WHY, NEW YORK CITY?

GAUGING THE QUALITY OF LIFE THROUGH THE THOUGHTS OF TWEETERS

by

SHERYL WILLIAMS

A master's capstone project submitted to the Graduate Faculty in Data Analysis and Visualization in partial fulfillment of the requirements for the degree of Master of Science, The City University of New York

2022

Why, New York City?
Gauging the Quality of Life Through the Thoughts of Tweeters

by

Sheryl Williams

This manuscript has been read and accepted for the Graduate Faculty in Data Analysis and Visualization in satisfaction of the capstone project requirement for the degree of Master of Science.

Date	Timothy Shortell Thesis Advisor
Date	Matthew K. Gold Executive Officer

THE CITY UNIVERSITY OF NEW YORK

ABSTRACT

Why, New York City?

Gauging the Quality of Life Through the Thoughts of Tweeters

by

Sheryl Williams

Advisor: Timothy Shortell

As a resource for social data, Twitter's platform has been used to measure the quality of life through sentiment analysis. This capstone project explores another methodological technique—querying Twitter data around specific keyword terms to determine dominant topics, word patterns, and sentiment leanings in a geographical area. Focusing on New York City and Los Angeles for comparative analysis, the keyword term "why" will be used to build a Python analysis around topic modeling and sentiment analysis. Using this approach, the analysis reveals social and cultural differences, the overall sentiment of tweets, and subjects of interest to tweeters.

GitHub Repository for all the files: https://github.com/shewilliams/whynyc.

Website: https://shewilliams.github.io/whynyc/.

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my advisor, Tim Shortell, for the continuous support of my master's capstone. This project has led me to many rabbit holes, but Tim's guidance and immense knowledge helped me stay on track in all the time of programming, research, and creation of this capstone.

Finally, I would like to thank my family for supporting and encouraging my studies. And to my dearest and closest friends that have kept me sane during my master's degree journey—Laura, Alexa, Christine, Carmine, and Jamie…I couldn't have done this without you!

CONTENTS

LIST OF FIGURES

LIST OF TABLES

DIGITAL MANIFEST

I. **Capstone Project Whitepaper (PDF)**

II. **WARC Files**

   a. Project Website

   https://shewilliams.github.io/whynyc/

III. **Code and other deliverables**

   a. Zip file containing the contents of the GitHub repository at the time of deposit.

   https://github.com/shewilliams/whynyc/

WHY, NEW YORK CITY? GAUGING THE QUALITY OF LIFE THROUGH THE THOUGHTS OF TWEETERS

Introduction

Why, New York City? Why, indeed. This project explores building a narrative by analyzing geo-targeted public data based on topic modeling, sentiment analysis, and using specific keyword terms. For more context, according to Simon Sinek, your "why" in experiences serves as a compass for your values, beliefs, and instincts. Furthermore, it attracts people who believe in what you believe in. While Sinek's Ted Talk, "Start with Why," takes a business-minded focus (2009), for this project, I will take a quality of life (QoL) focus. According to the World Health Organization, QoL is defined as "an individual's perception of their position in life in the context of the culture and value systems in which they live and in relation to their goals, expectations, standards, and concerns."

How exactly does the word "why" and QoL work together? "Why" has a significant part in the English language to identify the causes of an event. Specifically, the Oxford dictionary defines "why" as: used in questions to ask the reason for, the cause, or purpose of something; used in questions to suggest that it is not necessary to do something; used to give or talk about a reason. This simple word sparks curiosity or an investigative sense in determining what causes events to occur in one way, rather than taking a different trajectory.

My interest in this subject area stems from the idea of intertwining narratives and QoL to tell stories based on actions and beliefs from a geographical perspective. Given Twitter's position as a platform for public discourse, it is a perfect tool for gauging how tweeters voice their concerns and thoughts. While the primary focus will be the New York City area, this geographical location will be compared with the Los Angeles area. Using the keyword term "why," I hope to reveal word patterns, trends, popular topics, and sentiment levels to tell a narrative of how tweeters voice their opinions.

There are many ways to go about this project; therefore, to cover a broad overview of this narrative, the following will be covered through a data visualization analysis: an introductory exploration of the data; scoring the sentiment (positive, neutral, negative), and subjectivity of each tweet; collecting the most discussed "topics" and associated words (subtopics); and lastly the words will be visualized on a geographical map. This analysis will be accomplished by accessing Twitter's API with the Python programming language. The core of this paper will focus on the foundation and analysis behind this project. The analysis's technical aspects, methodology, and choices will appear in the appendices.

The link to the project (https://github.com/shewilliams/whynyc/) includes both the backend analysis and a website (https://shewilliams.github.io/whynyc/) containing a brief summary of a few data visualization charts. Feel free to take a look!

Framework

Traditionally, QoL research has been developed through qualitative or quantitative methods. An explicit example is WHOQOL, developed by the World Health Organization group to examine the cross-cultural quality of life assessment. Other examples are World Values Survey, OECD Better Life Index, and World Happiness Report. In recent years, public discourse on Twitter has been used to measure QoL by measuring feelings towards a particular topic. One method captured QoL via geotagged tweets and measured perception between different areas of

Bristol, England (Zivanovic et al. 2018). Specifically, in the NYC area, methods of geo-targeted tweets and sentiment analysis were used to gauge the feelings of public facilities (Hollander et al. 2018) and public parks (Plunz et al. 2019).

Furthermore, the courses I have taken in the Data Analysis and Visualization program have shaped my understanding of storytelling and narratives through data analytics. From a technical and statistical side: “Working with Data: Fundamentals,” “Data Analysis Methods,” and “Advanced Data Analytics”: I have gained programming knowledge in Python to process a dataset for different types of analysis, e.g., statistical, exploratory, and predictive. In particular, the “Working with Data: Fundamentals” class sparked my interest to study public discourse on Twitter and the Rokeach Value Survey—values or guiding principles of importance to someone.

Through a humanities lens, the themes that emerged from “Data, Culture, and Society” and “Alternative Data Cultures” are invisibility by design, the ethics of interpretation, the linearity in closed versus open narratives, and agential realism. In *Meeting the Universe Halfway: Quantum Physics and the Entanglement of Matter and Meaning*, Karen Barad defines this concept on agential realism: “practices of knowing and being are not isolable; they are mutually implicated. We don’t obtain knowledge by standing outside the world; we know because we are of the world. We are part of the world in its differential becoming.” (185) Furthermore, Barad states:

> *There is in this sense no privileged position from which knowledges can be produced, as the researcher is of the world. Researching phenomena, then, is a methodological practice of continuously questioning the effects of the way we research, on the knowledges we produce. This unfolds itself as an ethico-onto-epistemology of knowing in*

> *being. Ethics is about being response-able to the way we make the world, and to consider the effects our knowledge-making processes have on the world.* (381)

To simply put it, we as humans see, discover, and reveal the things we observe. This form of knowledge creation can either be included or excluded in their space. Therefore, I interpret this as observers will always be biased when their expectations, opinions, or prejudices influence what they perceive or record.

Other courses at the CUNY Graduate Center have sparked my interest in narratives and society. “Narratives of New York: Literature and Visual Arts” and “Metropolis: A Political, Historical, and Sociological Profile of New York” offered insight into the city’s past and present regarding the entities that defined and shaped NYC as a cultural, social, and economic institution. Additionally, the discussion of micropolitics and the activities to induce change among people when maneuvering communities. Also, I learned about the history and methods behind human population changes after taking a class in “Introduction to Demography.” This course solidified my understanding of the criteria (such as birth, education, or socioeconomic status) that affect societies or groups. Building on my studies in “Introduction to Demography,” “Hierarchical Linear Modeling” sparked my interest in society and structures from a statistical perspective by identifying hierarchical relationships between different phenomena such as test scores, grade level, and socioeconomic status.

The Narrative

So, what is the public discourse like around the word “why”? After removing stop words (a set of commonly used words in a language), hashtags, mentioned users, and links—based on the words used and given that the word "people" is the 2nd more frequently used word, it appears tweeters are providing their opinions from experiences, that people should be aware of.[1]

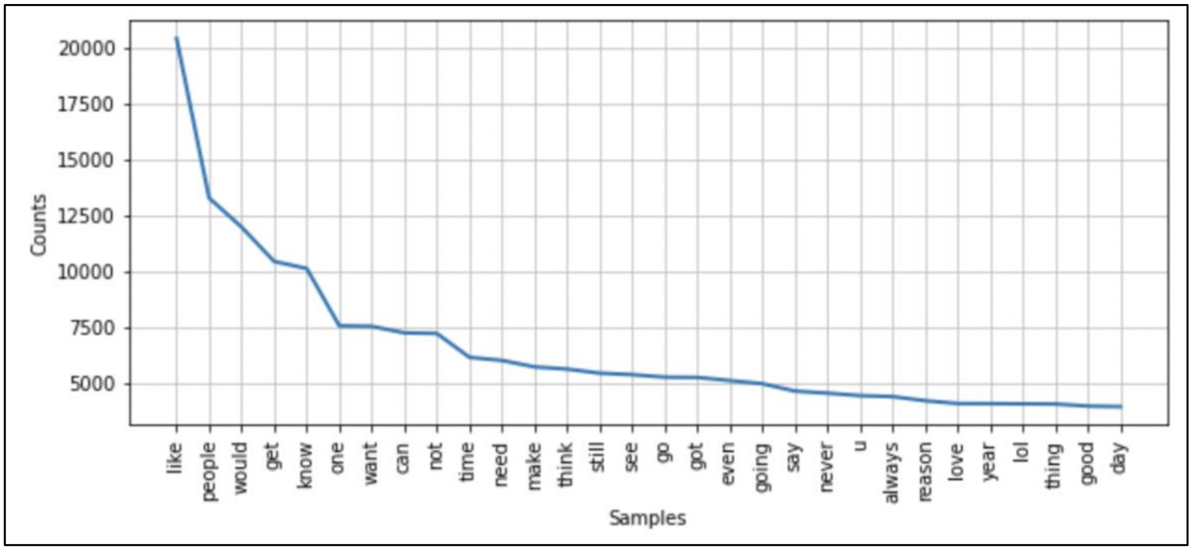

Figure 1 - The top 30 most frequently used words for the New York City area.

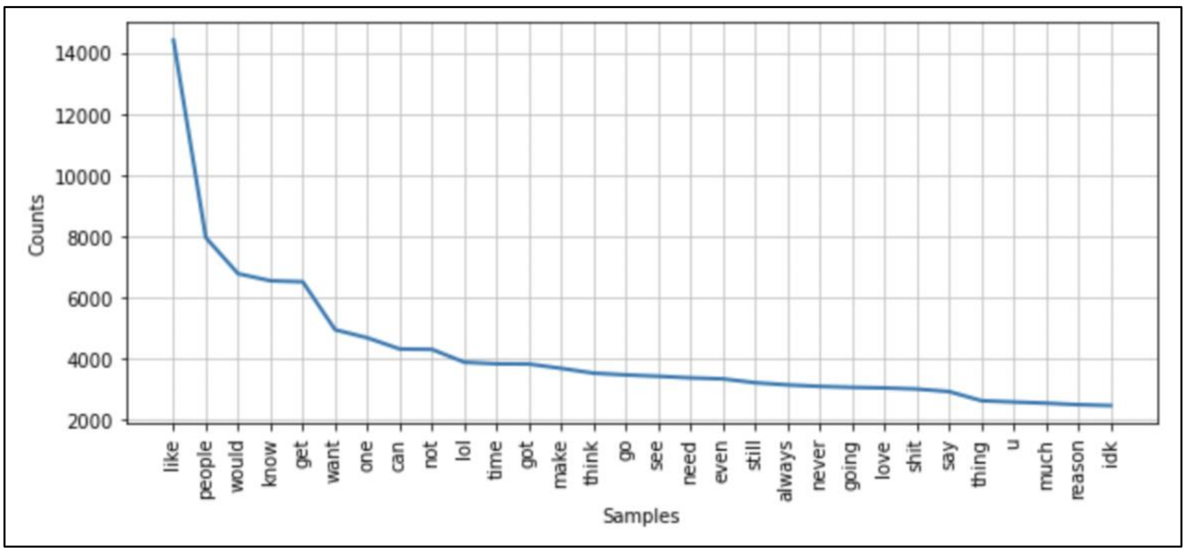

Figure 2 - The top 30 most frequently used words for the Los Angeles area.

