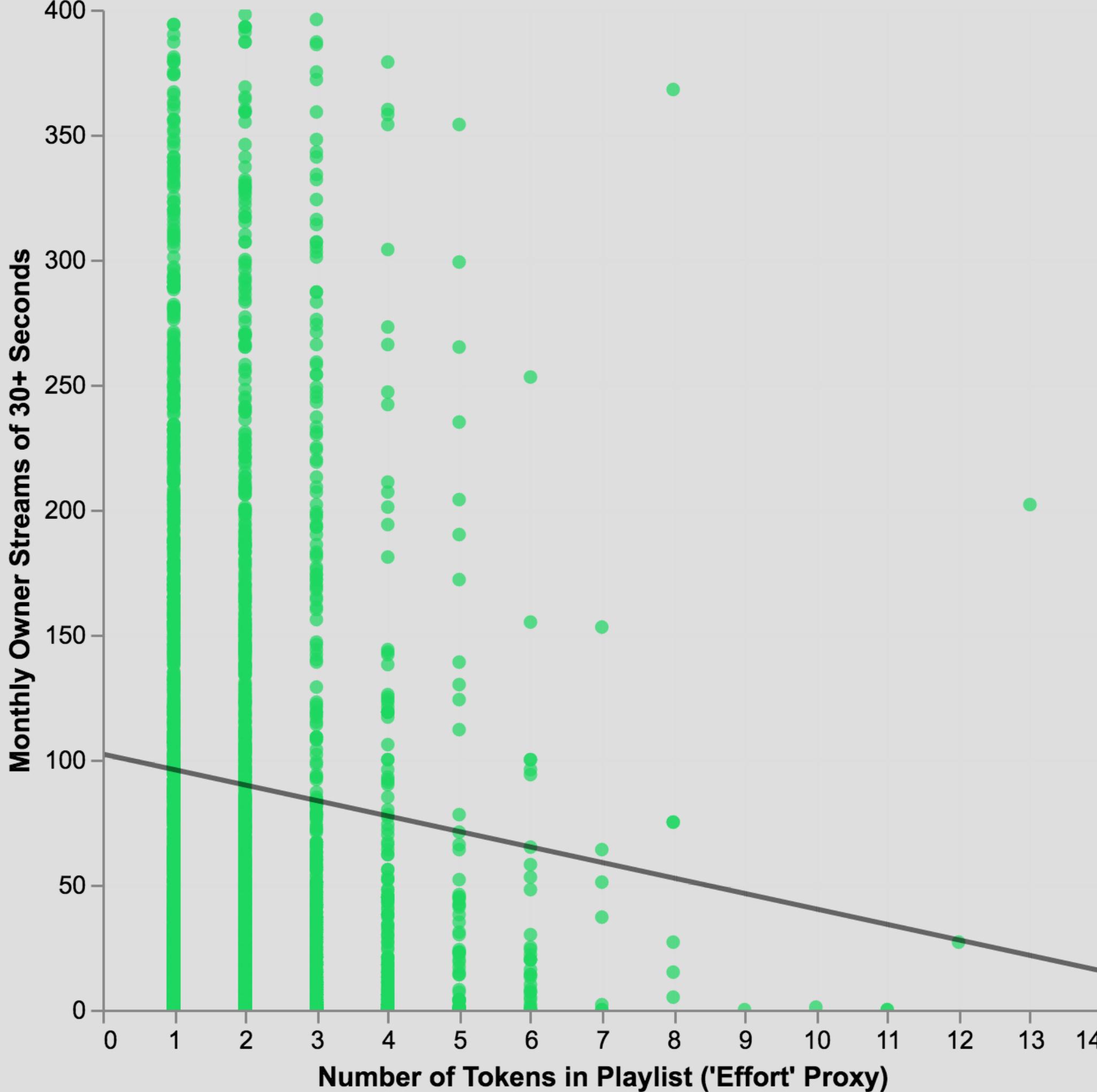


A Holiday Hunch...

Playlists with 'Holiday' as the Lead Genre are pretty rare. 158 out of 402,967 user playlists (this excludes the 399 official Spotify playlists).

But it sure looks like holiday playlists get a LOT of streams. I'd need more data to properly analyze this.





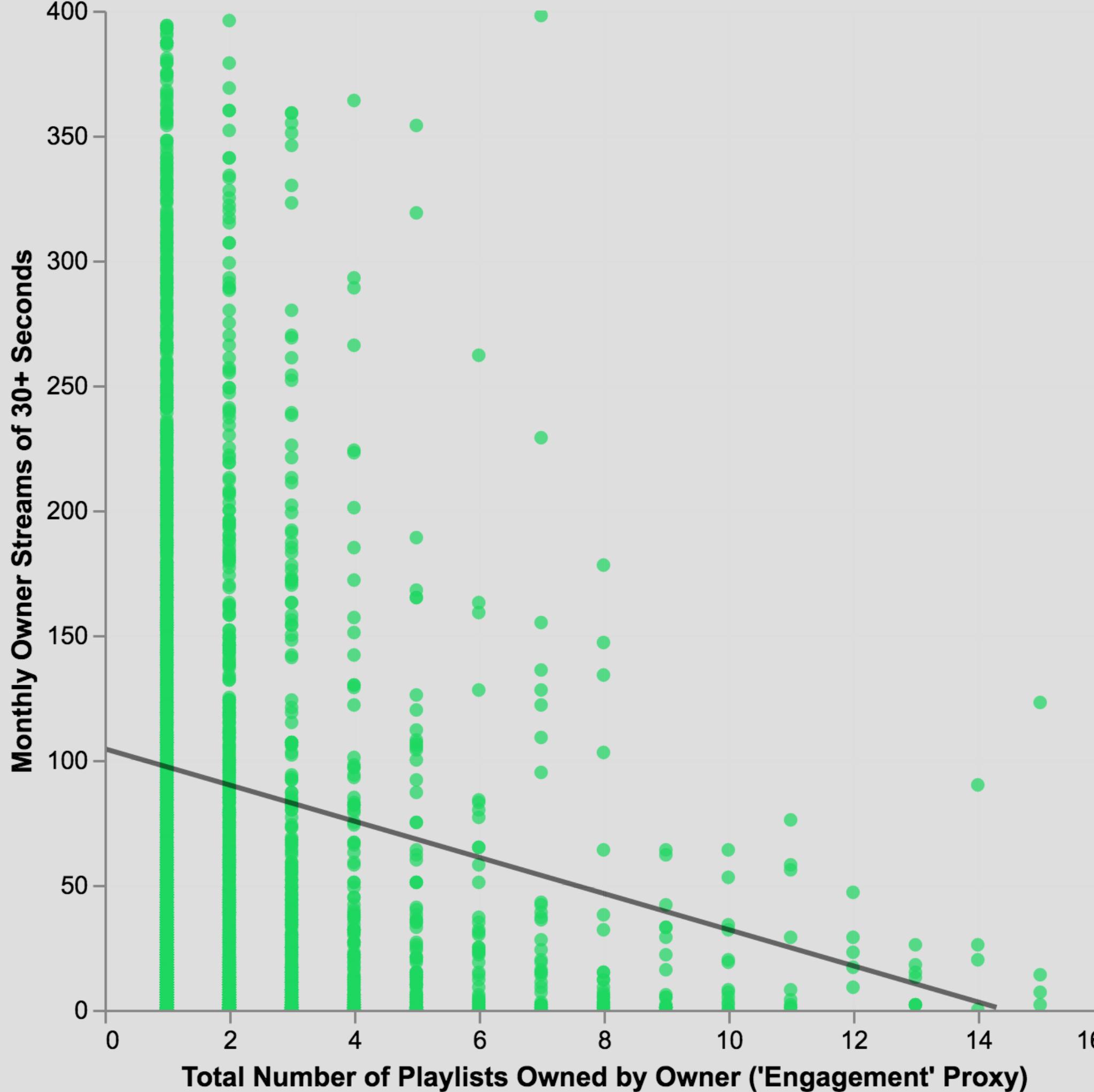
Sometimes the data surprises you

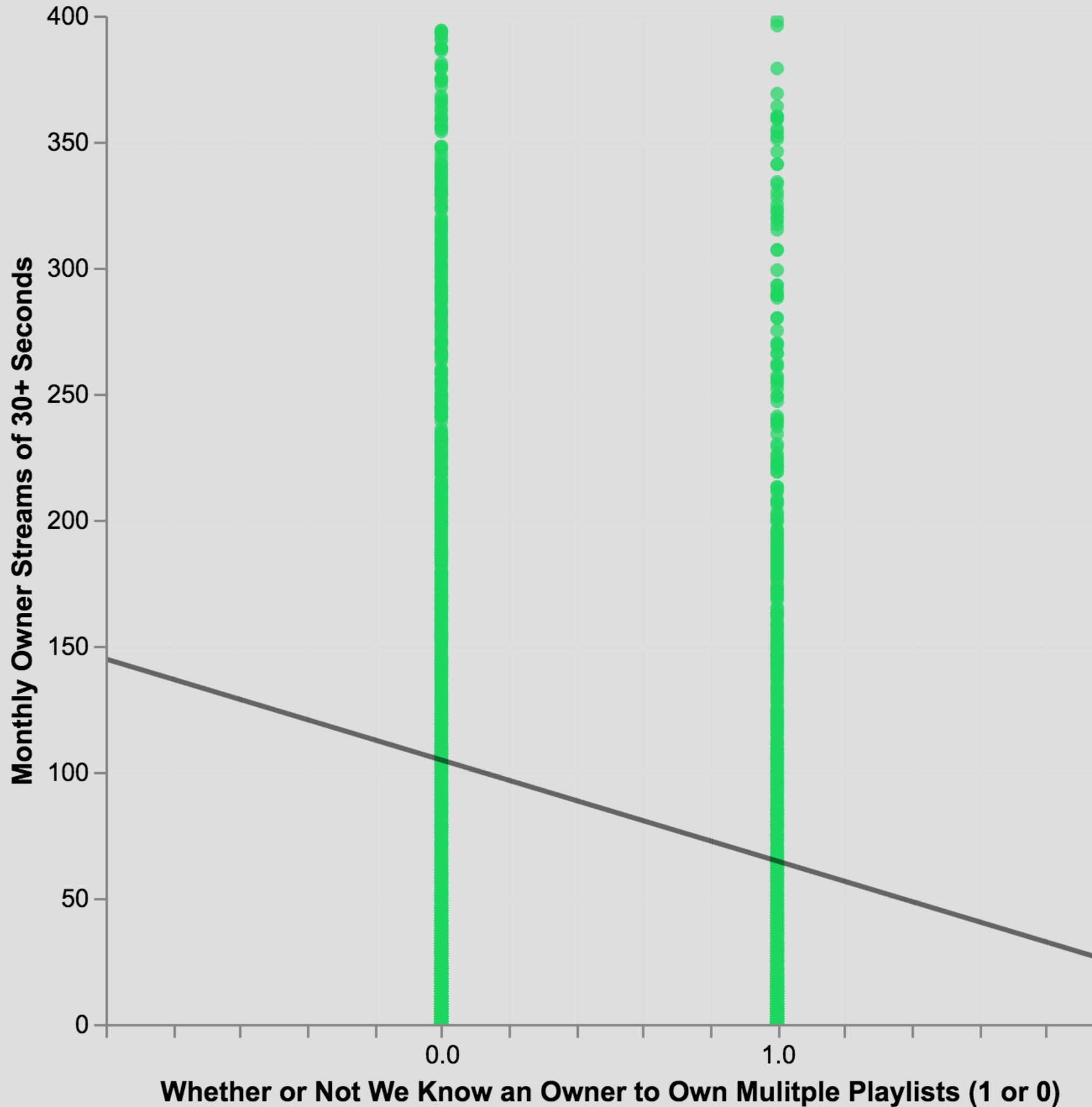
Unpacking the token data took a lot of work. But I figured it was worth it. Why? I thought the more 'effort' someone put into their playlist, maybe the longer its name would be, and then the more it'd get streamed. Wrong. The fewer the tokens, the more the streams!



