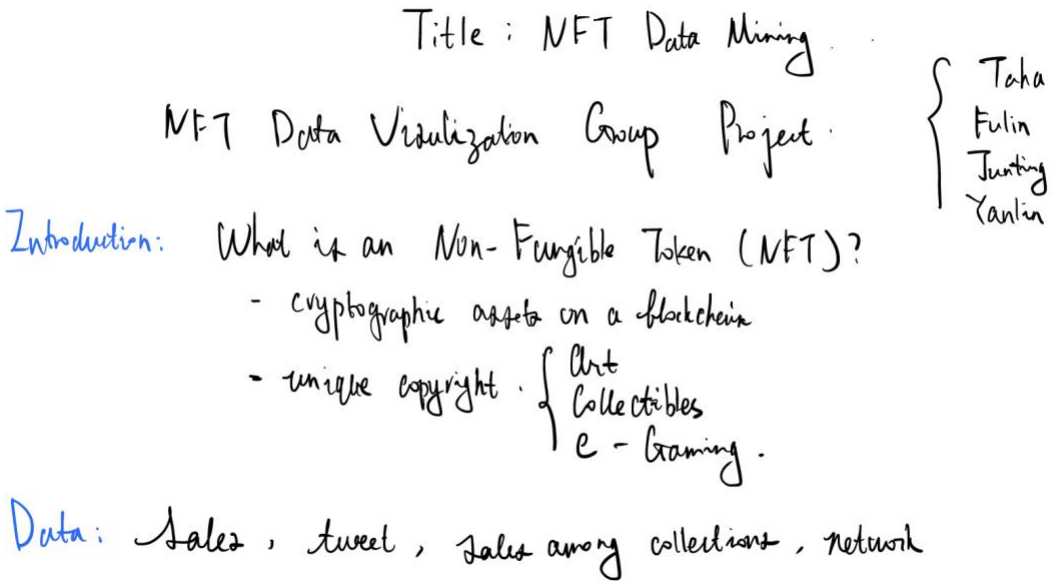
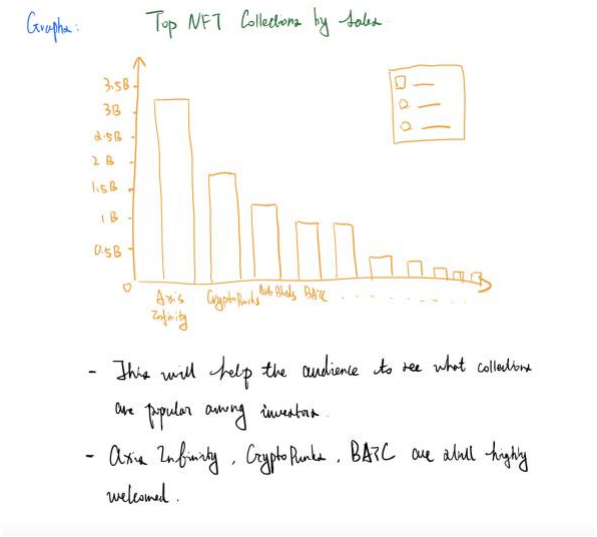
Group Y – NFT Data Mining Process Book

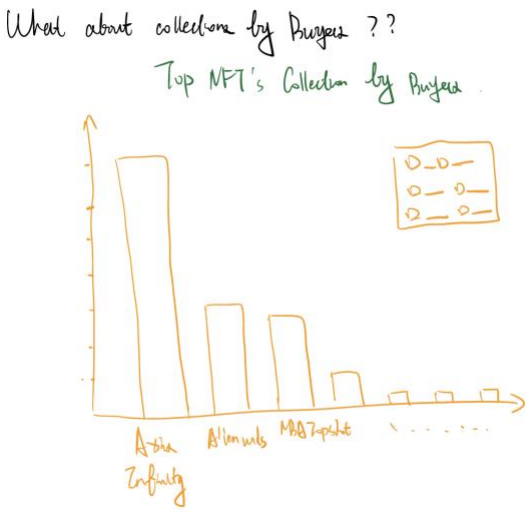
Yanlin Zhang, Junting Zhou, Taha Saeed, Fulin Wang