[1] Profanity, textspeak, and emojis were kept in the analysis to discern the frequency of usage.

Now, the words will be mapped in a word cloud, not geo-located, but shaped as their location. Whether or not it's randomly generated, it is interesting to see that the word "know," "need," and "people" are situated close to one another in both NYC and LA.

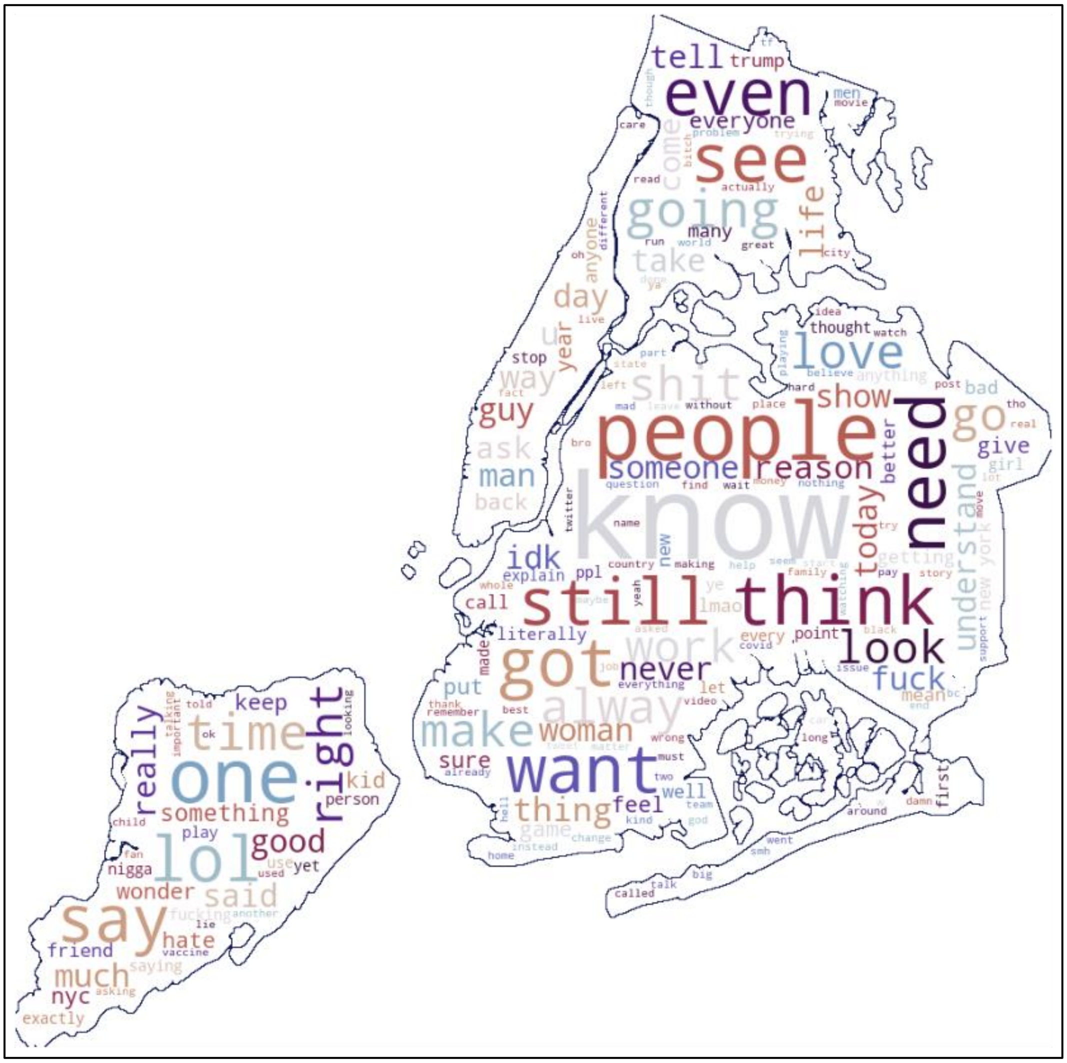

Figure 3 - A New York City-shaped word cloud graphic.

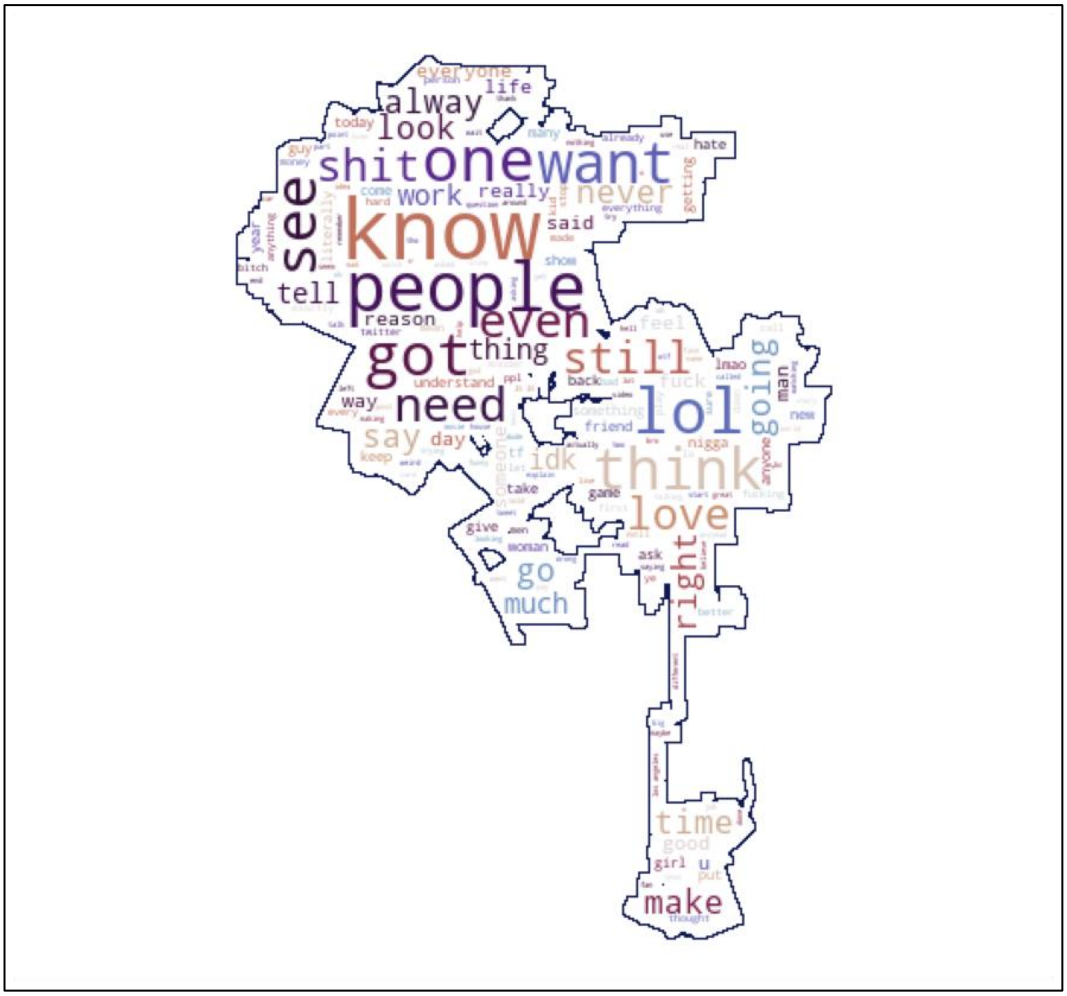

Figure 4 - A Los Angeles-shaped word cloud graphic.

N-grams are a continuous sequence of words that predicts the most probable word that might follow. I will look at the n-grams ranging from 1 to 4 and their frequencies for this analysis. For the unigrams and bigrams columns, it appears that tweets gear towards opinionated questions—especially with word pairings such as “look like,” “feel like,” “make sense,” and “want know.” This set and another set of word pairings, “people like” and “people think,” potentially demonstrate opinionated questions about the actions and events they have witnessed.

Moving into the trigrams and quadgrams analysis, several words are repetitious, indicating a tweet copied and shared on social media to bring attention to a cause.

Table 1 - N-grams analysis for LA dataset up to 4-grams.

```
ngrams_la
```

	unigrams	frequency	bigrams	frequency	trigrams	frequency	quadgrams	frequency
0	like	2499.263034	look like	412.877802	los angeles california	248.571760	lt lt lt lt	55.803542
1	people	1345.377911	feel like	343.755864	cosmos graphically audiovisual	100.289355	graphically audiovisual face race	32.907513
2	know	1306.119407	los angeles	303.459807	lt lt lt	57.832753	audiovisual face race age	32.907513
3	lol	1107.624707	make sense	232.736469	make make sense	53.187767	face race age nationality	32.907513
4	want	1004.736682	angeles california	176.882104	graphically audiovisual face	31.199698	race age nationality exact	32.907513
5	got	863.985108	want know	168.806545	audiovisual face race	31.199698	age nationality exact location	32.907513
6	love	782.042870	people like	142.731390	face race age	31.199698	cosmos graphically audiovisual face	32.700615
7	think	753.242563	sound like	137.523345	race age nationality	31.199698	nationality exact location everybody	27.344363
8	time	746.985075	year old	127.321350	age nationality exact	31.199698	los angeles hollywood california	13.905527
9	make	726.612333	social medium	121.212786	nationality exact location	31.199698	al haqq nur graphically	11.941114
10	idk	716.284884	year ago	108.906619	exact location everybody	25.998270	haqq nur graphically audiovisual	11.941114
11	shit	713.568398	understand people	100.629940	idk feel like	23.623206	cosmos graphically audiovisual body	11.901917
12	need	682.908697	graphically audiovisual	88.699178	gt gt gt	19.437887	guest catch live weeknight	9.367740
13	say	671.546791	people think	83.685771	make feel like	17.547489	catch live weeknight et	9.199194
14	going	659.456755	cosmos graphically	82.309168	today feel like	17.177489	south los angeles california	9.098973

Table 2 - N-grams analysis for NYC dataset up to 4-grams

```
ngrams_nyc
```

	unigrams	frequency	bigrams	frequency	trigrams	frequency	quadgrams	frequency
0	like	3544.885695	look like	659.396963	new york city	153.814275	new york new york	160.036687
1	people	2168.265941	new york	648.573531	new york new	117.045252	news network elected official	31.975772
2	know	1970.852427	feel like	507.794736	york new york	114.497567	network elected official silent	31.975772
3	want	1521.226327	make sense	402.291622	brooklyn new york	88.826989	elected official silent obvious	31.975772
4	lol	1237.363350	want know	315.133293	make make sense	79.606457	official silent obvious miscarriage	31.975772
5	time	1184.918183	year old	215.991991	manhattan new york	56.025358	silent obvious miscarriage justice	31.975772
6	got	1178.136160	people like	214.029920	gt gt gt	36.218751	obvious miscarriage justice social	31.975772
7	need	1171.725748	sound like	196.116324	idk feel like	32.939862	miscarriage justice social security	31.975772
8	think	1165.149483	year ago	187.608516	really want know	31.259159	justice social security irs	31.975772
9	make	1119.506357	social medium	182.148618	news network elected	31.205158	social security irs administration	31.975772
10	love	1067.797079	black people	161.587825	network elected official	31.205158	security irs administration ssi	31.975772
11	say	1042.704998	understand people	159.930516	elected official silent	31.205158	irs administration ssi veteran	31.975772
12	going	1022.824857	people think	136.607558	official silent obvious	31.205158	administration ssi veteran deserve	31.975772
13	reason	950.252868	acting like	133.910211	silent obvious miscarriage	31.205158	ssi veteran deserve date	31.975772
14	idk	912.889412	like know	133.420595	obvious miscarriage justice	31.205158	veteran deserve date expect	31.975772

Depending on who tweeted these messages, this may hint at cases of potential either civic engagement or collective action. Another observation is the word chain "make make sense," *[sic]* (the stop word "it" was removed, so the original phrase is possibly, "make it make sense"), a term generally used to make something easier to understand. This indicates a desire to know why things are a certain way. Next, I'll look at the sentiment of the tweets. For both areas, positive tweets account for the most, followed by negative and neutral. This can imply that the positive and negative tweets fall aligned with the idea of opinionated questions, whereas neutral tweets may be newspaper headlines or informative tweets.