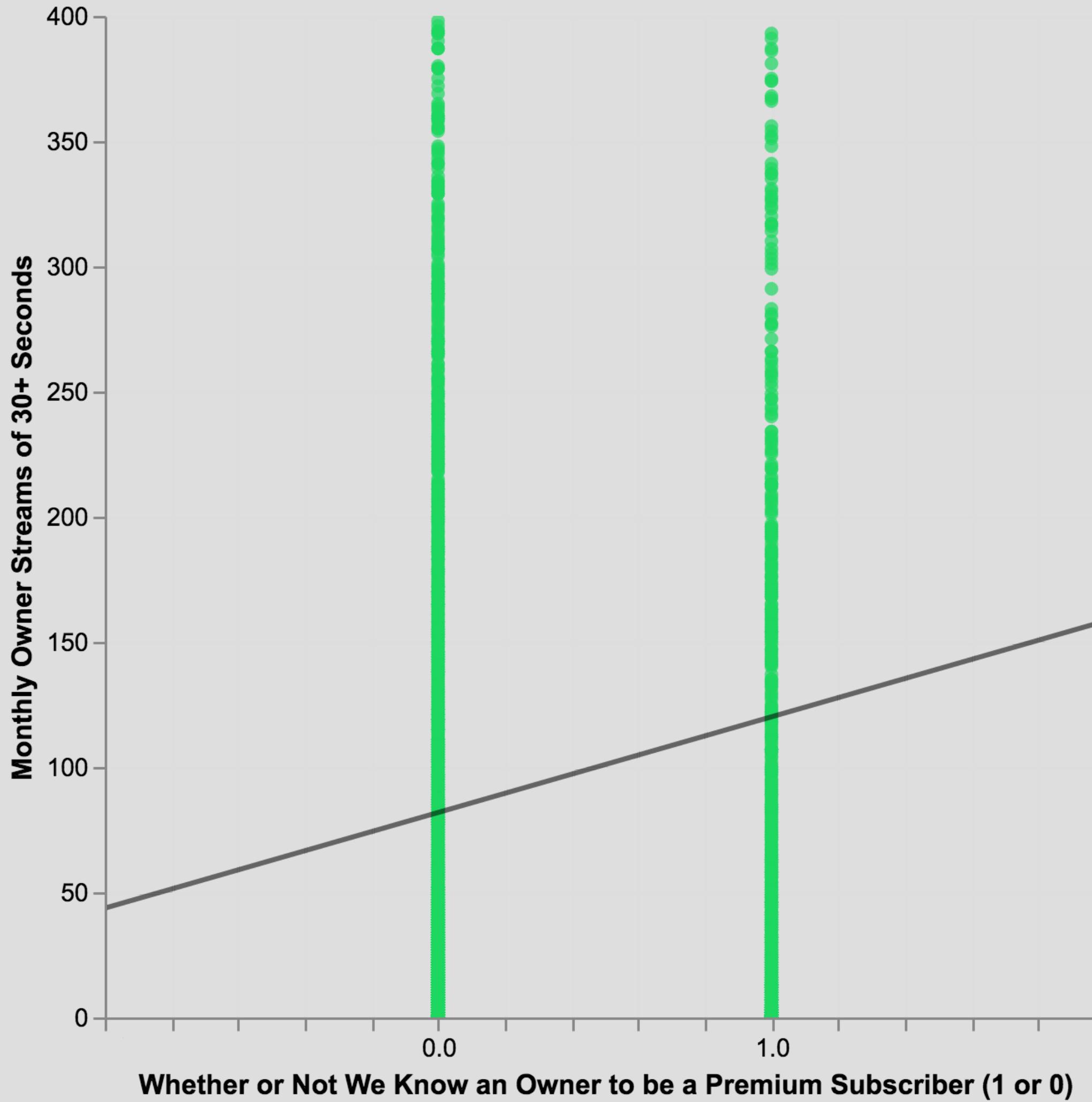
And still the data surprises you...

I thought the more playlists a user owned, the more engaged they would be on the platform, and the more times they would stream their playlists. Wrong. It looks like people who own just one or two playlists are more focused on them.





**...Having more
than one playlist
was associated
with fewer
streams.**



...But having Premium was associated with more streams.

According to Spotify's website, loading local tracks onto the platform is a Premium service. Thus, whenever a playlist had local tracks, I coded the owner as Premium. Note: a Premium user may also put zero local tracks in a playlist, so this list isn't exhaustive.



Idea 1: Intimacy. It's about you.

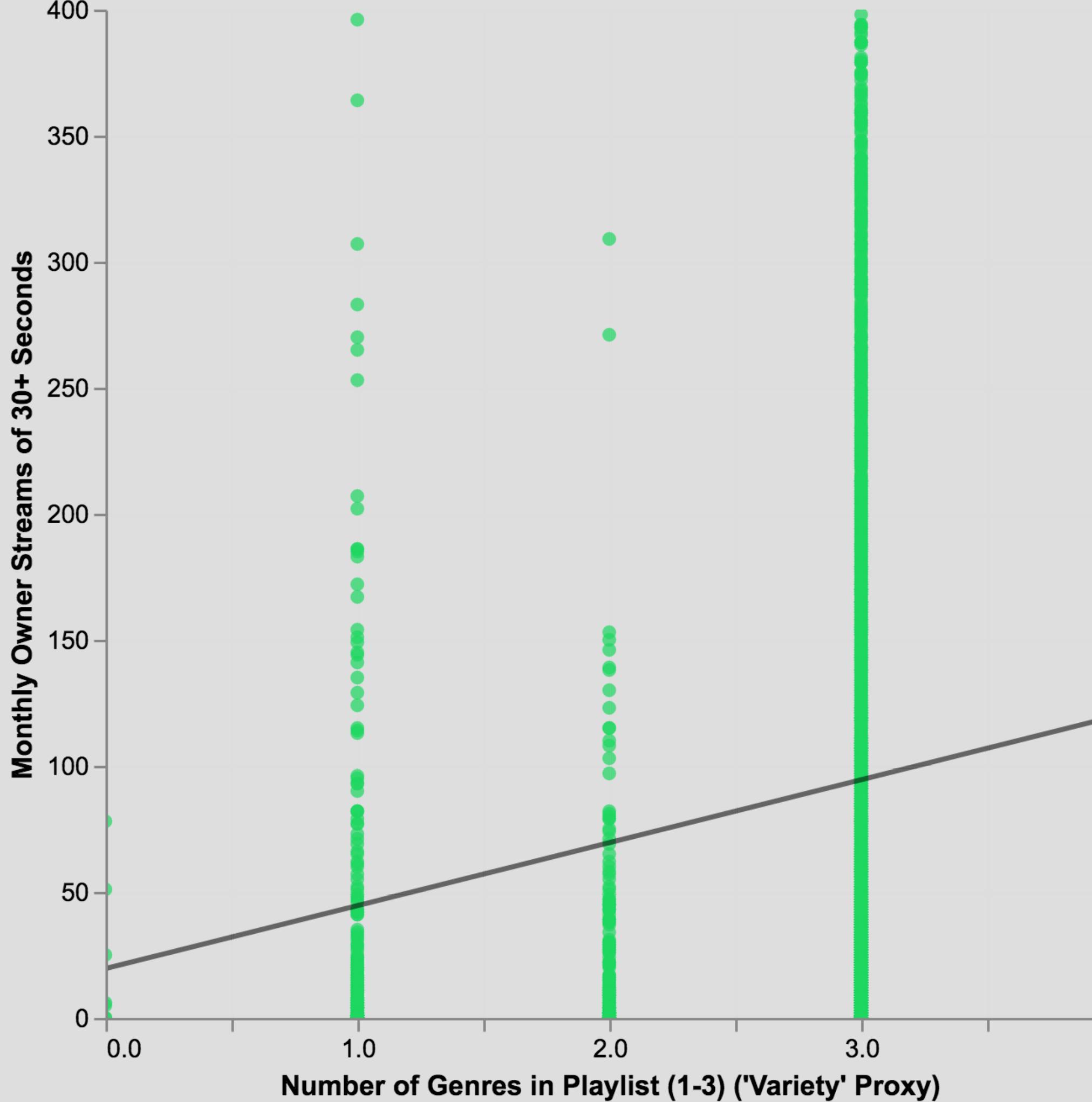
FOCUS, INTIMACY, AND PERSONALIZATION MAY BE KEYS TO HIGHLY-PLAYED USER-MADE PLAYLISTS.

