Kick-ing the Competition

Data documentation template adapted by Hannes Datta.
Originally based on Gebru, Morgenstern, Vecchione, Vaughan,
Wallach, Daumeé, and Crawford. (2018). Datasheets for Datasets.

1. Motivation

1.1 What primary research question, business problem, or knowledge gap motivated the creation of this dataset? How does the dataset offer insights into new phenomena, contribute to developing new models, or streamline gathering essential information? Why is this dataset valuable to the broader research community or industry stakeholders? Please provide a description about your research context.

We decided to investigate a new and upcoming streaming platform, Kick. This website came to be after a community guideline change from Twitch in which Twitch decided to sharpen their rules regarding gambling. They decided to ban certain sites from being streamed and that streamers could not be sponsored by gambling organizations to promote their website. One of these websites was Stake, which is a large online gambling website. They get a lot of traffic to their website from streams, and could lose this because they were not allowed to be streamed anymore on Twitch. Thus, Stake decided to launch their own live-streaming platform Kick. Kick now is fighting for market share with Twitch and YouTube. They invest a lot of money to 'buy' streamers over from Twitch and YouTube to solely stream on their platform. This is an interesting opportunity to see their growth in a mature market, and how well they perform over time. This itself could be a very interesting phenomenon to capture and research on how they managed to compete with the 'big boys' Twitch and YouTube.

However, what is also interesting is how Kick is related to the promotion of gambling since the platform is founded by an online gambling organization. For background information, Twitch banned Stake because Twitch believes that Stake artificially increases the odds of streamers to lure people into spending money on their website whilst these people then would have less favorable odds. Furthermore, the relationship between Kick and Stake can be found because a lot of streamers within the gambling categories are sponsored, which is shown in their titles where they have '#ad' or '!STAKE' displayed. As a clarification, '!STAKE' allows people to type this in the streamer chat, which in return will show an automated message telling people about Stake or how they can sign up to Stake. A lot of the time '!STAKE' is paired with a certain amount of money to sign up. We of course cannot confirm nor deny Stake to be biased towards streamers to lure people into gambling on their website.

Our data set will help us show how much Stake influences the gambling category with sponsorship, and how the gambling category is promoted on the website with recommendations. This can show insight into how the organization operates. Furthermore, we capture the biggest categories and how much they are viewed. Analyzing this over time can give insight into the growth of Kick and where this growth is coming from. All in all, our data sets can help us discover or improve the understanding of underlying phenomena of Kick's growth in a mature industry and therefore help boost ecological value (Boegershausen et al., 2020).

---

* https://arxiv.org/abs/1803.09010

1.2 The various websites and APIs you assessed relevant to your data context, why did you choose your specific data source? Discuss the research fit, efficiency of resource use, and any other factors that made it emerge as the best choice. What extraction methods did you consider, and why did you choose the method you used? Were alternatives to web scraping evaluated? How did you ensure the scope of your data context was appropriate to maintain validity and identify any other valuable information that might be relevant? Please motivate why you selected this particular data source.

We chose to collect our data on Kick, but there were also other big streaming platforms available to collect data on. The alternatives considered were Twitch, YouTube, DLive and YouNow. However, we quickly came to the conclusion that Kick would be more interesting to collect data on since it is still a relatively new platform that is undergoing its development into the industry. Furthermore, Twitch and YouTube have strict regulations surrounding gambling, whilst Kick is founded by a gambling organization. This makes Kick a more interesting site to collect data from. Twitch, DLive, and YouTube could provide high quality data since they both have an API, which is something that Kick does not have. However, we decided that the richness of new data surrounding Kick would be more interesting to investigate. This is especially the case in comparison to DLive and YouNow because the structuring of the Kick site provides more rich and easier collectable data. Finally, because we are interested in how the changes in Twitch influence other streaming platforms, Kick is preferred over DLive TV and YouNow because the formatting of the Kick website is nearly identical to that of Twitch. Viewers and streamers that leave Twitch need to get used to the layout of a new site. However, because Kick is a near exact copy of Twitch this threshold is lower. This makes Kick more accessible compared to DLive and YouNow.

Table 1: Scraping alternatives to Kick

	API	***Organizational Life Cycle***	***Policing of Regulations***	***Fit with Research Question***	***Gambling Category***	***Number of Users***	***Page Scraping Structure***	***Similarity to Twitch***
YouTube	Yes	Mature	Strict	Moderate	Yes	High	Good	Low
Twitch	Yes	Mature	Strict	Moderate	Yes	High	Good	-
DLive TV	Yes	Growth	Moderate	Good	Yes	Moderate	Moderate	Moderately High
YouNow	No	Growth	Moderate	Moderate	No	Moderate	Moderate	Moderate
Kick	No	Early Growth	Moderate	Good	Yes	Moderate	Good	High

1.3 Who created this dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?

This Data was compiled by Team 1 of the Online Data Collection and Management Course of Tilburg University (Spring 2024).