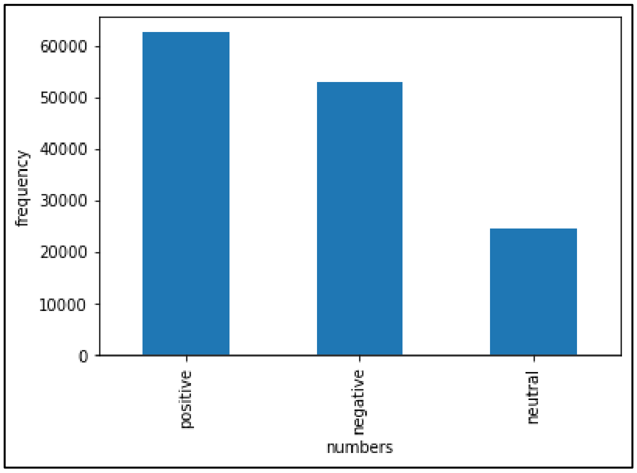

Figure 5 - The number of tweets labeled by the level of sentiment for the NYC area.

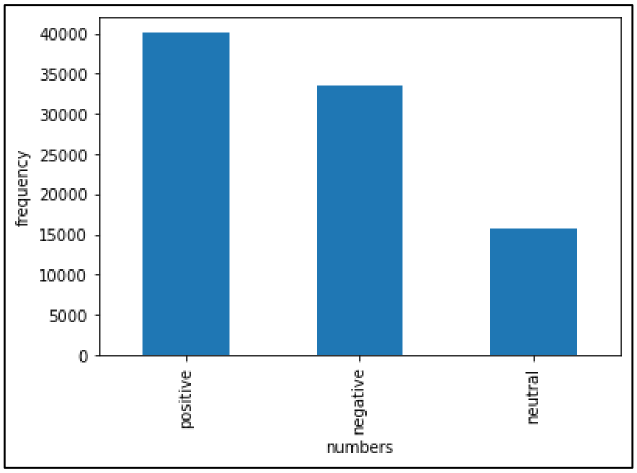

Figure 6 - The number of tweets labeled by the level of sentiment for the LA area.

Twitter’s tweet character count is 280; however, user mentions, hashtags, and links do not account for that character limit. They are not taken out for this portion of the analysis. Therefore, to get a better idea of how the positive, neutral, and negative are structured by word length, the following histograms in Figure 7 and Figure 8 were created.

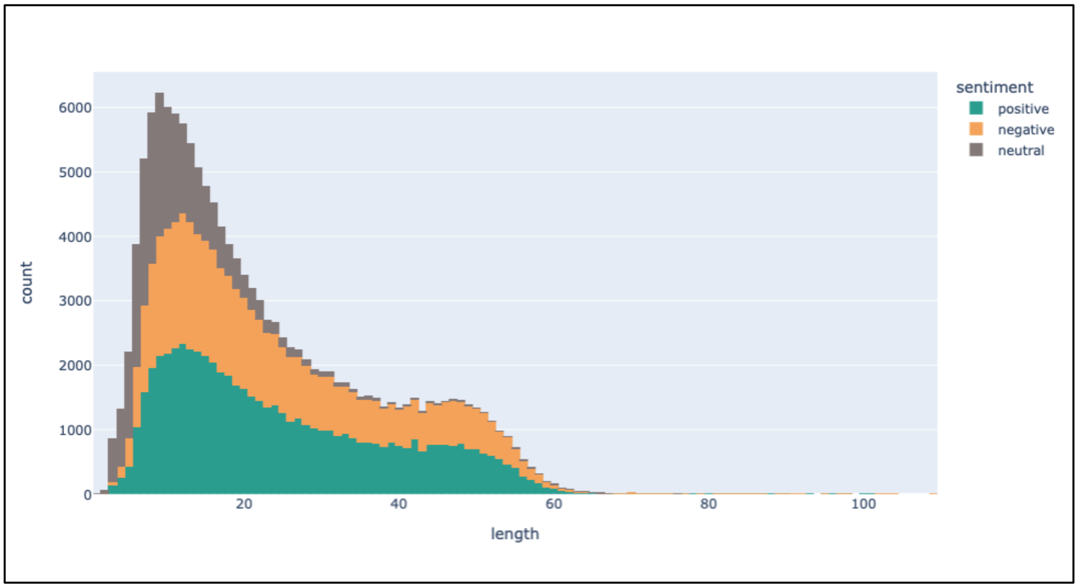

Figure 7 - Length of NYC tweets based on their measured sentiment.

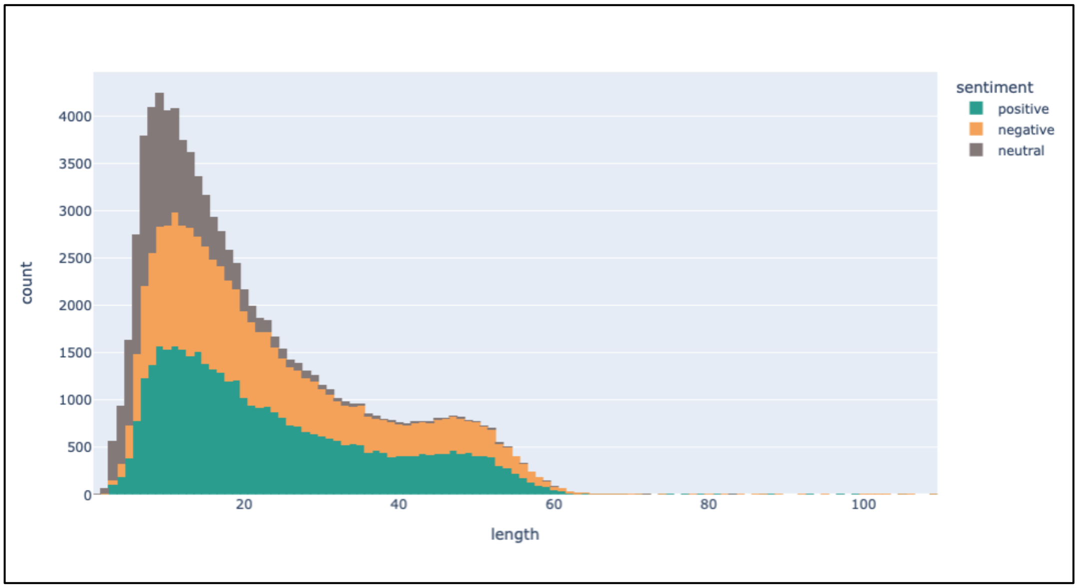

Figure 8 - Length of LA tweets based on their measured sentiment.

Both graphs are similarly shaped and showcase the same results—neutral tweets tend to be shorter in word length, whereas positive and negative tweets may go over 50 words. Overall, tweeters express their thoughts in 20 words or less. Now, I will look at the tweets over time (all tweets are pulled from 2021) in terms of sentiment.

Figure 9 - Timeline of sentiment in NYC.

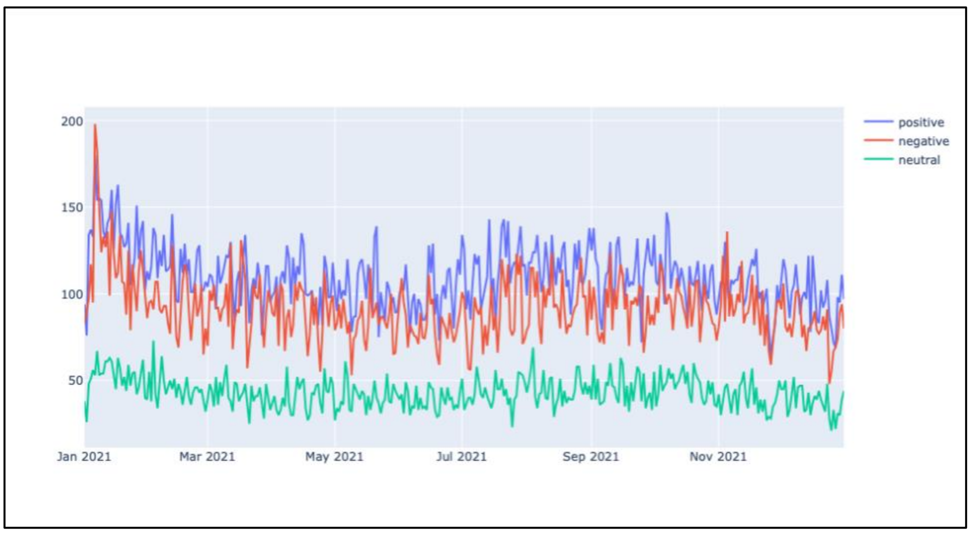

Figure 10 - Timeline of sentiment in LA.

Overall, there appears to be a sinusoidal wave throughout the year—indicating a possible seasonality in tweets. The first quarter and the summer season seem to have more tweets than the holiday season and the year's second quarter. There is a significant drop in tweets around Thanksgiving and Christmas, implying that tweeters are enjoying their time elsewhere. Now for NYC, there is a spike in negative tweets on January 6th and positive tweets on March 29th. The word clouds are pulled to window-in on those dates.

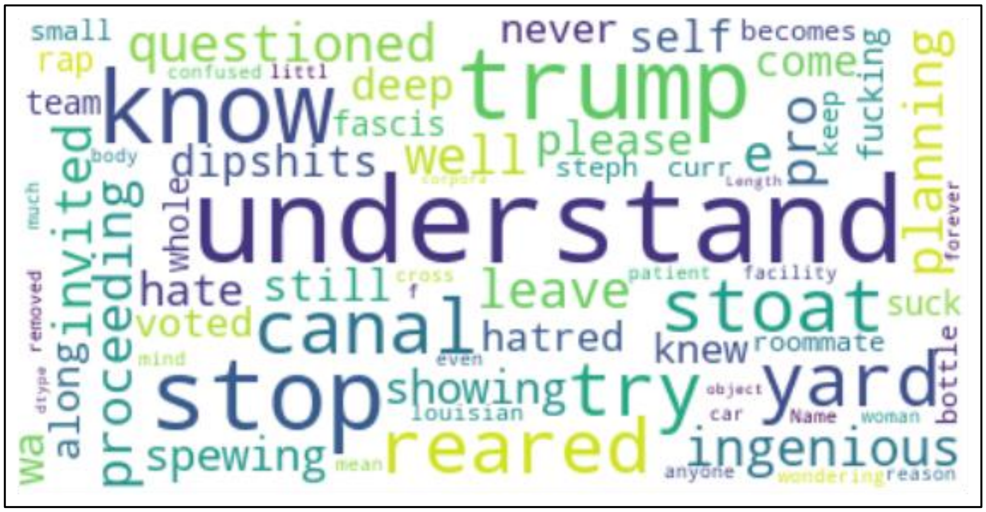

Figure 11 - Negative tweets on January 6th for NYC.

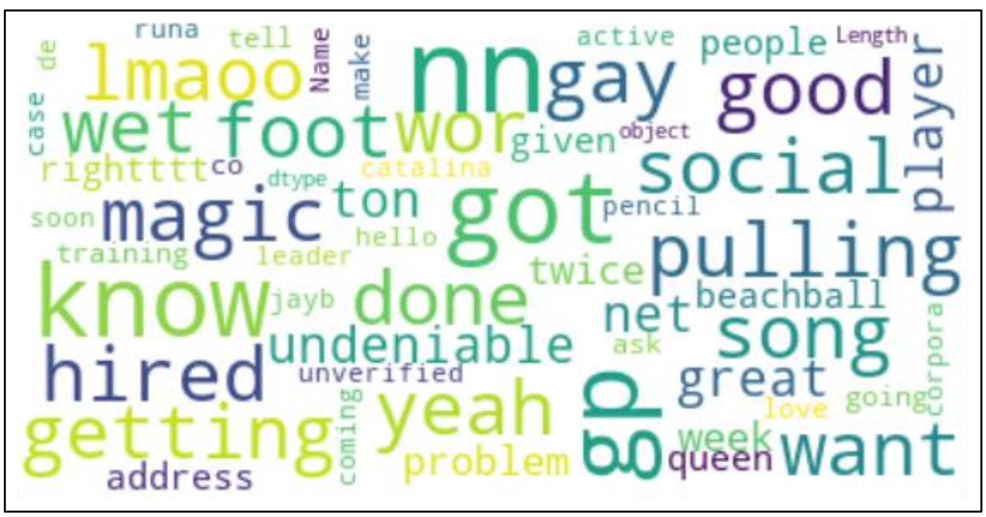

Figure 12 - Positive tweets on March 29th for NYC.