THE MORE NARROWLY TAILEDOR THE PLAYLIST IS TO YOU, THE BETTER, EVEN IF THAT MEANS FEWER PEOPLE (OTHER THAN YOU) ARE STREAMING IT.



Variety is Important

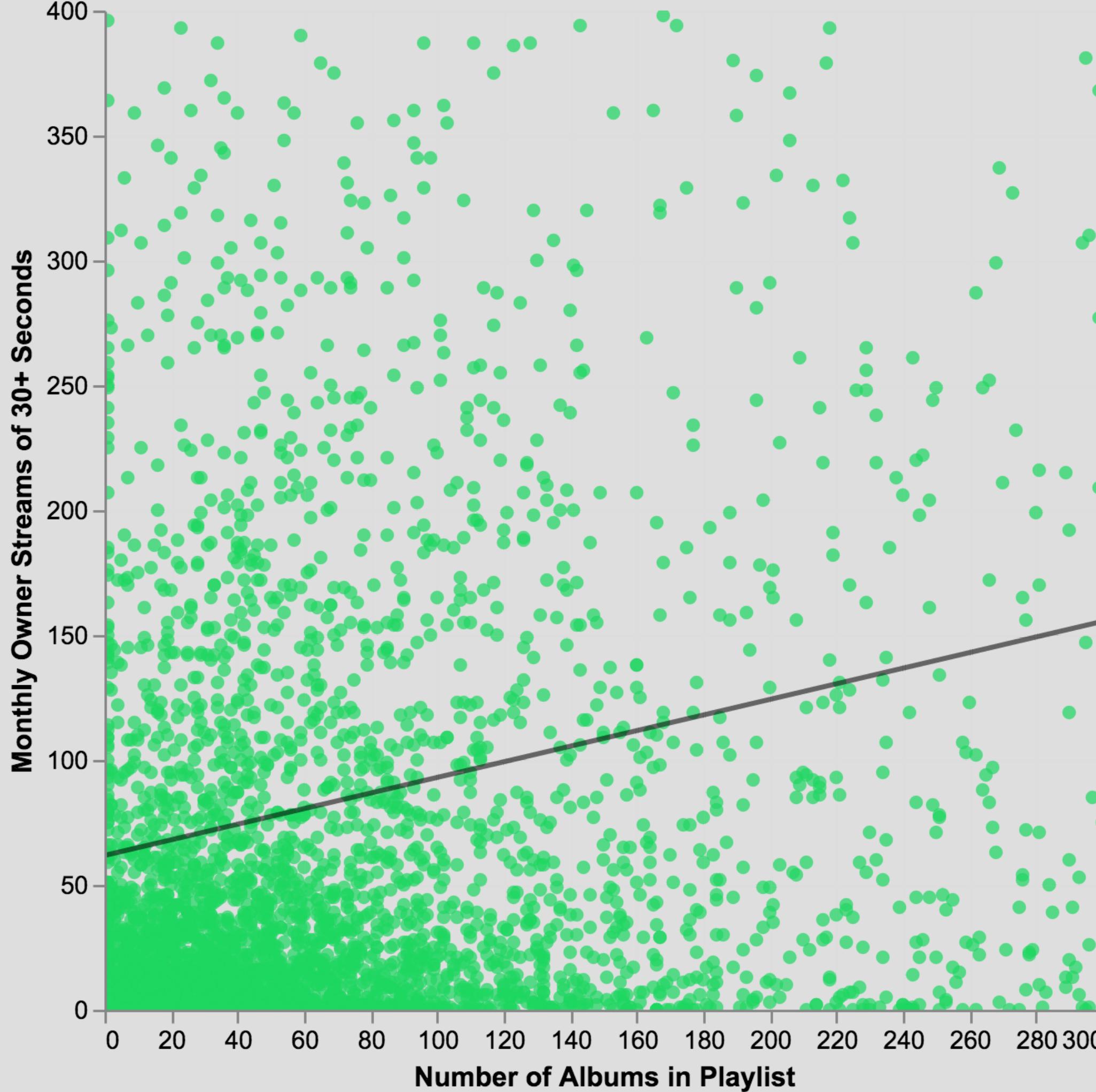
Playlists with three different genres had more streams than playlists with just one.

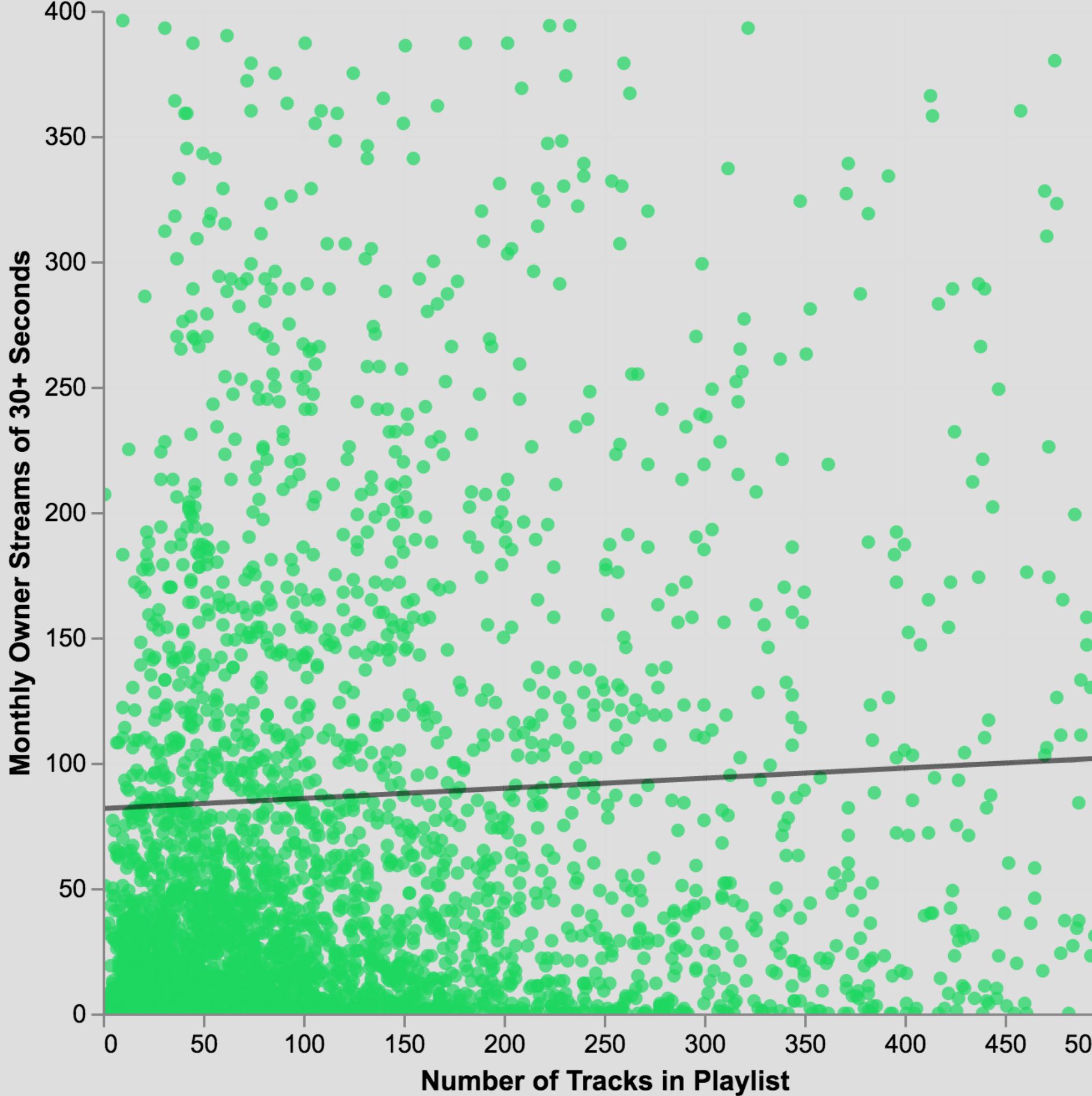




Variety is Important

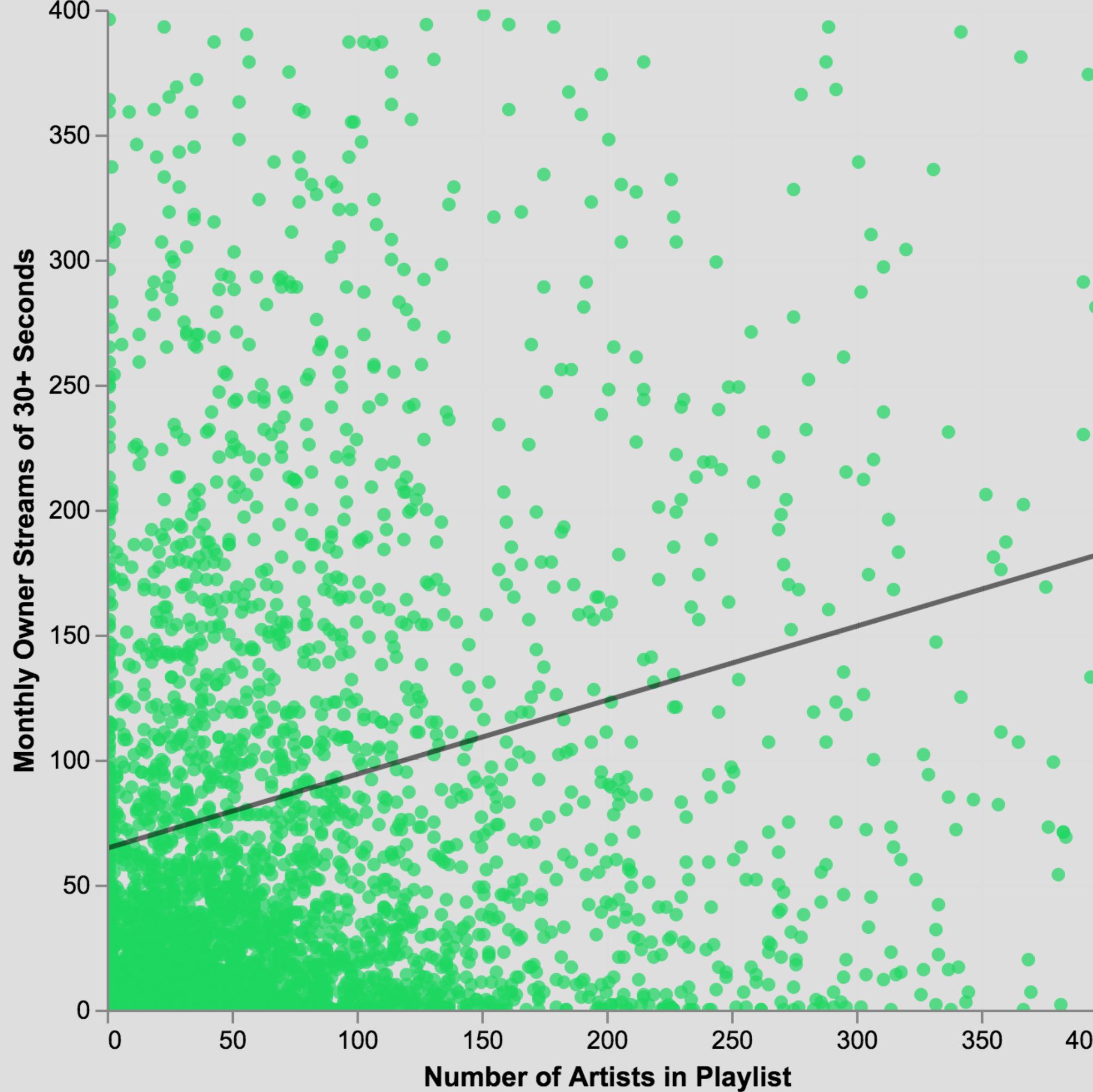
The more albums in a playlist, the better.