1.4 Who funded the creation of the dataset? If there is an associated grant, please provide the name of the grantor and the grant name and number.

There was no funding.

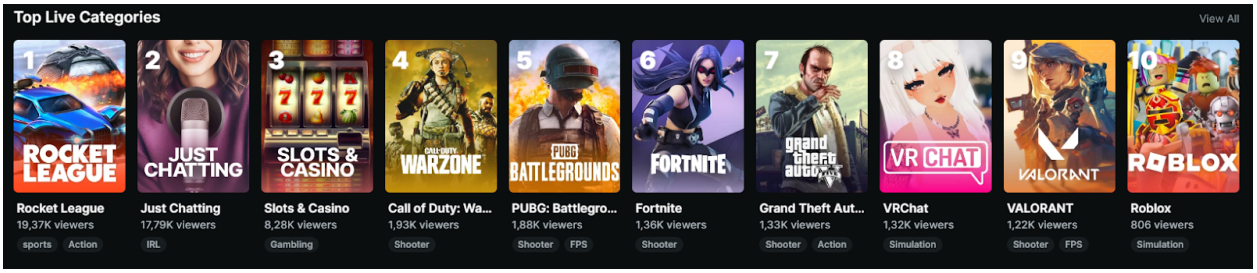
2. Composition

2.1 What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)? Are there multiple types of instances (e.g., movies, users, and ratings; people and interactions between them; nodes and edges)? Please provide a description.

The collected data sets will be compromised of string, factor and numeric values.

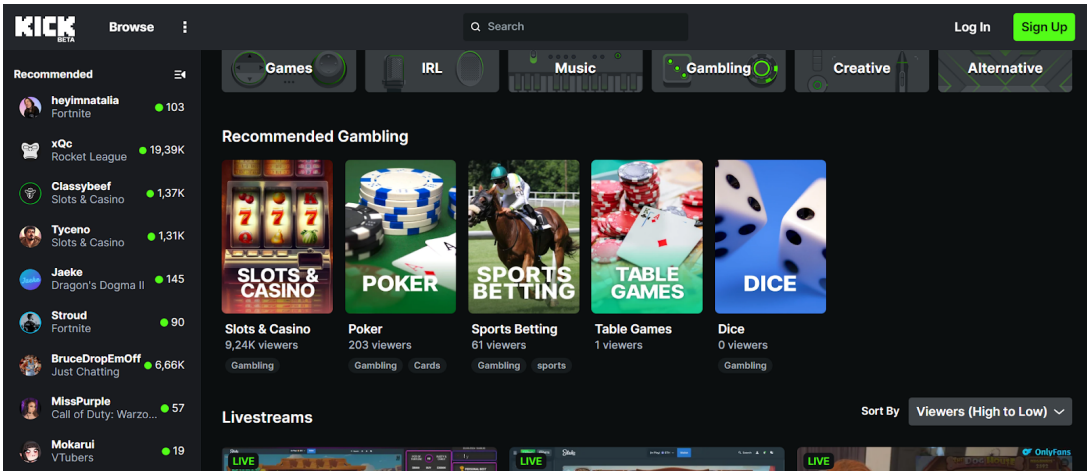
- First there is the "top_category" data set which includes information on the top ten categories streamed on Kick. This information is shown on Kick's home page (see Image 2.1 below). The values in the data set include: "Category", "Viewercount", "Subcategory", "Timestamp_of_extraction".

Image 2.1. The main page of Kick, top_live_categories

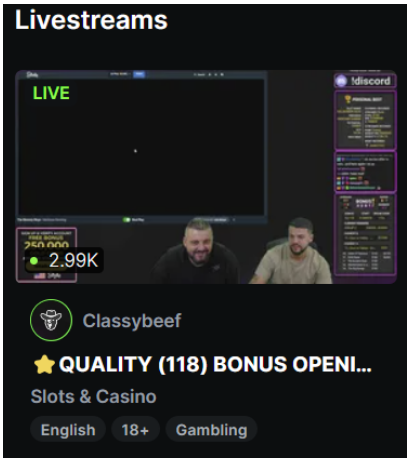


- The second data set "recommended_streams" contains data on: "Streamer_secret", "Category", "Viewercount", and "Timestamp_of_extraction". This data set has nine different values each time being scraped. This data set comes from the gambling categories page, but this information is also shown on the main page.