It’s difficult to discern the event of March 29, 2021, in NYC. Considering the date’s proximity to the start of the baseball season, it can be sports, with words such as “player,” and “training.” Another attempt to figure out what happened on this day is the top trending Twitter topics. According to WinCalendar, it’s these: Mike Woodson (sports), Derek Chauvin (BLM/police brutality), Andre Drummond (sports), India vs. England (sports), Austin Rivers (sports), Alabama basketball(sports), NFT (digital assets), Creighton basketball (sports), Petr Kellner (death of entrepreneur), Good Friday (holiday), Gonzaga (sports), and #GeorgeFloyd (BLM/police brutality). At this point, further analysis is needed, which is out of the scope of this project, to determine which content topics dominate NYC-based tweets on March 29th.

Regarding the January 6th tweets, words such as “understand,” “stop,” “fascis,” *[sic]* and “hatred,” may indicate the attack on the United States Capitol to overturn the results of the 2020 presidential election. More noticeable is the appearance of Donald Trump’s last name in Figure 11. In comparison to the results for the LA area on the same date, Trump’s name is not prevalent in this word cloud; however, the name of his wife, Melania, appears. This may imply less political interest in the LA area than in NYC, potentially indicating different areas of interest and concern.

Figure 13 - Negative tweets on January 6th for LA.

Moving to another part of the analysis, Tables 3 and 4 showcase the extracted topics and subtopics of tweets. Another side note, Figures 14 and 15 are map visuals of the topics using the data in Tables 3 and 4, respectively. However, the topics (or scatter points) are randomly generated geocoordinates—they are not mapped to specific locations. Additionally, we better understand the subtopics associated with the highest word frequencies in Figures 1 and 2. For example, the topic "people" is associated with "white," "sound like," "tax," "covid," and "vaccine" in NYC. Furthermore, the negative-leaning version has words like "trump," "republican," "american," [sic] "understand," and "never." In this case, the difference in tone and word usage may be the reason for different sentiments on the topic "people."

In LA, the associated words are "lakers," "help," "scared," and "stuck," and LA's "people" topic leans negative but is not associated with political words like NYC. This may imply a difference in cultural and social standards between the two cities, implying different values amongst tweeters. Another interesting point is the emoji usage in subtopics, even more so in Table 4 for LA. The 😂 (face with tears of joy) and 🥺 (pleading face) emoji topics have positive weight as a topic with associated words such as "men," "texas," *[sic]* "yelling," "question," "cannot," "play," "asking, "and dumb." Furthermore, compared to the NYC dataset, LA's subtopics feature diversity in skin tone when using human emojis. This brings up the question if LA tweeters are more likely to use visual cues and their persona to convey or enhance their thoughts compared to NYC tweeters.

Overall, similar words express opinions of different phenomena and actions between the geographical areas. Additionally, there appear to be cultural and social differences in terms of a phenomenon that is considered a regular occurrence or frequently discussed. For example, specific topics in the NYC dataset reflect discussions about sports, politics, health, COVID-19, the train system, and parts of New York City, e.g., Brooklyn. In LA, politics and sports are prominent topics—“trump” and “knicks” have a high word frequency. There are hints of the pandemic and vaccination in topics like “would” and “exactly.” Lastly, there appears to be a social life scene in LA with topics such as “last night” and “drink.” In retrospect, tweeters will talk about what’s important to them.

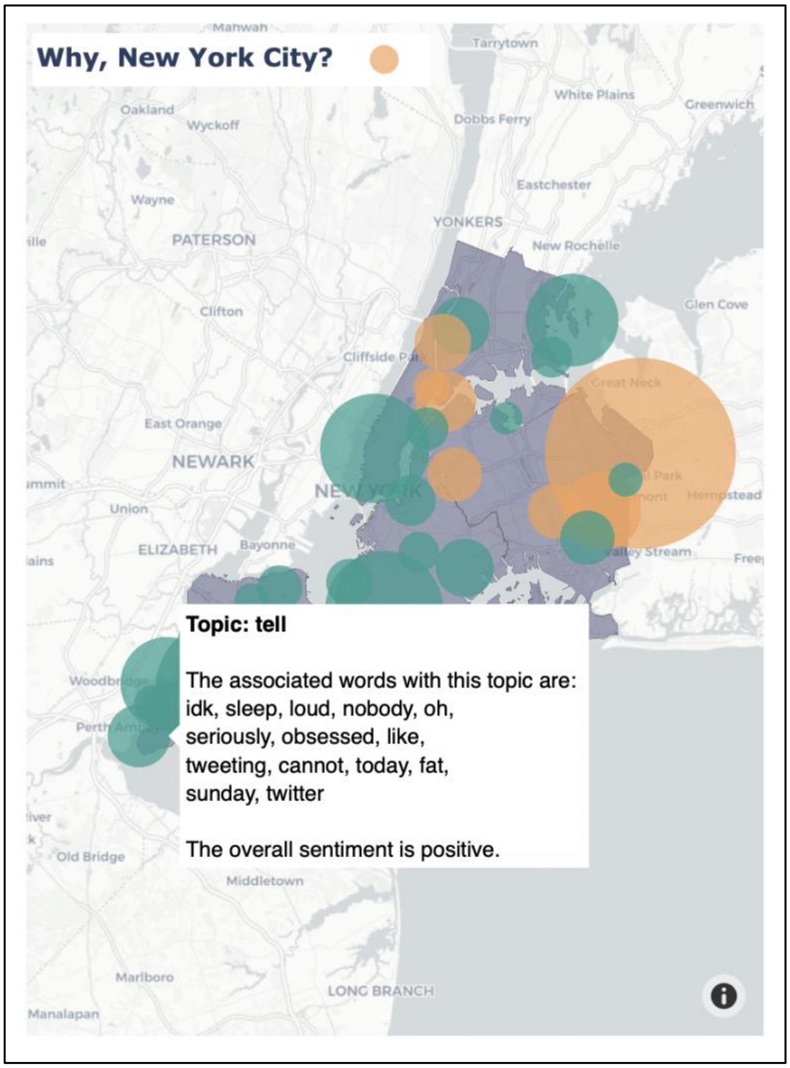

Figure 14 - Topic bubble map of NYC.

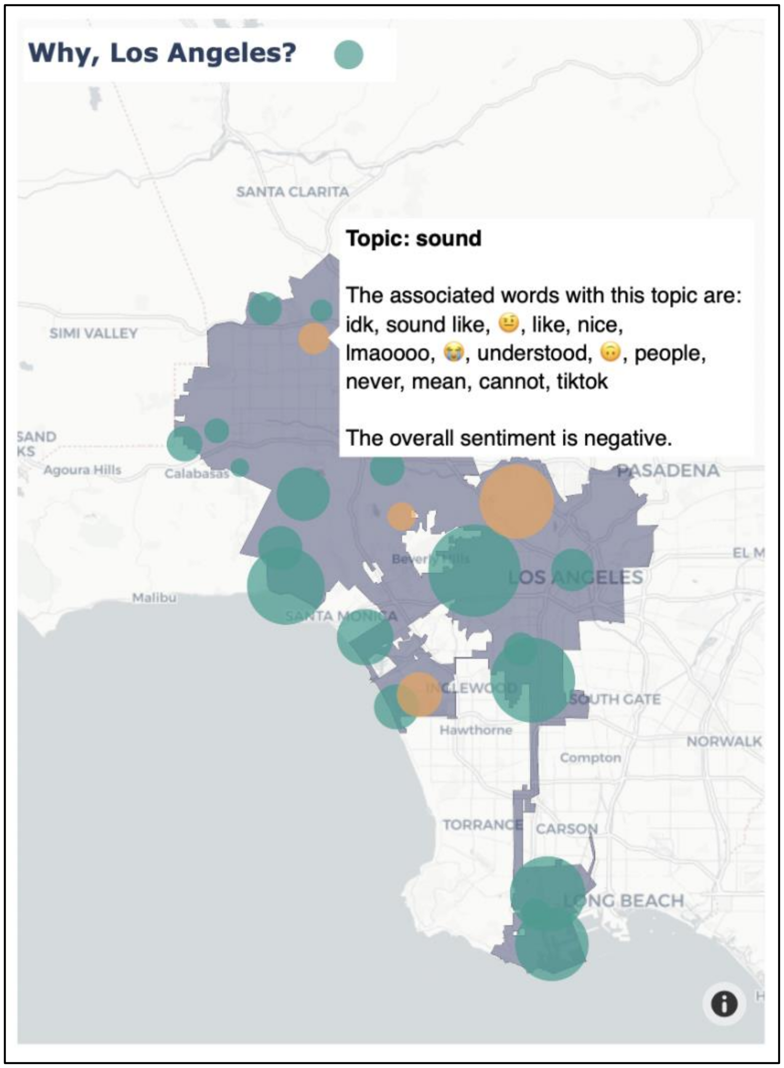

Figure 15 - Topic bubble map of LA.

Table 3 - Topic, subtopics, and level of sentiment created for the NYC area.

	topic	weight	subtopics	sentiment	category
0	like	742.7	feel, feel like, omg, shit, lmaoooo, sad, bring, yelling, people, shit like, way, fall, idk, going	-0.004653	negative
1	know	563.1	want, anyone, want know, would, would anyone, people, laughing, keep, said, even, email, like know, like, already	0.278455	positive
2	look	460.8	look like, like, 😭😭, bed, 👀, everyone, picture, man, birthday, remind, got, dark, horny, tag	0.058651	positive
3	would	426.2	say, even, cannot, 😭, mad, bother, god, would say, like, get, know, tl, see, people	0.101139	positive
4	new	364.0	york, new york, though, brooklyn, care, old, exist, choose, watch, year, city, york city, new york city, 🧐	0.034396	positive
5	love	362.3	would, 😢, blocked, much, believe, like, would love, 😫, sing, deleted, trust, dj, b, think	0.225104	positive
6	lol	309.7	tho, surprised, acting, tweet, lmaoo, real, good, like, high, business, acting like, question, taste, mind	-0.177921	negative
7	black	247.3	people, yes, medium, ask, social, social medium, black people, trump, election, gop, vote, would, day, still	0.024093	positive
8	tell	246.0	idk, sleep, loud, nobody, oh, seriously, obsessed, like, tweeting, cannot, today, fat, sunday, twitter	0.003726	positive
9	make	239.9	sense, make sense, nigga, 😂, like, come, bitcoin, would, home, back, annoying, tired, get, find	-0.276578	negative
10	cry	228.9	lmaooo, men, gay, stop, rn, like, feel, hard, fr, 😂😂😂😂, got, chicken, get, bout	0.379991	positive
11	train	228.0	🤣🤣🤣, minute, th, station, like, get, st, gas, running, hour, nyc, would, flight, smoking	0.028819	positive
12	people	222.2	understand, still, want, republican, lmfao, 🥺, cannot, american, would, understand people, like, never, state, get	-0.042978	negative
13	funny	215.5	understood, people, spend, money, never, 😩, never understood, time, like, degree, would, sport, give, much	-0.087613	negative
14	thank	214.0	wtf, many, ugh, like, people, beautiful, true, smoke, one, embarrassing, many people, drag, island, nail	-0.177987	negative
15	people	213.9	white, one, sound, u, reason, get, like, sound like, need, tax, let, covid, vaccine, would	0.061370	positive
16	fuck	199.5	bro, like, see, lie, post, red, smell, ad, cannot, would, lmfaoooo, first, flag, time	0.129042	positive
17	weird	186.6	wonder, damn, like, yo, favorite, 💀, happening, posting, song, dad, 🤔, jersey, sister, good	0.024052	positive
18	mask	179.2	vaccinated, wear, wait, serious, get, people, happen, going, alone, leave, really, know, bad, covid	0.040238	positive
19	go	169.7	hate, place, awake, first, want, first place, back, next, hear, read, voice, 🤣🤣, lol, go back	0.085015	positive
20	knicks	169.3	need, explain, please, team, player, last, nba, someone, night, earth, game, would, coach, lying	0.029421	positive
21	car	162.8	like, time, street, 😂😂😂, driver, drive, act, bike, park, uber, waste, subway, city, right	0.003726	positive
22	think	161.0	hell, cute, good, like, morning, haha, hungry, vote, news, looking, people, accurate, timeline, idea	0.061058	positive
23	ago	146.7	year, expensive, playing, another, year ago, much, reason, mean, game, perfect, eating, 😭😭😭😭, another reason, craving	-0.035676	negative
24	ever	146.1	😂😂, smh, like, people, ya, get, would, try, would ever, tryna, as, jesus, know, hate	0.056868	positive
25	cold	144.4	always, trending, friend, lmao, exactly, drink, cat, would, got, one, want, thing, else, early	0.375223	positive
26	sure	134.8	😭😭😭, take, single, figure, like, thought, dream, trying, feeling, really, one, best, could, hurt	0.086933	positive
27	get	133.3	nice, angry, people, health, 😔, thing, cannot, need, violence, time, terrorist, mental, would, worried	0.050290	positive
28	follow	129.4	yankee, mets, talking, sending, call, fan, 😅, getting, game, instagram, one, message, get, see	0.053181	positive
29	yeah	124.9	hot, drunk, 🤣, deserve, date, official, meme, elected, treat, 😐, start, steve, dat, network	0.146674	positive