...But number of tracks in a playlist does not seem to matter as much.

And this makes sense for our chosen dependent variable because each time a song is played for more than 30s it counts as a stream, even if it's the same song played on repeat.



**It's variety that's
key. The more
artists, the better
too.**



Idea 2: Variety is key.

ALTHOUGH THEME AND GENRE ARE IMPORTANT,
THIS MUST NOT COME AT THE EXPENSE OF
PLAYLIST VARIETY. VARIETY BOLSTERS THE USER
DISCOVERY PROCESS.



Spotify®

Modeling

A STATISTICIAN'S PERSPECTIVE • A STATISTICIAN'S PERSPECTIVE • A STA

"Remember that all models
are wrong; the practical
question is how wrong do
they have to be to not be
useful."

GEORGE E. P. BOX (1919-2013)



A STATISTICIAN'S PERSPECTIVE • A STATISTICIAN'S PERSPECTIVE • A STA

Song Metadata & Playlists																
Spotify ID	User	Country	Length (ms)	Explicit	Popularity	Key	Mode	Artist	Title	Album	Genre	Danceability	Loudness (dB)	Tempo (BPM)	Energy	Valence
spotify:user:7310382bc047820dc5d324d2772a03	playlist:2hf0f325qqt4111vrksjz	1 US 455 2 341 381 332 321	["slaylist", "looking"] Pop Rap R&B	Defiant	Excited	Energizing	1 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:35eb3d66eb58ae6e4fd35bab46e5542e	playlist:1vhA70enKQTWZOPPTij1gX	0 US 444 0 93 114 129 110	["christian", "best"] Religious Rock	35eb3d66eb58ae6e4fd35bab46e5542e	18	13	1 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:2ac357e58edb6da0f92ff7f693680104	playlist:5ofVnH33nxzyTZ68WZ9nUU	0 US 1726 0 632 699 933 930	["lista", "mami"] Latin Pop Dance & House	2ac357e58edb6da0f92ff7f693680104	9	9	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:7dcbc955cef93e082bc3905cad5c8a1	playlist:65EFm0jcCYfyeca1H3sVUH	0 US 154 0 75 75 328 99	["new"] Dance & House Electronica	7dcbc955cef93e082bc3905cad5c8a1	68	64	2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:2b4069fcfac91052eb60c8ed0eded2fd	playlist:4FnHzHjy0NFkbg798UhKgp	0 US 19 0 18 18 15 1	["soda", "pop", "summer"] Pop Dance & House	2b4069fcfac91052eb60c8ed0eded2fd	0	0	0 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:105fe547c61aa8f9d3831dbbf0103992	playlist:15s2Hjp1gGrWbQcmMRnZS5	0 US 79 0 2 4 11 6	["night", "sax", "music"] Jazz Rock	105fe547c61aa8f9d3831dbbf0103992	0	0	0 1 3 2 1 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:29ee8e37b23c6209d07e4186f79e62f3	playlist:3RRpMk6upNjCyF7aGip8lT	0 US 268 0 7 11 16 0	["ottmar", "liebert", "collection"] Traditional	29ee8e37b23c6209d07e4186f79e62f3	0	0	0 0 2 1 0 0 0 0 0 0 0 0 0 0 0 0									
spotify:user:400e06c9a9898115013a895ab04f92ca	playlist:5fyuKwejya1d7xohj4hM2F	0 US 128 0 46 46 235 234	["music"] Pop Indie Rock	400e06c9a9898115013a895ab04f92ca	0	0	0 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:6d01fc6bf88c2a2854b59267c4f8f88d	playlist:4tt2LqgrKDz dj7KNqbLdqS	0 US 183 0 152 161 72 50	[] Electronica Indie Rock	6d01fc6bf88c2a2854b59267c4f8f88d	0	0	0 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2									
spotify:user:b93fb3c8c8dd61f6aee58db5ef6114d	playlist:5QnmGbt6bM50AW097LQY57	0 US 330 1 258 270 74 1	["zoned"] Latin Dance & House	b93fb3c8c8dd61f6aee58db5ef6114d	8	4	1 2 5 4 3 3 3 3 3 3 3 3 3 3 3 3									
spotify:user:be98b3e1dc68d33b1d19f99b2c49d2c8	playlist:7mgw19ra8MT7ryUosbJ0Kx	0 US 9 0 1 1 20 1	["empire", "sun", "walking", "dream"] Electronica	be98b3e1dc68d33b1d19f99b2c49d2c8	5	5	1 2 4 5 1 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:268f3524d209485667e3ab43ac27c29e	playlist:6jQYogkmD5VwE04fNM0W7w	0 US 70 0 44 51 67 52	["country", "music"] Country & Folk	268f3524d209485667e3ab43ac27c29e	0	0	0 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:63aaea00df3ca978e92761292a2eaa90	playlist:5XH0WG5KE0ZxtATzewUaWg	0 US 66 0 1 5 78 1	["steve", "pettit", "evangelistic", "team", "high", "price"]	63aaea00df3ca978e92761292a2eaa90	0	0	0 2 3 3 1 1 1 1 1 1 1 1 1 1 1 1									
Empowering Sophisticated	Religious	-	-	Romantic	-	-	-	-	-	-	-	-	-	-	-	
spotify:user:4386d0317126baad62457b43bff3e500	playlist:3TR37MZYLGp7tp5aEiUH0h	0 US 501 15 133 141 56 48	[] Indie Rock Pop Alternative	4386d0317126baad62457b43bff3e500	15	3	1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:64672b35946481aa3953fa492273718f	playlist:0KIQsi7z3Ghu1SAjPhJcvm	1 US 521 2 245 79 78 15	["ragga", "jungle"] Dance & House Reggae	64672b35946481aa3953fa492273718f	1	1	1 2 4 4 2 2 2 2 2 2 2 2 2 2 2 2									
spotify:user:6234eea77f077d721bc8375bc2fe505e	playlist:4bPfPAonxfMaPBPRluDo4V	0 US 76 0 47 60 157 68	["viejitas", "bonitas"] Latin Jazz	6234eea77f077d721bc8375bc2fe505e	0	0	0 2 5 3 3 3 3 3 3 3 3 3 3 3 3 3									
spotify:user:9f472ecbc51af7e92e3efd23ac1d11ce	playlist:6Pmu2P102HWmg8zCB1BPZ9	0 US 31 0 2 7 17 8	["van", "halen"] Rock	9f472ecbc51af7e92e3efd23ac1d11ce	0	0	0 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:78a6655af797a286b32d6fb67584ba00	playlist:6DICLM6DYotMGeUmJqIulf	0 US 503 7 353 386 20 1	["top", "songs"] Rap R&B	78a6655af797a286b32d6fb67584ba00	0	0	0 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2									
spotify:user:631e3296e38597c66cd488becd7	playlist:1FQCF1LBHSWPuYOCXdNKAoH	0 US 78 0 59 67 25 11	["abby"] Indie Rock	631e3296e38597c66cd488becd7	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0									
spotify:user:920995a70302b6ae67c65065cf855764	playlist:4J0FXDR0DXVujDQ3YakyJc	0 US 14 0 1 1 208 68	["miles", "davis", "sketches", "spain"] Jazz	920995a70302b6ae67c65065cf855764	0	0	0 3 6 6 0 0 0 0 0 0 0 0 0 0 0 0									
spotify:user:61b3c743795a4f3e0f6c8896b2cdf55	playlist:14Q0ucvnz6BDLcT7WsvI77	0 US 387 3 86 101 181 165	["mah", "faves"] Indie Rock	61b3c743795a4f3e0f6c8896b2cdf55	5	3	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:0c39bfe9cd637f54001b91455ddc9727	playlist:4KXYr7LK0eY6C4Pjz6h4pp	0 US 58 0 23 39 126 109	["jesus"] Religious	0c39bfe9cd637f54001b91455ddc9727	0	0	0 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:4ed73df8f60ddf6f9fd3f7feb1c412d3	playlist:1QW8VQ49ibbdwBEKYQbnBp	0 US 73 0 59 68 4	["classic", "rock"] Rock	4ed73df8f60ddf6f9fd3f7feb1c412d3	12	6	1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2									
spotify:user:43ec110e5340b84f51bfe1b1c6238dc9	playlist:5QdYjbky6Rclfn2ZDH1McJ	1 US 14 0 1 1 212 19	["danger", "doom", "mouse", "mask"] Rap	43ec110e5340b84f51bfe1b1c6238dc9	12	10	1 12 14 14 14 14 14 14 14 14 14 14 14 14									
spotify:user:4caaecbffb7be674fa8446923b55c8a3	playlist:7LPYpGmu75IJw6c0DgQ04g	0 US 40 1 39 36 23 0	["list"] Dance & House	4caaecbffb7be674fa8446923b55c8a3	0	0	0 5 8 8 8 8 8 8 8 8 8 8 8 8 8 8									
spotify:user:270ebca4873a87ddc99e7aeda8a47344	playlist:1YzPf50CqHS0BAL92EPC6c	0 US 9 0 1 1 17 0	["oliver", "swain", "big", "machine"] Country & Folk	270ebca4873a87ddc99e7aeda8a47344	0	0	0 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0									
spotify:user:283fdcc361db98115ea4632404c1563a	playlist:2kJnXLwMu10vimiKZSxeQdG	0 US 108 0 104 104 44 9	["mix"] Dance & House	283fdcc361db98115ea4632404c1563a	0	0	0 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1									
spotify:user:b50af0f4de752c5b0988829cd3a929b7	playlist:3WdZ8U6Tknr3rqfQncxQPN	0 US 108 0 104 104 44 9	Latin Pop	b50af0f4de752c5b0988829cd3a929b7	3	1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1									