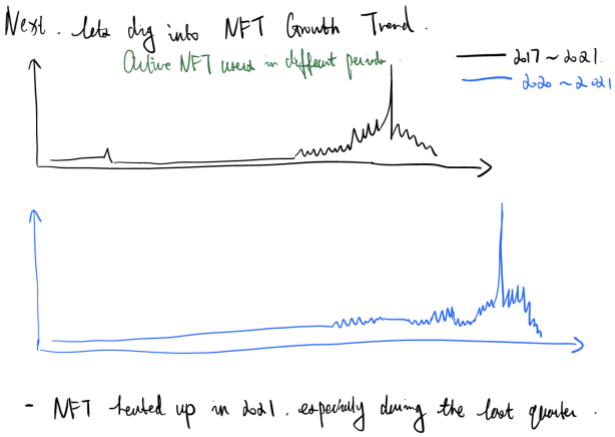
Our aim is to provide a comprehensive analysis of NFTs in terms of sales and user growth, networks on certain blockchains, and tweets relevant to NFTs. There was actually not enormous amount of NFT-related data out there, as compared to databases of different cryptocurrencies and blockchains. We thought of incorporating some visualization analyses purely on cryptos but that would be a whole other story and we'd deviate from our initial focus. So we discarded some of the initial datasets we collected like the Ethereum and Solana data.

The first part, as an intro, was not a big problem, since there're mostly descriptive bar charts comparing Top NFT collections and time-series plots visualizing user and price changes over time. There are a few existing datasets available on Kaggle to work with. Below are a few sketches based on the data, and we made real graphs quite smoothly.

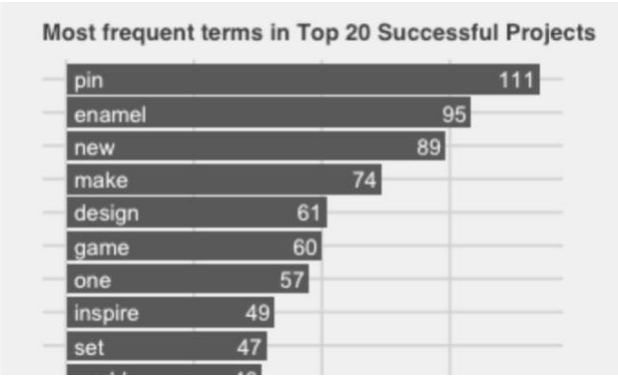








When it comes to making networks, we got the problem that existing NFT network data lacks the columns we wanted. Aside from NFT buyers, their holdings and the general sales volume, we hopefully also wanted the data to have some additional info to work with. There's only one good dataset available, which is in SQLite format. We had no clue how to scrap those kinds of data manually, but it's good to see that one dataset works well.



In terms of the text analysis part, the execution process was relatively smooth since the tough part was already done in the third assignment. There are a few bumps though, and one of them was that we could not find a way to horizontally adjust the text and frequency bar labels in ggplotly, and the default label position is quite off, kind of half way crossing the top of those bars (I cannot put it there unless I reverse engineer the existing codes, but you get the idea). This graph is what Yanlin did in HW3, but I was not able to adjust the label position like this in the project. However, since I was pretty

much doing everything in an interactive manner, we decided to drop the frequency labels and move the word labels back along the vertical axis. That's pretty much it. This part was not the real pain.

The real obstacles were in finding well-formatted tweet data and converting the codes to a nice publishable format. Previously we were trying to scrap real-time twitter tweets with NFT hashtags, ideally for a 3-4 year period and containing all sorts of additional useful columns like retweets, geolocation, mentions, etc. In that way we'd be able to do maps like choropleth or heatmap, text network, time-series graphs, etc. However, through the basic Twitter v2 API we were only able to scrap a limited number of tweets in an extremely short window. I tried to scrap 10,000 tweets and they were all from one user. We also did not find a way to specify relevant parameters while scrapping. The elevated API needs to be applied and by providing thorough explanations, usually took forever to get any response. Therefore, we had to proceed with some Kaggle tweet datasets that only includes basic features like the text, username, post time, and so on. Fortunately, the existing dataset itself was quite fine, but there was just less room of visualizing a variety of stuff.

We also took quite a lot of time to render the output. Websites are nice, but we think the slide presentation is going to be more succinct. Each slide would be embedded with an interactive plot, and everything would be easy to follow along, just like Dr. Brambor's class slides. We tried the xaringan theme at first, since the syntax is easier to work with. However, the external images and html files, like the pyLDAvis and networkvis interactive graphs, were not visible in the output, unfortunately. We then switched to reveal-js, which I personally think is not as nice-looking and quite cumbersome in terms of inserting different CSS codes, hardship to change output globally, etc. We used some of the output settings from Dr. Brambor's slides and specified the output size and text alignment for every individual slide. It was just a tedious process of searching syntaxes, trying out different visualization options, adjusting details, back and forth. Thankfully the final rendered output looks nice.