Table 4 - Topic, subtopics, and level of sentiment created for the LA area.

	topic	weight	subtopics	sentiment	category
0	lie	98.7	people, need, though, reason, traffic, business, like, covid, show, pay, u, even, degree, n	0.029435	positive
1	expensive	89.2	album, people, cat, get, lmfaooo, queen, time, dress, like, flight, life, reply, real, saturday	0.125059	positive
2	exactly	74.6	remind, act, ball, vaccinated, understand, act like, king, pretty, tired, must, like, short, store, ppl	0.267071	positive
3	taste	55.1	n, better, seen, waste, long, people, said, exact, time, many, location, chicken, interview, never	0.043798	positive
4	look	356.3	look like, like, weird, cute, hate, talking, god, lmaoo, looking, good, trust, bout, one, got	0.051773	positive
5	like	327.8	😩, 😂😂, awake, gay, keep, 😐, getting, anxiety, phone, picture, slow, following, shit, voice	0.001596	positive
6	would	300.5	think, vaccine, know, accurate, would say, get, say, test, believe, covid, taking, like, people, next	0.042839	positive
7	lol	291.8	tho, wait, figure, high, pic, girl, lmaooo, school, like, ice, si, happened, trying, sudden	0.071350	positive
8	know	287.9	want, want know, like, come, lying, damn, would, lol, fat, apple, really, got, shit, would want	-0.072236	negative
9	feel	284.3	feel like, los, angeles, los angeles, like, california, angeles california, los angeles california, late, hating, screaming, best, take, asleep	0.101430	positive
10	love	241.6	make, sense, cannot, make sense, see, people, would, like, reason, matter, one, yes, life, attention	0.057091	positive
11	go	219.6	sleep, always, trending, 😂😂😂, going, smh, dodger, get, back, like, want, tl, cannot, people	0.076182	positive
12	funny	207.7	tweet, 😭😭, medium, twitter, like, social, sad, leave, 😭, social medium, 🥲, drunk, ok, delete	0.045823	positive
13	lmao	176.0	mad, 🥴, like, 😔, text, instagram, anyone, trash, would anyone, would, ugly, get, win, want	0.040961	positive
14	thank	175.4	tf, yes, ever, sure, explains, man, finding, anyone, 😌, would, like, 🤦🏾‍♂️, explain, know	-0.089369	negative
15	last	175.0	night, last night, blocked, know, craving, 🤦🏿‍♂️, uber, club, jail, 🤣🤣, toxic, another, cheese, broken	0.385698	positive
16	😂	172.6	men, follow, texas, happening, like, care, yelling, bad, like 😂, question, c, lol, yeah, b	0.398788	positive
17	hot	166.6	cold, acting, hell, suck, like, perfect, earth, acting like, morning, hahaha, day, coffee, bed, married	0.027085	positive
18	wtf	139.1	lmfao, everyone, else, like, nobody, ruin, watch, people, like wtf, timeline, know, long, shoe, attacking	0.281062	positive
19	trump	135.6	republican, vote, 😭😭😭, bother, state, people, surprised, biden, even, exist, laughing, election, hollywood, president	0.133676	positive
20	single	132.3	wear, every, ugh, one, time, want, get, day, ❤️, like, date, every time, always, sunday	0.102073	positive
21	first	131.1	oh, got, place, hurt, 🤔, favorite, haha, let, first place, go, wow, miss, u, hoe	0.282963	positive
22	sound	123.8	idk, sound like, 🤨, like, nice, lmaoooo, 😭, understood, 🙃, people, never, mean, cannot, tiktok	-0.340262	negative
23	cry	120.5	hard, obsessed, like, call, 😅, 👀, laugh, police, hungry, smell, kill, dawg, capitol, clue	0.104855	positive
24	rn	120.2	still, like, playing, wearing, 💀, early, fr, hair, mask, people, 😆, give, game, sexy	0.108978	positive
25	people	111.5	lt, like, thing, 🤣, help, want, minute, hear, lakers, scared, stuck, get, say, catch	-0.100640	negative
26	much	108.0	loud, fuck, like, da, blue, love, cannot, understand, never understand, people, need, know, 🤦🏻‍♀️, never	0.067394	positive
27	nigga	105.1	like, say, wonder, 😒, gym, 😂😂😂😂, 🤣🤣🤣, fuckin, dog, close, 😳, dat, walk, old	0.027282	positive
28	🥺	102.2	play, omg, cannot, drinking, stop, get, keep, as, like, asking, listening, dumb, ask, got	0.146250	positive
29	drink	101.2	black, tell, good, people, way, like, bruh, like lol, water, la, black people, block, lol, u	-0.113818	negative

Concluding Remarks

In the introduction, I mentioned that I chose to take an exploratory route as there are several ways to take this project. To build a better understanding of this narrative, I would like to explore the following ideas further with the same methodological framework: determining the sentiment around parts of a tweet (user mentions, links, or hashtags), analyzing the topic modeling of stop words and n-grams, interpreting the topics with CorEx, and determining social networks.

Users who have used Twitter will know there is a section for trending topics such as hashtags. While it was not the primary focus of this project, determining the frequency of mentioned users and hashtags around feelings of sentiment and keyword terms is an area of interest to me. I'd like to take a deeper dive into which words are more correlated with positive, neutral, and negative sentiment—especially from a geographical focus.

Additionally, all stop words, including the word, "why," were removed from the analysis. Next time, I'd like to run a similar analysis from this project, including all stop words this time. The hope through this process is to determine any discernible word patterns at trigrams and above—which may uncover virally-shared tweets or word patterns in speech to discern opinions about various phenomena. Regarding the NYC topic bubble map, the word "new" has words related to New York leaning towards negative sentiment. It would be interesting to see how the sentiment changes if the words "new york" are removed from the list of stop words.

On a technical side, I would like to explore hierarchical topic modeling through the CorEx topic library. According to Gallagher et al., this library incorporates user-specified anchor words. On top of that, using sklearn's Cohen's Kappa to measure inter-rater reliability between categorical data with the aid of topics or associated words, I'd like to see if it's possible to score my interpretation and understanding of the topics and words used. This follows a methodology of interpretation and validating topic models by Patrick van Kessel. Additionally, Twitter has a 'context_annontations' field that may be used for topical analysis. It would be interesting to implement this into the analysis.

Also, I would like to take a deeper dive into the sentiment analysis kit from NLTK and TextBlob. In a preliminary analysis, I noticed that both libraries had their interpretation of polarity. For example, manually looking through the whynyc and whyla dataframes, the NLTK library categorizes one tweet as positive, while TextBlob classified it as negative. Moreover, some tweets received a positive score that I felt wasn't framed positively. Furthermore, I created a predictive model for the 'sentiment' and 'text' columns and received a 60-70% accuracy score. I'd like to see if there are some ways to improve that.

On a data visualization end, I would like to figure out how to create a proper topic bubble geographical map. As visualized by the drawing in Figure 16 (on the following page), the scatters points do not overlap, and there are labels on the points mapped within the geographical boundaries. The audience can intuitively understand that the topics are relevant to the area of interest without hovering over the scatter points.

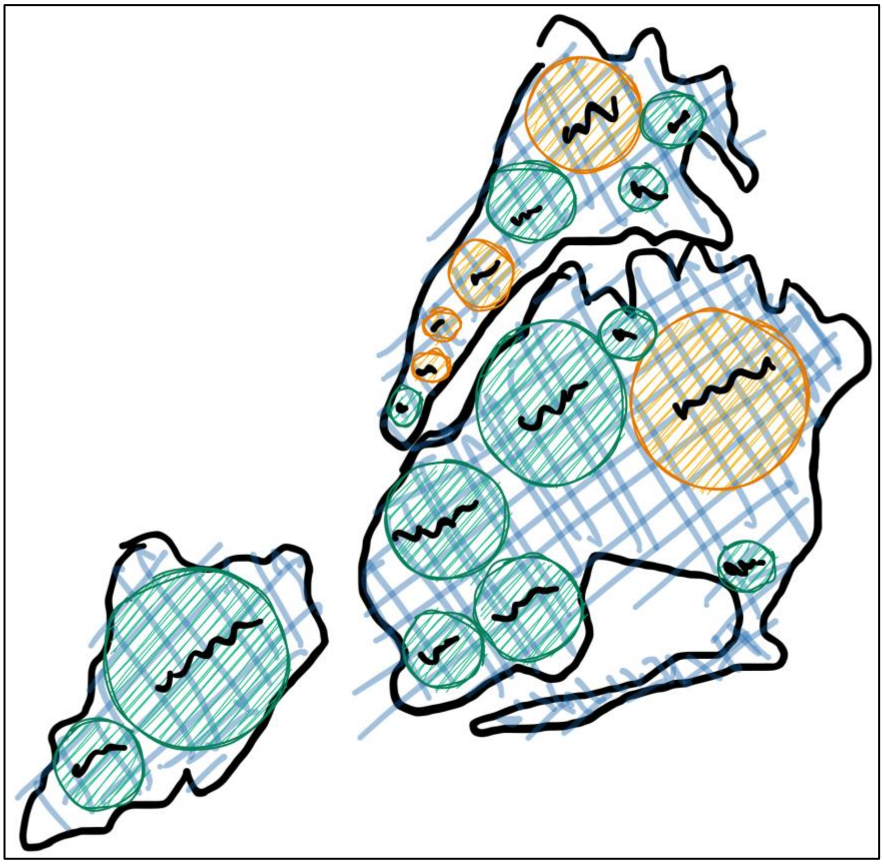

Figure 16 – Sketch of NYC topic bubble map.

Something not mentioned in my core analysis (due to Twitter's policy on sharing data), I created a few scatter plots (Figures 17 to 20) to enhance my understanding of individual tweets in perspective to the overall narrative. Earlier, I had mentioned that the accuracy of sentiment might be due to the misclassification of several tweets. By enhancing this part with the comprehensive analysis, I would like to see how individual narratives show how tweeters feel and think overall. And how it enhances our understanding and place in this world by using Twitter to express those thoughts.