1. READ IN THE DATA

```
In [ ]: import pandas as pd  
import numpy as np
```

```
In [ ]: spotify = pd.read_csv("playlist_summary_external.txt", delimiter='\t')
```

```
In [ ]: spotify.shape
```

```
Out[ ]: (403366, 25)
```

```
In [ ]: pd.set_option('display.max_columns', None)  
spotify.head(9)
```

```
Out[ ]:
```

country	n_tracks	n_local_tracks	n_artists	n_albums	monthly_stream30s	monthly_owner_stream30s	tokens	genre_1	genre_2	genre_3	mood_1
US	52	0	4	7	30	27	["ambient", "music", "therapy", "binaural", "b..."]	Dance & House	New Age	Country & Folk	Peaceful
US	131	0	112	113	112	94	["good", "living"]	Pop	Indie Rock	Alternative	Excited
US	43	0	35	36	63	0	["norte\u00f1a"]	Latin	-	-	Lively
US	27	1	27	26	154	108	[]	Dance & House	Electronica	Pop	Excited
US	52	0	47	51	230	0	["cheesy", "pants"]	Indie Rock	Alternative	Electronica	Excited
US	8	1	7	7	73	44	["aids", "walk"]	Indie Rock	Alternative	Pop	Brooding
US	8	0	7	6	9	8	["classy"]	Classical	Soundtrack	Alternative	Tender
US	33	0	5	24	256	53	["iranian", "rap"]	Rap	Soundtrack	Rock	Excited



2. TOKEN COUNT (EFFORT PROXY)

FEATURE ENGINEERING EXAMPLE

```
In [ ]: #length of tokens
token_count = []
for i in range(len(spotify.tokens)):
    length = len(spotify.tokens[i].split(","))
    token_count.append(length)
#this gives us a count of how many words (tokens) there are in a set with 1 being the lowest
#even an empty list will be a 1
```

```
In [ ]: token_count[:10]
```

```
Out[ ]: [9, 2, 1, 1, 2, 2, 1, 2, 1, 2]
```

```
In [ ]: spotify['token_count'] = token_count
```



3. GENRE COUNT (VARIETY PROXY) FEATURE ENGINEERING EXAMPLE

```
In [ ]: #now I am trying to count how many genres are listed in one playlist
genre_count = []
for i in range(len(spotify.genre_1)):
    row_genre_count = 3
    if spotify.genre_1[i] == '-':
        row_genre_count += -1
    if spotify.genre_2[i] == '-':
        row_genre_count += -1
    if spotify.genre_3[i] == '-':
        row_genre_count += -1
    genre_count.append(row_genre_count)
```

```
In [ ]: genre_count[:11]
```

```
Out[ ]: [3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 1]
```

```
In [ ]: spotify['genre_count'] = genre_count
```



4. OWNER PLAYLIST COUNT (ENGAGEMENT) FEATURE ENGINEERING EXAMPLE

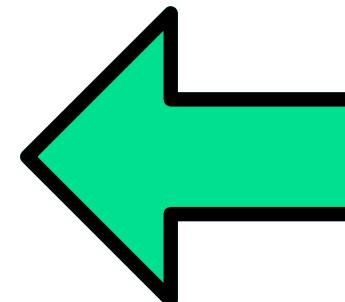
```
In [ ]: owner_frequency = spotify['owner'].value_counts().to_dict()
```

```
In [ ]: owner_frequency['47f677584fac29bc905cb4fe70c22c88'] #incredible!!!
#this is one of the times i've really seen pandas roar
```

```
Out[ ]: 3
```

```
In [ ]: owner_frequency
```

```
Out[ ]: {'spotify': 399,
'6987037f34b5cee787a1a5e8de9b2689': 48,
'a5add6d0d5fbebb01abb2fbe7e47208d': 47,
'f54f32d1c706754a70f8392aa1dbad46': 44,
'4a03268bef1505a49f8b3eced879f311': 43,
'9ff1b837bb1262d44f2194549748385a': 40,
'17d6159195b25d619e0eff98f809c90e': 40,
```



number of playlists
owned by given user

```
In [ ]: owner_playlist_count = []
for i in range(len(spotify.owner)):
    needed_string = spotify.owner[i]
    owner_playlist_count.append(owner_frequency[needed_string])
```

```
In [ ]: owner_playlist_count[:20]
```

```
Out[ ]: [2, 1, 2, 3, 1, 1, 3, 1, 1, 1, 1, 1, 4, 4, 2, 3, 1, 2, 1, 1]
```

```
In [ ]: spotify['owner_playlist_count'] = owner_playlist_count
```



5. % OWNER STREAMS (INTIMACY PROXY) FEATURE ENGINEERING EXAMPLE

```
In [  ]: #monthly_owner_stream30s_to_monthly_stream30s_ratio
monthly_owner_stream30s_to_monthly_stream30s_ratio = []
for i in range(len(spotify.playlist_uri)):
    numerator = spotify.monthly_owner_stream30s[i]
    denominator = spotify.monthly_stream30s[i]
    ratio = numerator / denominator
    monthly_owner_stream30s_to_monthly_stream30s_ratio.append(ratio)
```

```
In [  ]: spotify['monthly_owner_stream30s_to_monthly_stream30s_ratio'] =\
monthly_owner_stream30s_to_monthly_stream30s_ratio
```



6. SEPARATING USER-MADE FROM OFFICIAL SPOTIFY-MADE PLAYLISTS

```
In [  ]: spotify_user_playlists = spotify[spotify.owner != 'spotify']  
  
In [  ]: spotify_official_playlists = spotify[spotify.owner == 'spotify']  
  
In [  ]: spotify_user_playlists.shape  
Out[  ]: (402967, 199)  
  
In [  ]: spotify_official_playlists.shape  
Out[  ]: (399, 199)  
  
In [  ]: spotify_user_playlists.to_csv("spotify_user_playlists.csv", index = False)  
  
In [  ]: spotify_official_playlists.to_csv("spotify_official_playlists.csv", index=False)
```



7. SENDING THEM TO SQL :)

```
In [ ]: spotify_official_playlists.to_sql(name='spotify_official_playlists', con=engine,\n      if_exists = 'replace', index=False)
```

```
In [ ]: #and now....the big one
```

```
In [ ]: spotify_user_playlists.to_sql(name='spotify_user_playlists', con=engine,\n      if_exists = 'replace', index=False, chunksize = 25000)
```





8. THIS IS HOW I READ SQL IN PYTHON!

```
In [ ]: #this is where we are getting some info from the data set
df = pd.read_sql("""
select
playlist_sole_id as 'Playlist ID',
tokens as 'Tokens',
round(token_sentiment_score, 4) as 'Token Sentiment Score'
from sys.spotify_user_playlists2
order by 3 asc
limit 11;
""", con=db)
```

```
In [ ]: df
```

```
Out[ ]:
```

	Playlist ID	Tokens	Token Sentiment Score
0	1awu4P3xtUYkkpkTn2H6fl	["boss", "ass", "bitch", "bitch", "bitch", "bi...]	-0.9623
1	1v8iwl5ehBWHgTfpJHYu9f	["killer", "killed", "killer", "killed"]	-0.9618
2	5rnoJcL2ssx56EgOHIKxEs	["killer", "killed", "killer", "killed"]	-0.9618
3	5uFFoFd935fHpsjNyS0gjo	["punk", "punk", "goes", "black", "death", "he...]	-0.9413
4	3eZxZLBMuncTetclqdFu17	["yung", "based", "boy", "shit", "feel", "homi..."]	-0.9403
5	561AfqMPTVuCcBLHf7GqQB	["fuck", "bad", "bad", "bitch"]	-0.9360
6	3ABdz7AF9UGsf0p6ls0RTQ	["dope", "ass", "trick", "ass", "sucka", "ass"...]	-0.9360
7	3XYRrF0FdmbNYJzVZIB9ml	["hell", "yeah", "bitch", "go", "hard", "hell"...]	-0.9217
8	7eSo6eg25wtRnMQeYmuMWr	["depressing", "ass", "depressed", "shit"]	-0.9186
9	6eXrG7eVailErqsyWyY3rU	["shit", "kill", "shit"]	-0.9169
10	4jyKtjc4TUEoY9tjCCg6sD	["kill", "people", "burn", "shit", "fuck", "sc..."]	-0.9153





9. SET UP THE DUMMY VARIABLES

SK-LEARN AUTOMAGICALLY HANDLES THE DUMMY VARIABLE TRAP

```
In [  ]: spotify_dummies = pd.get_dummies(spotify[['genre_1', 'genre_2', 'genre_3', 'mood_1','mood_2','mood_3']])
```

```
In [  ]: spotify_dummies.sample(4)
```

```
Out[  ]:
```

	genre_1_-	genre_1_Alternative	genre_1_Blues	genre_1_Children's	genre_1_Classical	genre_1_Country & Folk	genre_1_Dance & House	genre_1_Easy Listening	genre_1_Electro
61563	0	0	0	0	0	0	0	0	0
331772	0	0	0	0	0	1	0	0	0
101341	0	1	0	0	0	0	0	0	0
167513	0	0	0	0	0	0	0	0	0

```
In [  ]: "Combine DataFrame objects horizontally along the x axis by passing in axis=1."
spotify = pd.concat([spotify,spotify_dummies], axis = 1)
```

```
In [  ]: spotify.shape
```

```
Out[  ]: (403366, 193)
```