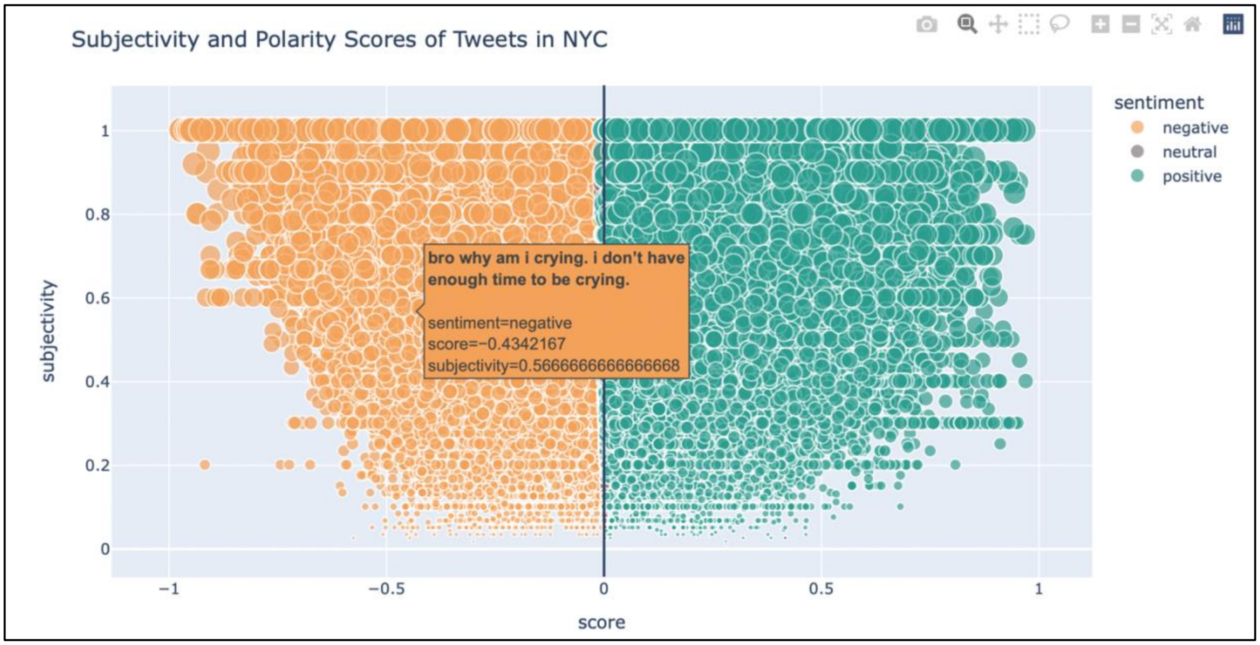

Figure 17 - Spread of NYC tweets by subjectivity and polarity.

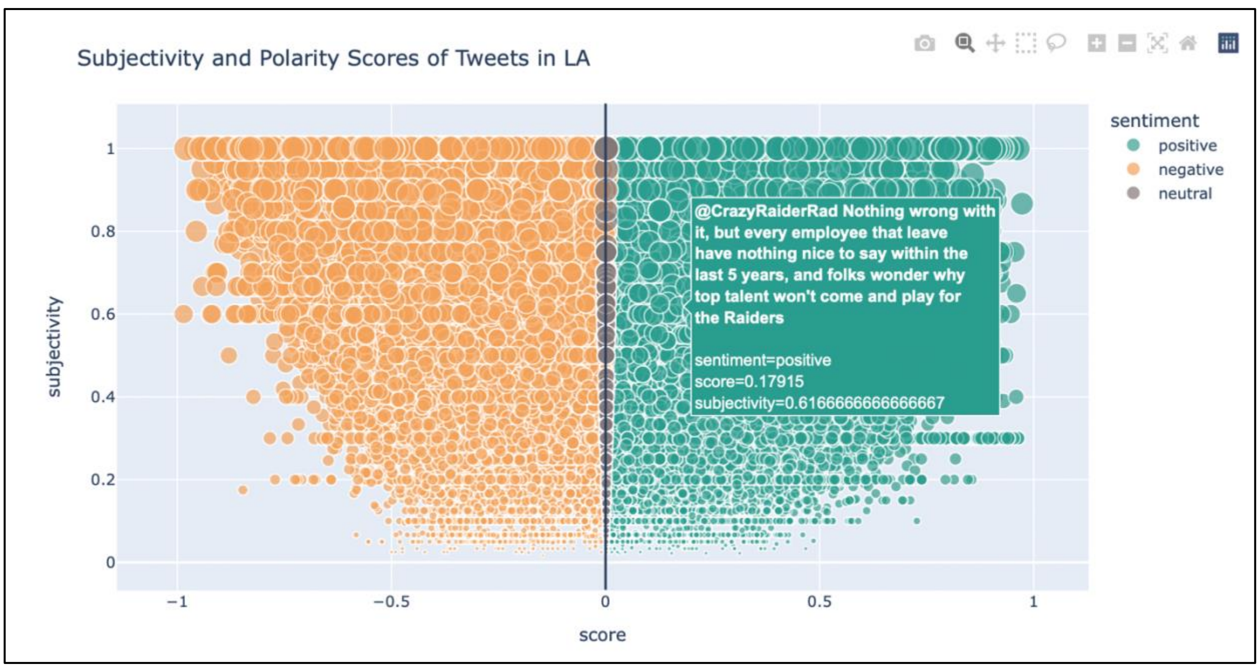

Figure 18 - Spread of LA tweets by subjectivity and polarity.

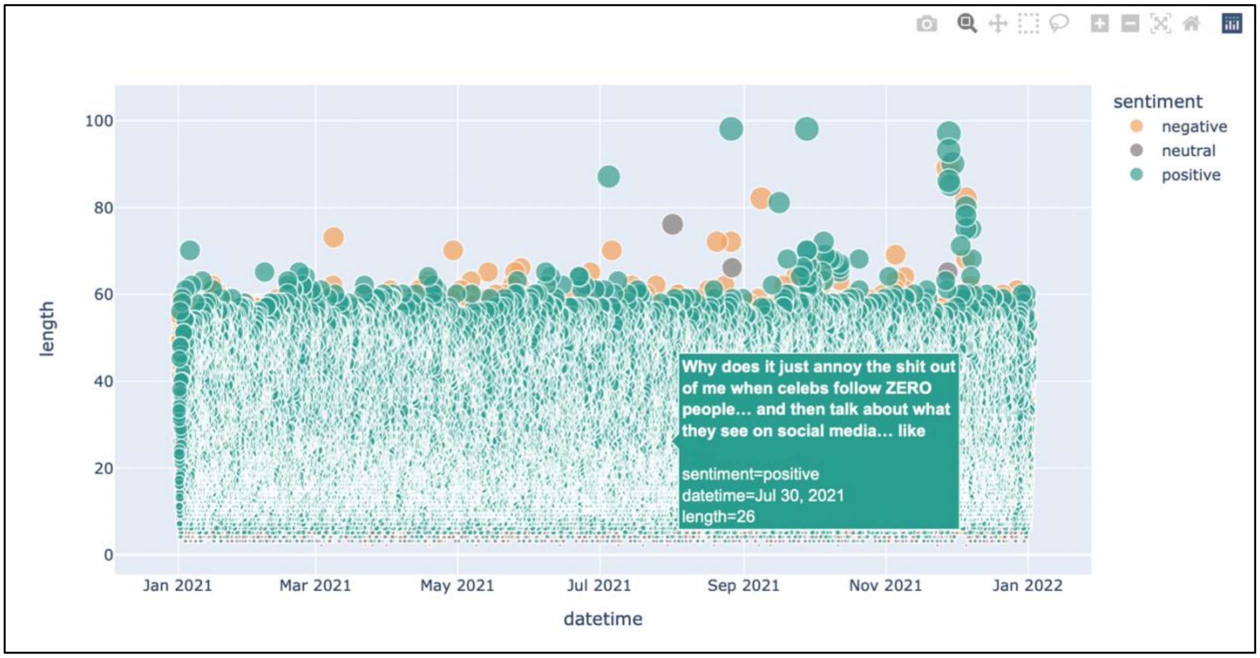

Figure 19 - Sentiment of NYC tweets throughout 2021.

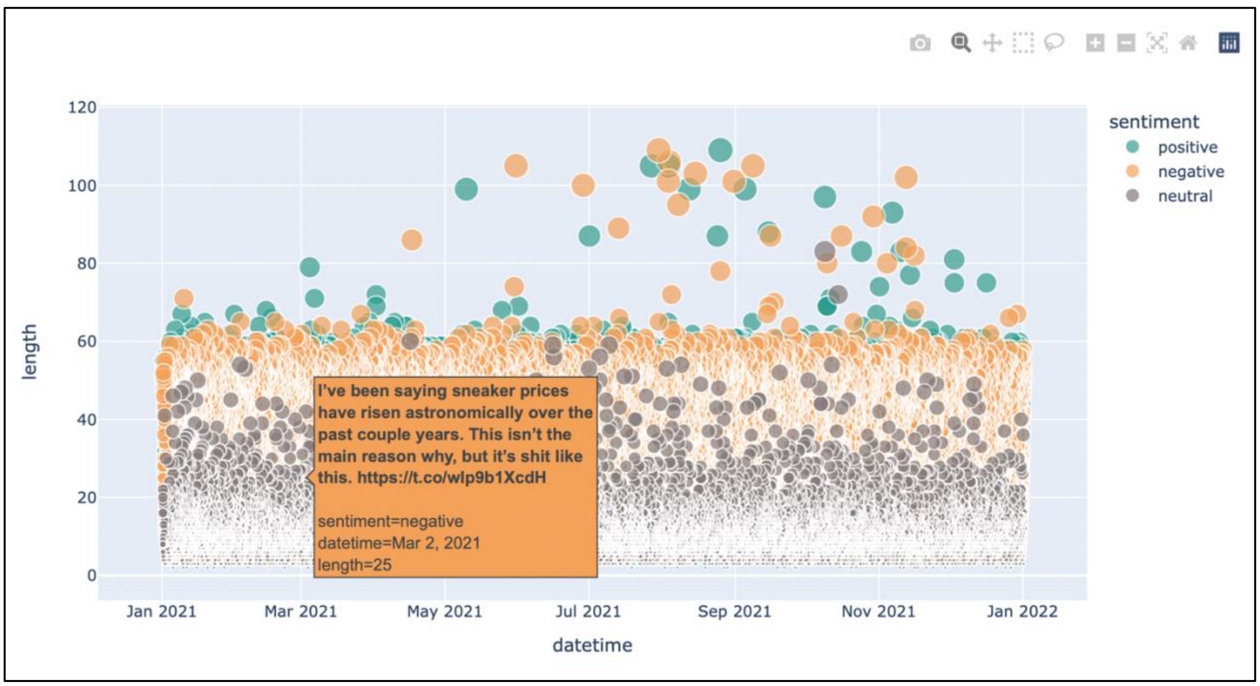

Figure 20 - Sentiment of LA tweets throughout 2021.

And that brings me to my last area of interest, knowledge and power. Building on my earlier comments on Badan's agential realism, there are some ties to Michel Foucault's power-knowledge theory is a concept that power reproduces knowledge by shaping it by its anonymous intentions (Foucault 1975). Observing a social structure of words, patterns, and users reveals connections and relationships that may tie to sources of knowledge creation or power—considering the sociological implication of behavior and meaning in social patterns through power and knowledge. The end goal would be to map out the quality of life in a geographical area through a structure of users, topics, word patterns, and hashtags.

APPENDICES

Appendix A: A Note on Technical Specifications

To successfully replicate and run the backend analysis of this project, the following is required:

- Python 3 using Jupyter notebook
- Twitter API access

To run Python 3, I have used Anaconda Distribution as this software has an intuitive graphical user interface (GPU) and allows easy access to various programming packages and content. Twitter offers varying levels of access for users. For myself, I applied for the Academic Research access as this version allows access to historical public data and full archival search. Once approval is received, a set of keys are received. Lastly, please note that the dataset will not be shared due to Twitter's Developer Agreement and Policy for Content Redistribution. However, the code will be shared for replication purposes.

Libraries

To collect the data and run the code, the following Python 3 packages must be installed. The option to download these packages is included in the Python code if you do not have them already.

- **searchtweets-v2:** This serves as a search client that supports the academic research tier of Twitter's API v2 for all publicly available tweets since March 2006.

- **nltk:** This is a common library for working with human language data. Classification, tokenization, lemmatization, and semantic reasoning will be used for this library. Additional libraries that are installed after importing the NLTK library:

    1. **nltk.download('stopwords'):** to remove stopwords from the English language.

    2. **nltk.download('punkt'):** to use the word_tokenize() function.