10. SET UP POPULATION SAMPLE

WWW.SURVEYMONKEY.COM/MP/SAMPLE-SIZE-CALULATOR/

Calculate your sample size

Population size ? 402967	Confidence level (%) ? 99	Margin of error (%) ? 2
-----------------------------	------------------------------	----------------------------

Sample size
4,118

```
In [ ]: spotify_lite = spotify.sample(4118).reset_index()
```

```
In [ ]: spotify_lite.shape
```

```
Out[ ]: (4118, 200)
```

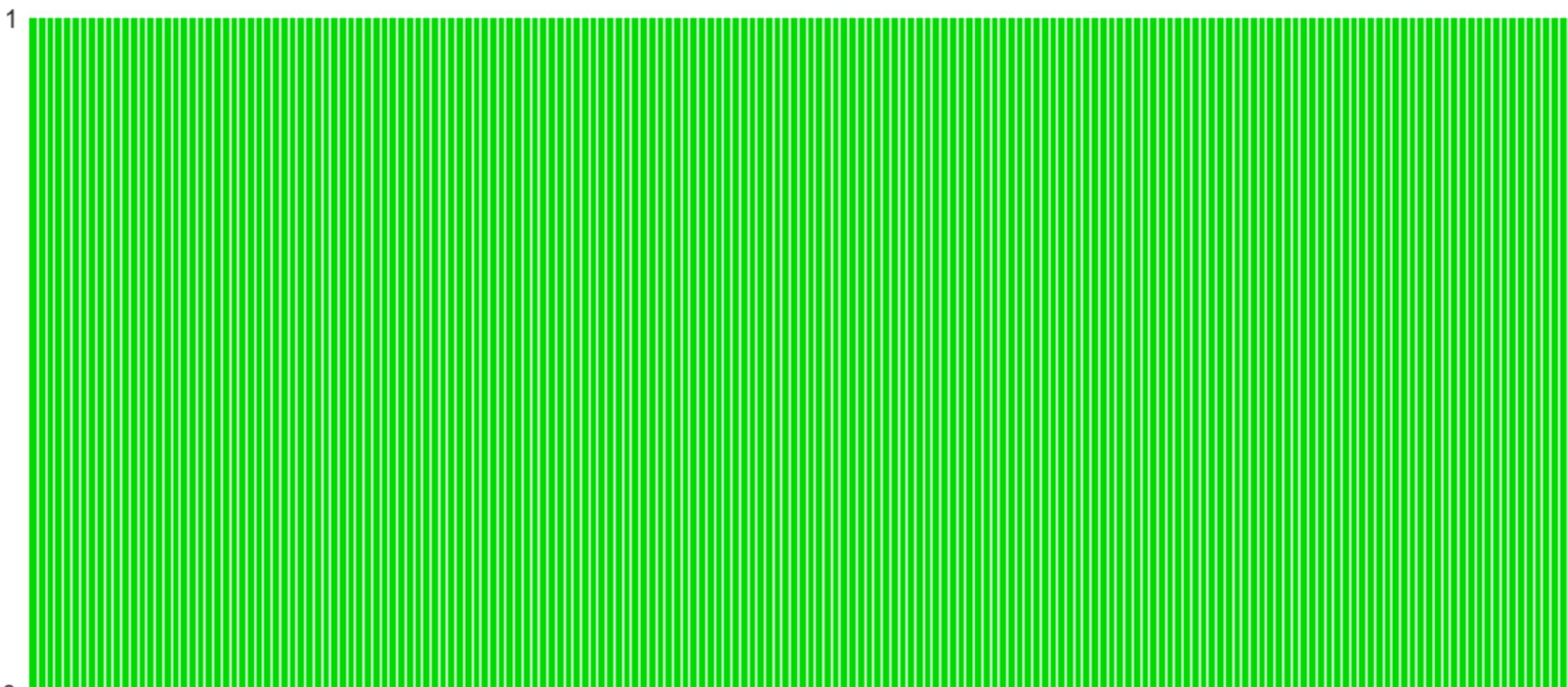


11. CHECK SAMPLE FOR NULLS

```
In [ ]: spotify_lite_numeric = spotify_lite.drop(columns = ['index','playlist_uri','owner',\n        'owner_country','tokens','genre_1','genre_2','genre_3','mood_1','mood_2',\n        'mood_3','playlist_sole_id','stream30s_to_streams_ratio'])\n#we need to also drop stream30s_to_streams_ratio b/c it has NaN's in it...
```

```
In [ ]: #checking for completeness\nmsno.matrix(spotify_lite_numeric, color = (.0,.85,.0))
```

```
Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1adb7470>
```





12. REMOVE OUTLIERS

```
In [ ]: outlier_watch_cols = ['streams', 'stream30s', 'dau', 'wau', 'mau', 'mau_previous_month', \
                           'mau_both_months', 'users', 'skippers', 'n_tracks', 'n_local_tracks', \
                           'n_artists', 'n_albums', 'monthly_stream30s', 'monthly_owner_stream30s']
#the other columns are basically metrics based on these or things where it only goes from
#say 1 to 3 (as in the genre count one)
for i in range(len(outlier_watch_cols)):
    spotify_lite_numeric_no_outliers =
        spotify_lite_numeric[np.abs(spotify_lite_numeric\
                                     [outlier_watch_cols[i]] - \
                                     spotify_lite_numeric[outlier_watch_cols[i]].mean()) \
                             <= (2.17*spotify_lite_numeric[outlier_watch_cols[i]].std())]
#2.17 covers basically 98.5% of all data under a normal distribution curve
#just to 'feel out' the data, I want to take out the top 1.5# of numbers
```

```
In [ ]: spotify_lite_numeric_no_outliers.shape
```

```
Out[ ]: (4014, 187)
```

```
In [ ]: #just a little intermediary check for completeness
temp_spotify_lite_numeric_no_outliers_no_na = spotify_lite_numeric_no_outliers.dropna()
temp_spotify_lite_numeric_no_outliers_no_na.shape
```

```
Out[ ]: (4014, 187)
```



13. SPLIT FEATURES & INDEPENDENT VAR.

THIS IS FOR INITIAL EXPLORATION BEFORE TRAIN/TEST SPLIT

```
In [ ]: #the independent variables (features):
features = spotify_lite_numeric_no_outliers.drop(columns = ['monthly_owner_stream30s'])
#the dependent variable (y):
y = spotify_lite_numeric_no_outliers.monthly_owner_stream30s
```

Now we begin the process.



14. ANY STATISTICAL SIGNIFICANCE?

```
In [ ]: regressor_OLS.summary()
```

```
Out[ ]: OLS Regression Results
```

Dep. Variable: monthly_owner_stream30s **R-squared:** 0.412

Model: OLS **Adj. R-squared:** 0.385

Method: Least Squares **F-statistic:** 15.02

Date: Wed, 31 Jul 2019 **Prob (F-statistic):** 5.85e-321

Time: 13:01:06 **Log-Likelihood:** -22940.

No. Observations: 4014 **AIC:** 4.624e+04

Df Residuals: 3834 **BIC:** 4.737e+04

Df Model: 179

Covariance Type: nonrobust

coef	std err	t	P> t	[0.025	0.975]
------	---------	---	------	--------	--------

0.0421	0.607	0.076	0.707	12.671	20.157
--------	-------	-------	-------	--------	--------



15. FEATURE SELECTION

This begins the sklearn portion of the show.

```
In [ ]: most_relevant_features = ['stream30s', 'dau', 'wau', \
    'mau', 'mau_both_months', 'users', \
    'n_artists', 'monthly_stream30s', \
    'token_sentiment_score', \
    'owner_has_multiple_playlists', \
    'genre_count', 'genre_1_children_s', \
    'genre_1_country_and_folk', \
    'genre_1_electronica', 'genre_1_punk', \
    'genre_2_electronica', 'genre_3_none', \
    'mood_1_defiant', 'mood_1_excited', \
    'mood_1_other', 'mood_2_sophisticated', \
    'mood_3_cool', 'mood_3_lively', \
    'mood_3_rowdy', 'tracks_per_album', \
    'stream30s_to_monthly_stream30s_ratio', \
    'monthly_owner_stream30s_to_monthly_stream30s_ratio']
```

```
In [ ]: most_relevant_features2 = ['streams', 'dau', 'wau', \
    'mau', 'mau_both_months', 'users', \
    'n_artists', 'monthly_stream30s', \
    'owner_has_multiple_playlists', \
    'stream30s_to_monthly_stream30s_ratio', \
    'monthly_owner_stream30s_to_monthly_stream30s_ratio']
```



16. FEATURE PRUNING

```
In [  ]: most_relevant_features3 = ['n_artists', 'n_albums', 'genre_count', \
    'monthly_owner_stream30s_to_monthly_stream30s_ratio', \
    'token_count', 'owner_playlist_count', \
    'owner_known_to_be_premium']
```

```
In [  ]: most_relevant_features4 = ['n_albums', 'genre_count', \
    'monthly_owner_stream30s_to_monthly_stream30s_ratio', \
    'owner_playlist_count']
```



17. THE CHOSEN ONES

```
In [ ]: regressor_OLS = sm.OLS(endog = y, exog = chosen_features).fit()  
regressor_OLS.summary()
```