    3. **nltk.download('vader_lexicon'):** to import the SentimentIntensityAnalyzer() function for sentiment analysis.

- **contractions:** This library reverses the shortened version of contractions in the English language to their original words.

- **wordcloud:** This library generates a visual representation of the most frequently used words.

- **sklearn:** A machine learning tool for predictive data analysis. This will be used for the topic modeling approach to tweets and predicting the accuracy of sentiment on tweets.

- **plotly:** To graph the results of the analysis in this code. I will also use this library to export the topic bubble maps to an HTML code.

- **geopandas:** This library will generate a set of random geocoordinates within the boundaries of the areas in this project. They will be applied to the topics.

- **textblob:** Like the NLTK library, this will be used to pull subjectivity and sentiment scores.

Libraries not included and specified above are listed below. For me, these libraries were already installed when installing Anaconda.

i. **pandas:** For manipulating the Twitter data as a table.
ii. **numpy:** Perform mathematical operations and generate random numbers.
iii. **re:** To search string patterns, primarily for cleaning the tweets.
iv. **string:** Customizes string formatting, mainly used to split text in the project.
v. **textwrap:** This library manipulates and formats string text.
vi. **matplotlib:** Used to create basic graphs in the "Exploratory Data Analysis" section.
vii. **warnings:** Hides warnings that may appear when running the code.

GitHub Repository

The repository on the GitHub pages will contain the following files:

- **GeoJSON:** The boundary map for Los Angeles (la.geojson) and New York City (nyc.geojson), will be used when mapping topic bubbles.

- **Images:** PNG files of the Los Angeles (la.png) and New York City (nyc.png) boundaries. These files are used to create the word clouds in Figures 3 and 4. A folder with all the figures and tables (**figures** and **tables** in the GitHub repository) in this document will also be provided.

- **twitter_keys.yaml:** This file contains the tokens to access the Twitter API. Twitter does not allow token sharing; therefore, this file is provided as a template.

- **whynyc.ipynb:** The notebook contains all the Python coding for this analysis. Please note that this code does not show the results. GitHub has a file limit of 25MB, and the original file is 190MB. A PDF version is provided if you would like to see the results.

- **whynyc.pdf:** A PDF version of the Jupyter notebook with the results.

- **README.md:** A brief description and overview of the project.

- **HTML & CSS files[2]:** Visual Studio Code is used to put these files together.

    i. **index.html:** A webpage hosting the topic bubble map made with Plotly.

    ii. **la.html:** An export of topic bubble map from Plotly library for LA.

    iii. **nyc.html:** An export of topic bubble map from Plotly library for NYC.

    iv. **style.css:** A style customizing sheet for the index.html webpage.

[2] I will not dive into aspects of the public-facing website in this paper. Building the website is not the project's primary focus and it was created for the audience to see the topic bubble maps. Moreover, the HTML files for the maps were exported from Plotly's write_html() function.

There will also be two zip files: **whynyc.zip** contains the entire project (Python and website code) and **whynyc-code.zip** contains only the Python code.

Datasets

As stated before, Twitter does not allow the sharing of data except for the User ID and/or Tweet ID fields. I will not include a dataset file with this project; however, I will provide information on what you can expect when working with the *whynyc* and *whyla* dataframes. The final columns you can expect are:

- **id:** Pulled when querying the Twitter API. To ensure the uniqueness of each tweet.
- **created_at:** Pulled when querying the Twitter API. Used for time series analysis.
- **text:** Pulled when querying the Twitter API. This column is core to the analysis.
- **corpora:** The cleaned version of the 'text' column that removes stop words, punctuation marks, and parts of a tweet (links, user mentions, and hashtags).
- **geo.place_id:** Pulled when querying the Twitter API. It will be used for future analysis.
- **subjectivity:** Calculated subjectivity of a tweet from the TextBlob library.
- **score:** Average of the polarity of tweets calculated from the NLTK and TextBlob library.
- **sentiment:** Categorical value based on polarity score. Depending on the score, the tweet is labeled positive (score > 0), neutral (score == 0), or negative (score < 0).

- **datetime:** The conversion of the ‘created_at’ from object to datetime

- **day:** The tweet’s date was created in YYYY-MM-DD format from the ‘created_at’ column.

- **month:** The month the tweet was created, pulled from the ‘created_at’ column.

- **length:** The number of discernible words in a tweet.