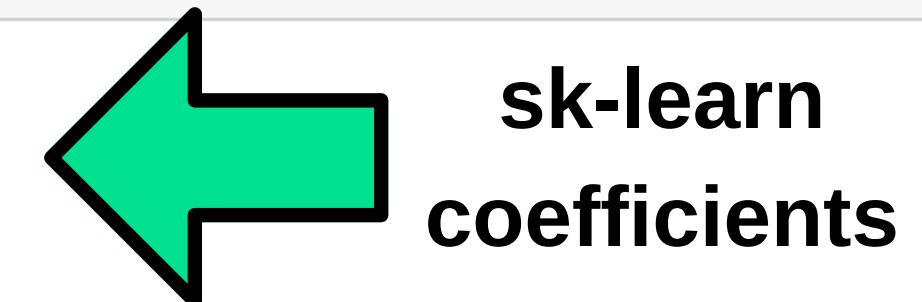
Out[]: OLS Regression Results

Dep. Variable:	monthly_owner_stream30s	R-squared:	0.533				
Model:	OLS	Adj. R-squared:	0.533				
Method:	Least Squares	F-statistic:	1146.				
Date:	Wed, 31 Jul 2019	Prob (F-statistic):	0.00				
Time:	15:50:31	Log-Likelihood:	-23282.				
No. Observations:	4014	AIC:	4.657e+04				
Df Residuals:	4010	BIC:	4.660e+04				
Df Model:	4						
Covariance Type:	nonrobust						
		coef	std err	t	P> t	[0.025	0.975]
	n_albums	0.1879	0.012	16.029	0.000	0.165	0.211
	genre_count	-3.5340	0.955	-3.702	0.000	-5.406	-1.662
monthly_owner_stream30s_to_monthly_stream30s_ratio	132.3355	3.735	35.427	0.000	125.012	139.659	
owner_playlist_count	-2.2438	0.593	-3.782	0.000	-3.407	-1.081	



18. TRAIN/TEST SPLIT

```
In [ ]: X_train, X_test, y_train, y_test = train_test_split\  
        (chosen_features, y, test_size = 0.2, random_state = 1)  
  
In [ ]: model = LinearRegression()  
model.fit(X_train,y_train)  
  
Out[ ]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,  
                        normalize=False)  
  
In [ ]: sorted(list(zip(most_relevant_features4,model.coef_))\  
            ,key = lambda x: abs(x[1]),reverse=True)  
  
Out[ ]: [('monthly_owner_stream30s_to_monthly_stream30s_ratio', 135.33557249728625),  
         ('genre_count', 2.874949752342296),  
         ('owner_playlist_count', -2.050225875348983),  
         ('n_albums', 0.15740051956660164)]  
  
In [ ]: y_predicted = model.predict(X_test)  
  
In [ ]: #I want to get a dataframe ready for alttari  
model_scoring = pd.DataFrame(\r  
    {'y_test': y_test, 'y_predicted': y_predicted, 'residual':\r  
     (y_predicted-y_test)},\r  
    columns=['y_test', 'y_predicted','residual']).reset_index()
```



sk-learn
coefficients



19. MODEL SCORING & CROSS-VALIDATION

```
In [ ]: r2_score(model_scoring.y_test, model_scoring.y_predicted)
```

```
Out[ ]: 0.3297546526887427
```

```
In [ ]: mean_absolute_error(y_test, y_predicted)
```

```
Out[ ]: 54.1607115248918
```

```
In [ ]: median_absolute_error(y_test, y_predicted)
```

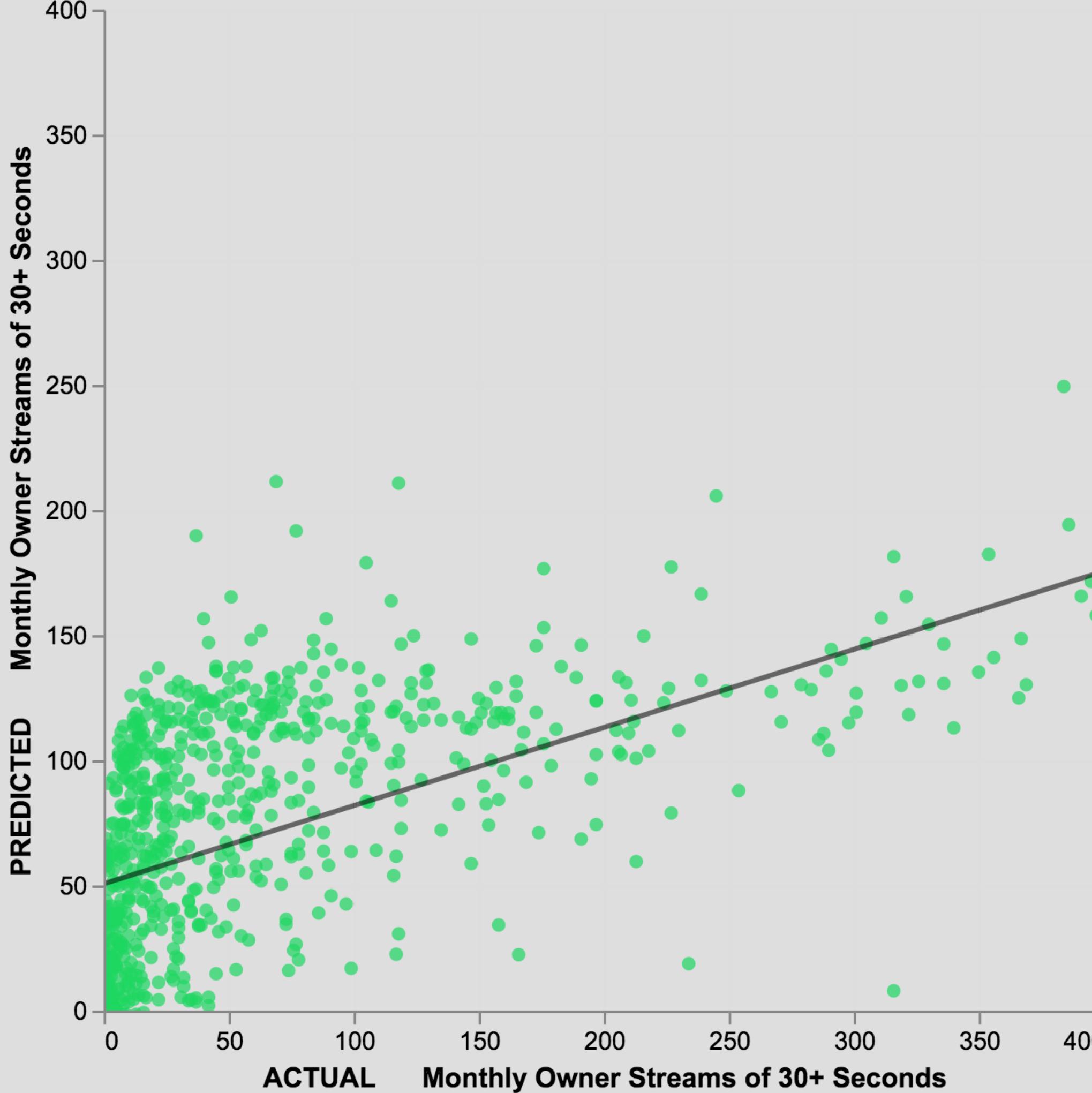
```
Out[ ]: 39.92515627211054
```

```
In [ ]: #last let's look at the cross validation
print("Cross-Validation Scoring")
print('Mean Absolute Error: {}'.format(-1*round(cross_val_score(LinearRegression(),\
chosen_features, y, cv=10, scoring='neg_mean_absolute_error').mean(),2)))
#no option for RMSE in cross val score, so looking at mean absolute error
#multiply by -1 to get a positive value to take a look at it
print('R^2: {}'.format(round(cross_val_score(LinearRegression(),\
chosen_features, y, cv=10, scoring='r2').mean(),2)))
```

```
Cross-Validation Scoring
Mean Absolute Error: 52.23
R^2: 0.3
```

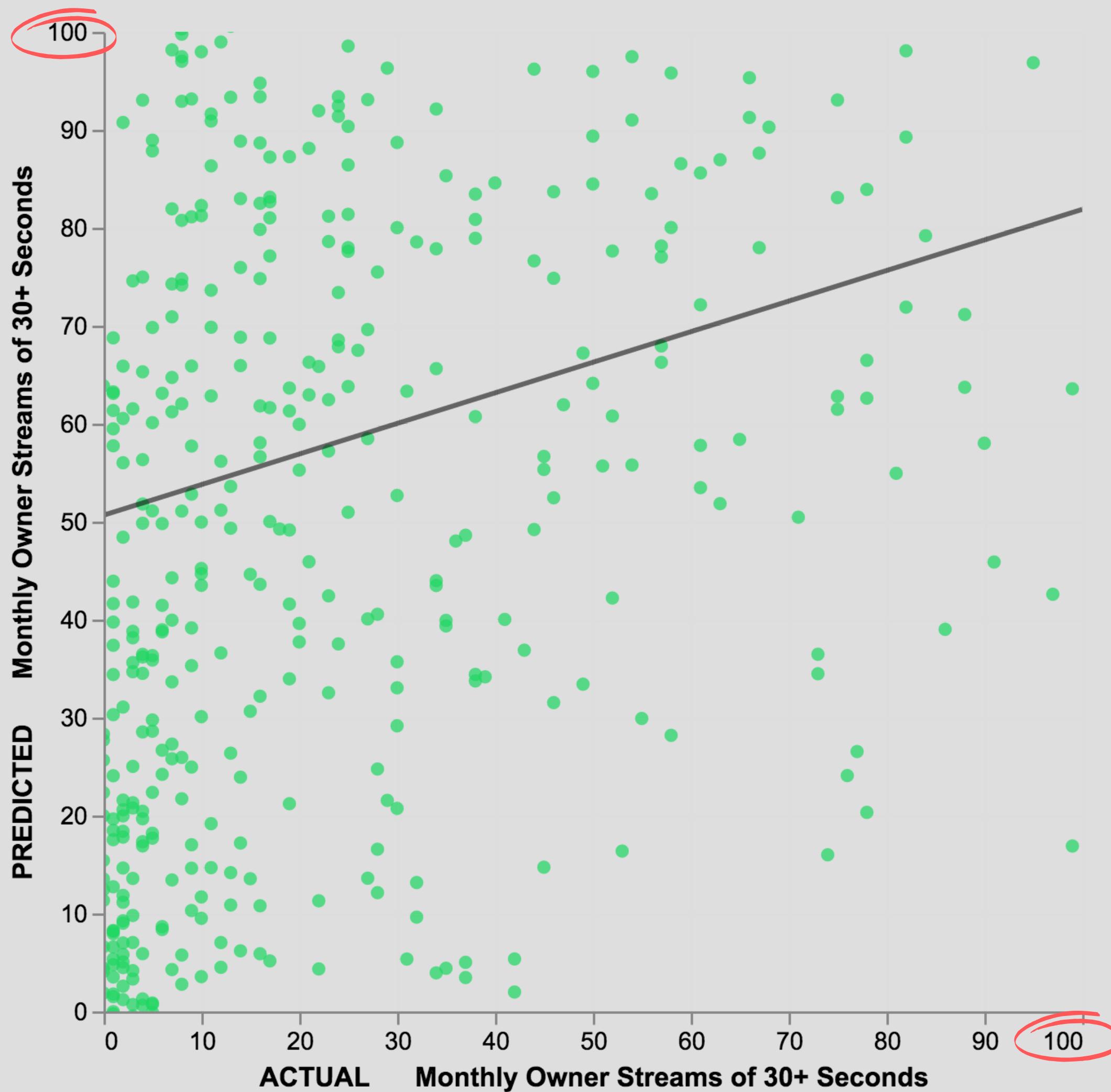


Actual (x axis) vs. Predicted (y axis)



Not bad, but gets worse as the cases get more extreme. This makes sense because I eliminated the highest values from the training set.