Table 5 - Running whynyc.info() provides the information about the dataframe, including the index, dtype and columns, non-null values, and memory usage.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 139199 entries, 0 to 139198
Data columns (total 12 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   id            139199 non-null  object
 1   created_at    139199 non-null  object
 2   text          139199 non-null  object
 3   corpora       139199 non-null  object
 4   geo.place_id  139199 non-null  object
 5   subjectivity  139199 non-null  float64
 6   score         139199 non-null  float64
 7   sentiment     139199 non-null  object
 8   datetime      139199 non-null  datetime64[ns, UTC]
 9   day           139199 non-null  object
 10  month         139199 non-null  int64
 11  length        139199 non-null  int64
dtypes: datetime64[ns, UTC](1), float64(2), int64(2), object(7)
memory usage: 12.7+ MB
```

Appendix B: Methodology

This section entails a brief description of the thought process and methodology for each section of the Python code.

Twitter Query

A bounding box was used to precisely retrieve tweets from the geographical areas of interest to determine the Place IDs of the following locations below. Using the names or a bounding box of the geographical area proved to be a cumbersome process.

- Manhattan, NY - 01a9a39529b27f36
- Brooklyn, NY - 011add077f4d2da3
- Queens, NY - 00c39537733fa112
- The Bronx, NY - 002e24c6736f069d
- Staten Island, NY - 00c55f041e27dc51
- Los Angeles, CA - 3b77caf94bfc81fe

There are a variety of tweet fields for the request parameters that can be included in this query; however, ‘id’ ‘created_at,’ ‘text,’ and ‘geo’ will be pulled for this project. Additionally, retweets are removed (-is:retweet) because it is not a focus at this time. Ads are removed (-is:nullcast) because it is not relevant to this project, and the primary language of tweets will be English (lang:en). Lastly, the max number of tweets reflects the number of tweets pulled for the timeframe of focus. For 2021, there are about 140,000 tweets and 90,000 tweets in the NYC and LA areas, respectively. Due to Twitter’s rate limits, depending on the scope of a query, you may have to break down the search into manageable fragments.

Data Cleaning

This section aims to extract parts of a tweet (links, mentions, and hashtags) from the 'text' column and clean out stop words, contractions, numbers, and symbols. When the exploratory data analysis is performed, this removes the "noise" of the 'text' column. The 'text' column will stay as-is for future analysis.

Measuring and Categorizing Sentiment

In an earlier analysis, I noticed that the NLTK and TextBlob libraries categorize text differently from their polarity scores to measure sentiment. I decided to pull the scores from both libraries, take the average, and categorize the sentiment (positive, neutral, and negative) from that value. Additionally, I chose to include subjectivity to determine the subjectivity level of tweets. The subjectivity analysis solely depends on TextBlob because NLTK lacks scoring subjectivity.

Finally, the following colors are used for positive, negative, and neutral, respectively: Parisian Green (#2A9D8F), Sandy Brown (#F4A259), and Rocket Metallic (#847979). The choice behind this is to account for color-blindness in the red-green color scheme.

Data Analysis

This portion looks at the data, e.g., using describe() function on dataframes. On a side note, I explored how many tweets included hashtags, links, and user mentions. Considering that the tweets account for less than 1/3 of total tweets, I chose not to go further into the analysis.

Lengths of Tweets

This analysis gives insight into the length of tweets. These tweets account for parts of a tweet and the actual tweet itself. On average, tweets tend to contain 22 words. Further analysis (Figures 7 and 8) showed that tweeters mentioned more than ten users in their tweets.

Word Frequency

The FreqDist() in the NLTK library determines the top 30 most used words in tweets. This relies on the 'corpora' column after converting the dataframe into a list.

Word Cloud

The creation of the geographic-shaped word clouds from Koray Tuğberk Gübür's methodology. This analysis showcases the word frequency graphs from Figures 1 and 2 in a different perspective by using a PNG image to create a mask and map out the terms.

N-grams

This section uses the n-grams function from the NLTK library and focuses on n-grams up to 4 to discern potential word patterns.

Time Series

By focusing on sentiment, this analysis focuses on how tweets spread out from January 2021 to December 2021. The goal is to determine which parts of the year may experience spikes or declines in tweets. One set looks at the frequency of sentiment throughout the year. Another set of graphs looks at the sentiment and length of the tweet. Days with obvious spikes include a word cloud for analysis.

Topic Modeling

This part consists of building a topic model using TfidfVectorizer (TFIDF) and the Latent Dirichlet Allocation (LDA) from the sklearn library. Other topic models were explored by pairing TFIDF and CountVectorizer with Latent Semantic Analysis (LSA), LDA, and Non-Negative Matrix Factorization (NMF). There were no specific results from these six models; therefore, LDA and TFIDF are used for the final analysis. Thirty topics were pulled to build a topic bubble map.

Also, the results from this modeling were rearranged into a topic dataframe with the following columns: 'topic,' 'weight,' 'subtopics,' 'sentiment,' and 'category.' The first two columns are the topics that have the highest weight in the model. Words associated with the most prevalent topics are merged. The sentiment is the calculated mean value of all the tweets containing the topic; then, it is categorized by sentiment. The range of sentiment is between -0.5 and 0.5.

Topic Bubble Map

This section aims to create a visualization from the topic modeling results. I did not find the standard visualizations offered in the Gensim library to be effective. Geopandas, Numpy, and Plotly are the libraries used to make the topic bubble map.

- Geopandas reads the GeoJSON file, sets the boundaries, and returns points from the unary union of the boundaries.

- Numpy generates a set of random points within the geographic boundaries.

- Plotly is used to create a scatter plot on a map to display the topics as bubbles. Hovering on the bubbles shows the topic, associated words (subtopics), and the overall sentiment it leans toward.

Accuracy of Sentiment Analysis

And lastly, this section is created to determine the accuracy of sentiment on the dataset. Logistic Regression and Multinomial Naïve Bayes are used with TfidfVectorizer from the sklearn library.

Appendix C: List of Variables

Non-bold variables indicate they are referenced inside of a function.

accuracy	Calculates the accuracy of the classification in Logistic Regression.
accuracy_mnb	Calculates the accuracy of the classification in Multinomial Naïve Bayes.
avg	Calculates the mean of the NLTK and TextBlob polarity scores.
bag_of_words	Extracts the words from the provided text.
bigrams	Stores 2-gram words and their frequencies.
end_time	To reference the end date of a timeframe for tweets. This is in YYYY-MM-DD format.
fdist_whyla	Stores the FreqDist values of the most frequent words for LA.
fdist_whynyc	Stores the FreqDist values of the most frequent words for NY.
gdf_points	Contains the latitude and longitude values generated for the geographical area.
gdf_polys	Reads the GeoJSON file.
granularity	Specifies the level of aggregation in which the metrics should be returned (by day, hour, or minute). By default, Twitter query will give you Tweets per hour.
la_neg_wc_j6	Word cloud of negative tweets on January 6^{th} for LA.
label	Used as a temporary reference to label sentiment on tweets.
lemmatizer	Stores WordNetLemmatizer() function.
lr	Stores LogisticRegression() function.
mask	Reads and opens image file for word cloud.

maskable_image	Creates a usable mask by mapping values.
max_tweets	Maximum number of tweets to receive from the query request.
mean_of_topic	Stores the mean of sentiment for tweets that contain a word.
mnb	Stores the MultinomialNB() function.
model	Stores the LatentDirichletAllocation() function.
n	Set at 100 to generate a random set of coordinates.
ngrams	Concats and stores the unigrams, bigrams, trigrams, and quadgrams dataframe.
ngrams_la	LA-focused n-grams dataframe.
ngrams_nyc	NYC-focused n-grams dataframe.
no_top_words	Stores the preferred number of words for topic.
number_of_topics	Stores the preferred number of topics to be pulled from modeling.
nyc_neg_wc_j6	Word cloud of negative tweets on January 6th for NYC.
positive_wc_m29	Word cloud of positive tweets on March 29th for NYC.
predictions	Stores the predictions of X_test for Logistic Regression.
predictions_mnb	Stores the predictions of X_test for Multinomial Naïve Bayes.
punctuation	Stores punctuation symbols in string.punctuation from the string library.
quadgrams	Stores 4-gram words and their frequencies.
query	Used to build a query of search terms for the Twitter API.
query_whyla	Stores the query results for the LA area.
query_whynyc	Stores the query results for the NYC area.
results_per_call	Number of tweets or counts returned per API.
sa_nltk_list	A list that stores the polarity scores from the NLTK library.

sa_tb_list	A list that stores the polarity scores from the TextBlob library.
search_args	Stores Twitter access tokens.
sent_count	Stores frequency of sentiment by date.
sentiment_colors	Stores the colors for positive, neutral, and negative.
sid	Stores the SentimentIntensityAnalyser() function.
start_time	To reference the start date of a timeframe for tweets. This is in YYYY-MM-DD format.
stopwords	Stores the English stop words from the NLTK library.
sum_words	Adds the N-grame of words stored in bag_of_words.
symbol	Stores additional symbols not contained in string.punctuation.
temp_list	A list that stores the sentiment score for a particular word.
tf	Transforms the ‘corpora’ column for topic modeling.
tf_feature_names	Maps out the word in the matrix.
topic_df	A temporary dataframe to store the topic modeling values.
topic_dict	A temporary dataframe that pulls associate topics of the data.
trigrams	Stores 3-gram words and their frequencies.
tweet	Cleans and stores the dataset in the clean_df() function.
tweet_fields	Root-level fields contained within a Tweet object.
tweet_token_list	Split and store the words in a list.
unigrams	Stores 1-gram words and their frequencies.
values	A list that stores the average of the NLTK and TextBlob polarity scores.
vectorizer	Stores the TfidfVectorizer() function. Inputs text into number vectors for machine learning.

whyla The final dataframe used for the LA analysis.

whyla_df Converted queried data from JSON to Pandas dataframe for LA.

whyla_lda Stores topic modeling results for LA.

whyla_list Contains tokenized version of ‘corpora’ for LA.

whyla_pot Dataframe that contains parts of a tweet: mentioned users, hashtags, and links for LA.

whyla_sent Stores the sentiment frequency and date results for LA.

whyla_topics Dataframe for top 30 topics and subtopics for LA.

whyla_tweets Queried data as JSON for LA.

whynyc The final dataframe used for the NYC analysis.

whynyc_df Converted queried data from JSON to Pandas dataframe for NYC.

whynyc_lda Stores topic modeling results for NYC.

whynyc_list Contains tokenized version of ‘corpora’ for NYC.

whynyc_pot Dataframe that contains parts of a tweet: mentioned users, hashtags, and links for NYC.

whynyc_sent Stores the sentiment frequency and date results for NYC.

whynyc_topics Dataframe for top 30 topics and subtopics for NYC.

whynyc_tweets Queried data as JSON for NYC.

words_freq Stores the sum of a word that appears in the text.

x Stores randomly generated data within x_min and x_max bounds.

x_max The maximum x boundary from gdf_polys.total_bounds value.

x_min The minimum x boundary from gdf_polys.total_bounds value.

X_test The testing part of the first sequence for X.

X_train	The training part of the first sequence for X.
y	Stores randomly generated data within y_min and y_max bounds.
y_max	The maximum y boundary from gdf_polys.total_bounds value.
y_min	The minimum y boundary from gdf_polys.total_bounds value.
y_test	The testing part of the first sequence for y.
y_train	The training part of the first sequence for y.

Appendix D: Glossary of Functions

add_sentiment(why_df) Create list of NLTK and TextBlob polarity scores for each tweet, calculate the mean, store the value in a new column, and create a categorical label.

clean_df(tweet) Remove parts of a tweet, punctuation, contractions, numbers, and stopwords. Lemmatize words. Split and store into a new column.

collect_topics(x, why_df) Returns a dataframe that pulls the values from the why_lda_model. Calculates the average sentiment for topics and categorizes them.

create_wordcloud(file_name, list_name) Reads in an image file, creates a mask, and returns a word cloud of the most frequently used words.

display_topics(model, feature_names, no_top_words) Using the variables, model, feature_names, and no_top_words, returns a dataframe of abstract topics in the dataset and their importance.

find_hashtags(tweet) Returns hashtags in ‘text’ column.

find_links(tweet) Returns links in ‘text’ column.

find_mentioned(tweet) Returns mentioned users in ‘text’ column.

find_retweeted(tweet) Returns retweeted users in ‘text’ column. This function is not used in the analysis.

get_ngrams(text, ngram_from=2, ngram_to=2, n=None, max_features=20000)	Uses TfidfVectorizer to transform and map words that are the most frequent for n-gram analysis.
model_accuracy(df_name, why_df)	Splits train and test data. Uses Logistic Regression and Multionomial Naïve Bayes to predict model. Prints accuracy and confusion matrix scores of the model.
ngrams_table(why_df)	Returns a merged dataframe of unigrams, bigrams, trigrams, and quadgrams.
plot_cloud(wordcloud)	Returns a figure of the word cloud.
pol_and_sub_of_tweets(title, why_df)	Returns a Plotly scatter plot of polarity and subjectivity of tweets.
polarity(text)	Returns the polarity scores of texts from TextBlob library.
sentiment_category(sentiment)	Categorizes sentiment scores by positive (sentiment > 0), neutral (sentiment = 0), and negative (sentiment <0).
sentiment_count(why_df)	Returns a dataframe of sentiment frequency by dates.
sentiment_plotly(why_df)	Returns a Plotly time series of positive, neutral, and negative frequencies.
SetColor(x)	Maps and sets the color for topic bubble map, depending on sentiment value.
subjectivity(text)	Returns subjectivity score from TextBlob library.
topic_map(file_name, why_df, title)	Calculates geocoordinates based on geographical boundaries read in gdf_polys. Returns a topic bubble map using Plotly's mapbox and scatter plot.

transform_zeros(val)	Converts 0 (RGB value of black) to 255 (RGB value of white).
tweets_timeline_plotly(why_df)	Returns a Plotly scatter plot of date, length, and sentiment of a tweet.
why_lda_model(why_df)	Returns a dataframe of topics from modeling methodology.

BIBLIOGRAPHY

Anaconda. *Anaconda Software Distribution*. V 2.1.4. Computer software.Nov 2016. https://anaconda.com.

Barad, Karen Michelle. *Meeting the Universe Halfway: Quantum Physics and the Entanglement of Matter and Meaning*. Durham N.C.: Duke University Press, 2007.

"Borough Boundaries. GIS Data: Boundaries of Boroughs (Water Areas Excluded)." Borough Boundaries | NYC Open Data. NYC Department of City Planning, January 29, 2013. https://data.cityofnewyork.us/City-Government/Borough-Boundaries/tqmj-j8zm.

"City Boundary | City of Los Angeles Hub." City of Los Angeles Hub, November 13, 2015. https://geohub.lacity.org/datasets/city-boundary/explore.

Foucoult, Michel. "Discipline and punish." A. Sheridan, Tr., Paris, FR, Gallimard (1975).

Gallagher, Ryan J., Kyle Reing, David Kale, and Greg Ver Steeg. "Anchored correlation explanation: Topic modeling with minimal domain knowledge." *Transactions of the Association for Computational Linguistics* 5 (2017): 529-542.

Gübür, Koray Tuğberk. "Create Word Cloud with Masks in Python." Holistic SEO, January 25, 2021. https://www.holisticseo.digital/python-seo/word-cloud/.

"Geopandas 0.10.2+0.g04d377f.Dirty." GeoPandas 0.10.2+0.g04d377f.dirty - GeoPandas 0.10.2+0.g04d377f.dirty documentation. https://geopandas.org/en/stable/.

Hollander, Justin B., and Henry Renski. Measuring urban attitudes using Twitter: An exploratory study. Lincoln Institute of Land Policy., 2015.

Hollander, Justin B., Henry Renski, Cara Foster-Karim, and Andrew Wiley. "Micro quality of life: Assessing health and well-being in and around public facilities in New York City." Applied Research in Quality of Life 15, no. 3 (2020): 791-812.

Kessel, Patrick van. "Interpreting and Validating Topic Models." Medium. Pew Research Center: Decoded, September 24, 2021. https://medium.com/pew-research-center-decoded/interpreting-and-validating-topic-models-ff8f67e07a32.

Kooten, Pascal. "Contractions 0.1.68." PyPI. https://pypi.org/project/contractions/.

"Learn to Use Twitter Data for Academic Research | Twitter Developer Platform." Twitter. Twitter. Accessed September 1, 2021. https://developer.twitter.com/en/use-cases/do-research/academic-research/resources.

The Matplotlib development team. “Visualization with Python.” Matplotlib. https://matplotlib.org/.

Microsoft. Visual Studio Code 1.66.2 [Software]. 2022. https://code.visualstudio.com.

“Monday, March 29, 2021.” March 29, 2021: History, News, Top Tweets, Social Media & Day Info. Sapro Systems LLC. Accessed March 14, 2022. https://www.wincalendar.com/Calendar/Date/March-29-2021.

Mueller, Andreas. “WordCloud for Python Documentation.” WordCloud for Python documentation - wordcloud 1.8.1 documentation. https://amueller.github.io/word_cloud/.

“Natural Language Toolkit.” NLTK :: Natural Language Toolkit. https://www.nltk.org/.

NumPy. Open-source Software. 2005. https://numpy.org/.

“Pandas.” pandas documentation - pandas 1.4.2 documentation. https://pandas.pydata.org/.

Pigott, Fiona, Jeff Kolb, Aaron Gonzales, and Jim Moffitt. “Searchtweets-v2 1.1.1.” searchtweets-v2 - PyPI. https://pypi.org/project/searchtweets-v2/.

“Plotly.” Plotly Python Graphing Library. https://plotly.com/python/.

Plunz, Richard A., Yijia Zhou, Maria Isabel Carrasco Vintimilla, Kathleen Mckeown, Tao Yu, Laura Uguccioni, and Maria Paola Sutto. "Twitter sentiment in New York City parks as measure of well-being." Landscape and urban planning 189 (2019): 235-246.

“Re - Regular Expression Operations.” re - Regular expression operations - Python 3.10.4 documentation. https://docs.python.org/3/library/re.html.

“Scikit-Learn: Machine Learning in Python.” scikit-learn: machine learning in Python — scikit-learn 1.02 documentation. https://scikit-learn.org/stable/.

Sinek, Simon. “How Great Leaders Inspire Action.” Simon Sinek: How great leaders inspire action | TED Talk, September 2009. https://www.ted.com/talks/simon_sinek_how_great_leaders_inspire_action.

“Sklearn.metrics.cohen_kappa_score.” sklearn.metrics.cohen_kappa_score — scikit-learn 1.0.2 documentation. scikit-learn developers. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.cohen_kappa_score.html.

“String - Common String Operations.” string - Common string operations - Python 3.10.4 documentation. https://docs.python.org/3/library/string.html.

“TextBlob: Simplified Text Processing.” TextBlob: Simplified Text Processing - TextBlob 0.16.0 documentation. https://textblob.readthedocs.io/en/dev/.

"Textwrap - Text Wrapping and Filling." textwrap - Text wrapping and filling - Python 3.10.4 documentation. https://docs.python.org/3/library/textwrap.html.

"Twitter API Documentation." Twitter. n.d. https://developer.twitter.com/en/docs/twitter-api.

"Warnings - Warning Control." warnings - Warning control - Python 3.10.4 documentation. https://docs.python.org/3/library/warnings.html.

"WHOQOL - Measuring Quality of Life| The World Health Organization." World Health Organization. World Health Organization. Accessed January 4, 2022. https://www.who.int/tools/whoqol.

"Why_1 Adverb - Definition, Pictures, Pronunciation and Usage Notes." why_1 adverb - Definition, pictures, pronunciation and usage notes | Oxford Advanced American Dictionary at OxfordLearnersDictionaries.com. Accessed January 5, 2022. https://www.oxfordlearnersdictionaries.com/us/definition/american_english/why_1.

Zivanovic, Slavica, Javier Martinez, and Jeroen Verplanke. "Capturing and mapping quality of life using Twitter data." GeoJournal 85, no. 1 (2020): 237-255.