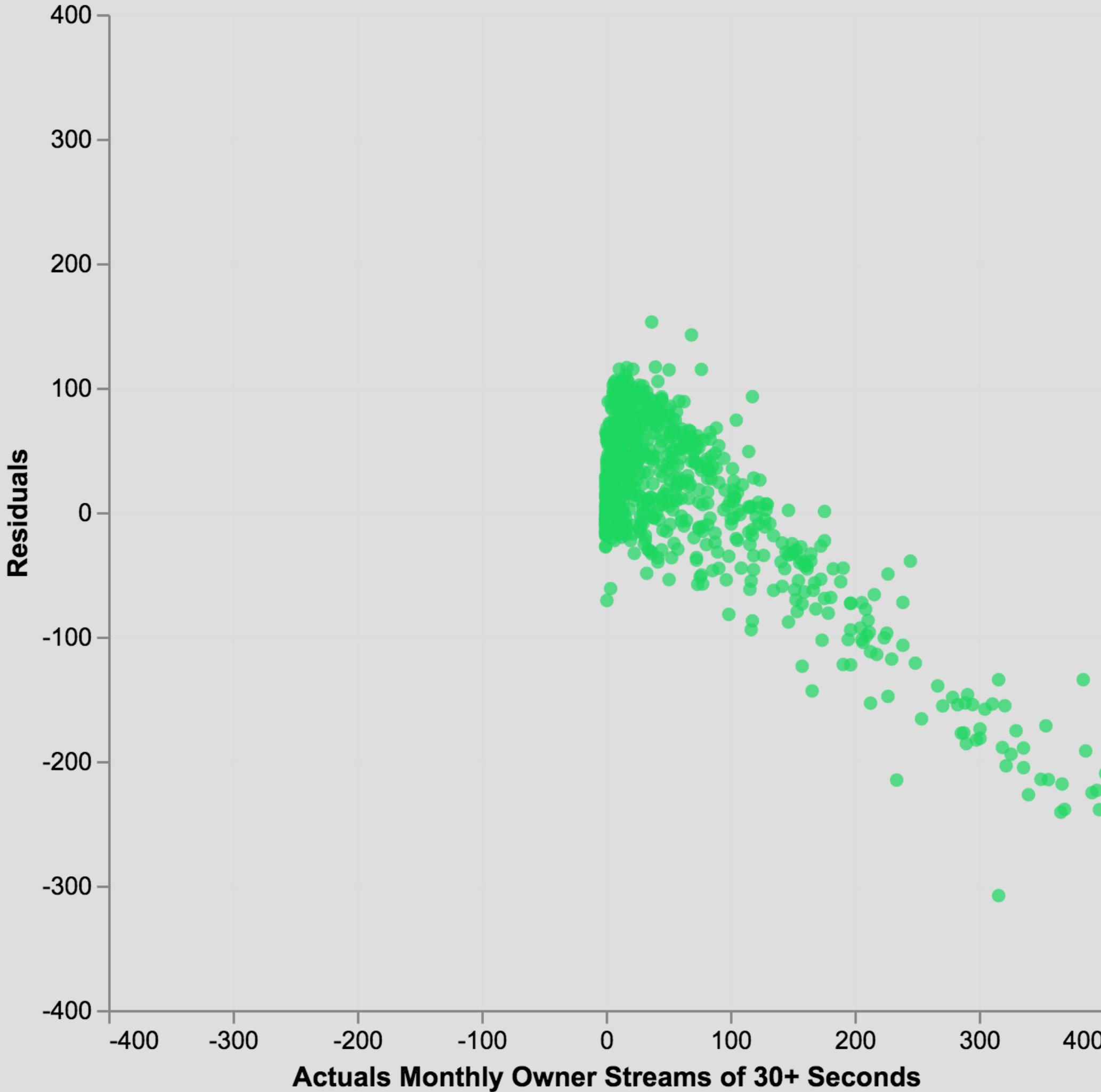
We may consider focusing the model on the more common lower-frequency users.



100-stream level: Actual (x axis) vs. Predicted (y axis)

This looks better to me, but there's always room for improvement.

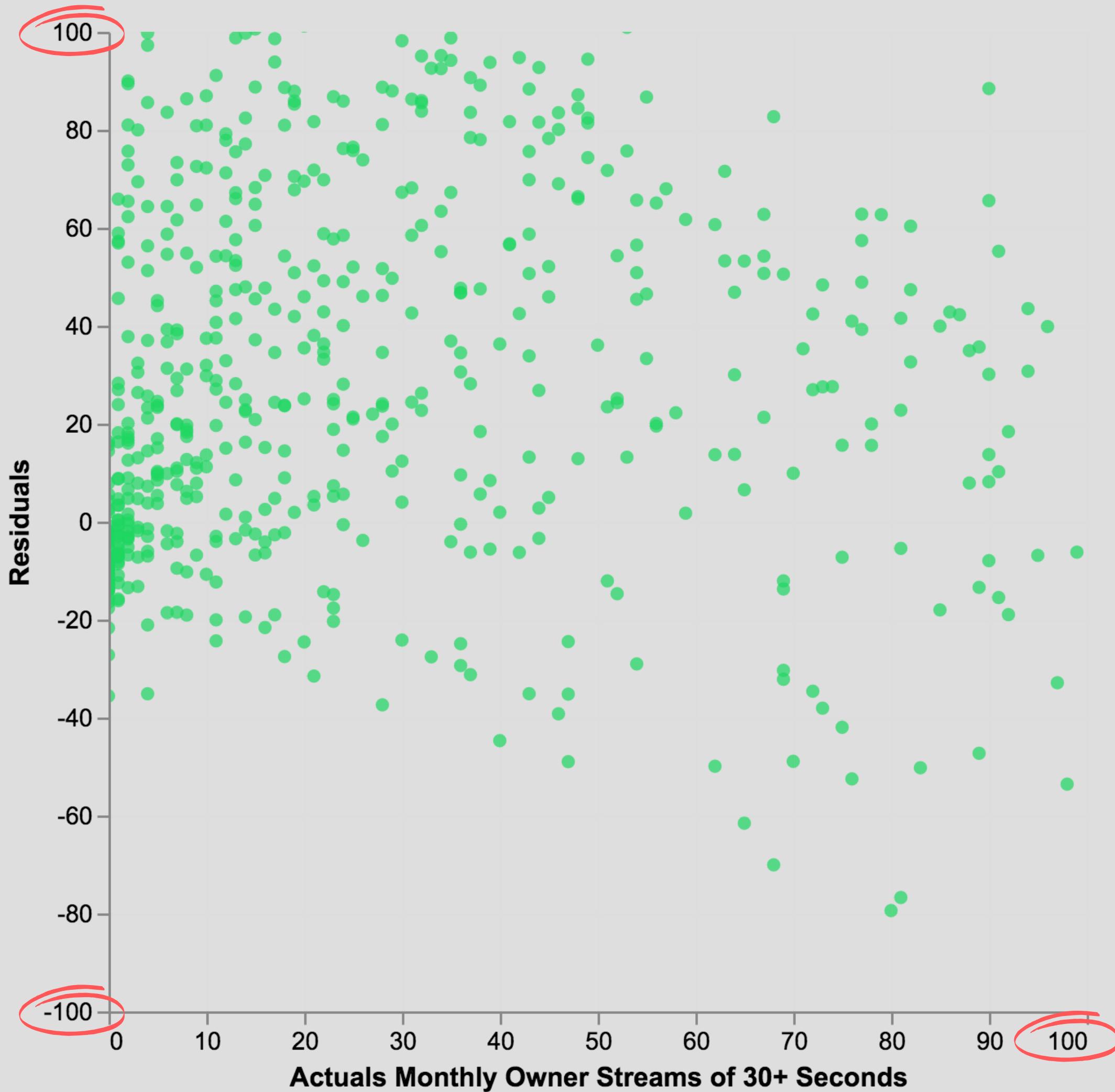




Plotting the residuals.

Again, not bad at first, but gets worse as the cases get more extreme. This makes sense because I eliminated the highest values from the training set.

Just as before, we may consider focusing the model on the more common lower-frequency users.



100-stream level:

Plotting the residuals.

This looks better to me, but there's always room for improvement. I also restricted this to positives because there can't be a negative stream count.





Spotify®

So...
what makes a
playlist
successful?

INTIMACY, FOCUS

Pos Corr: Owner's Proportion of Monthly Streams

Neg Corr: Number of Playlists User Owns

VARIETY WITHIN THEME

Pos Corr: Number of Genres (up to 3)

Pos Corr: Number of Albums



Spotify®

Next Steps

Next steps...

Ideas

- Bring In Other Types of Machine Learning Models
- Get More Spotify Data
- Do New Analysis Focusing Only on Spotify-Official Playlists



Spotify®

Thank you!

George John Jordan Thomas Aquinas Hayward, Optimist