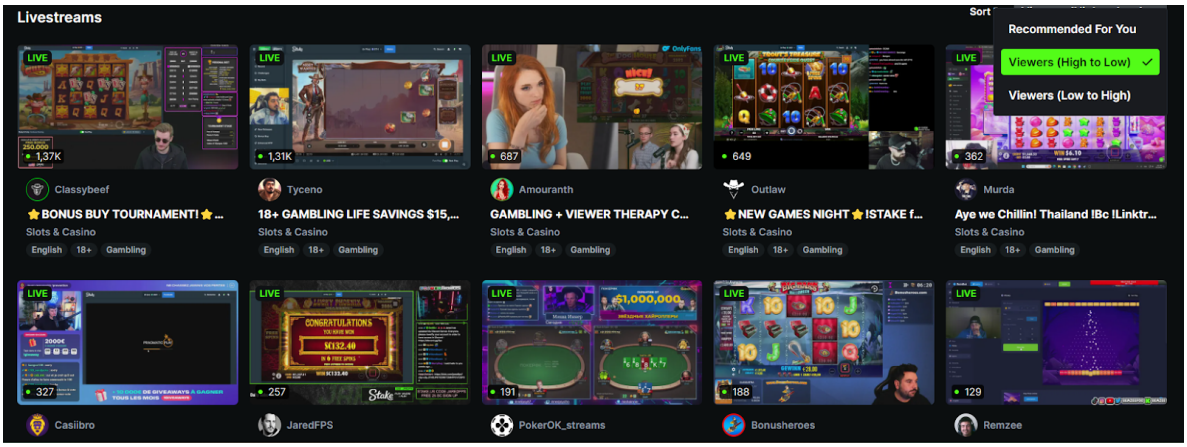
Image 2.2: The gambling category page, with recommended_streams on the left side of the page



- The third data set "streamer_information" contains information on, at maximum 50, streams in the category in ascension of most- to least viewed. The data columns contain information on "Streamer_secret", "Title", "Language", "Viewercount", "Subcategory" and "Timestamp_of_extraction". It is expected that there is overlap between data sets. More specifically, it is expected that the most viewed streams in "category_streams" are also shown in "recommended_stream_in_category".



Images 2.3: The gambling category page *Image 2.4: streamer_info with viewers from high to low ^*



- The last data set "gambling_category" contains information about the five specific gambling categories. The data columns contain "Subcategory", "Viewercount", and "Timestamp_of_extraction". This information is extracted from Kick's gambling page.

Image 2.5 The gambling category page, gambling categories

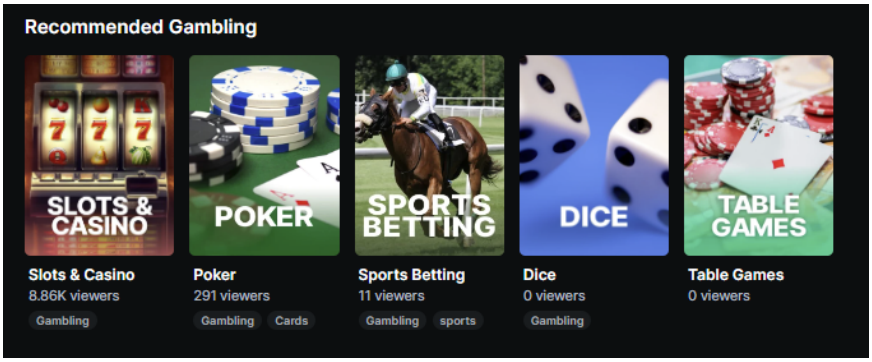


Table 2: The Data Set Variable Description

		Present in Data set			
Column Variable Name	***Content***	**Top_live_categories**	**Recommended_Streams**	**Streamer_Info**	**Gambling_Categories**
Category	The topic of the recommended category. E.g., Just Chatting; Slots and Casinos	✓	✓		
SubCategory_topic	The sub category in which the trending topic belongs. E.g., slots in slots and casino's	✓			✓
SubCategory_streamer	The game that is being played by the streamer			✓	
ViewerCount	The total number of viewers for that particular category at the timestamp_of_extraction	✓	✓	✓	✓
Timestamp_of_Extraction	The date and time on which the observation was collected formatted Year-Month-Day Hour-Minute-Second	✓	✓	✓	✓
Streamer_secret	The deterministic anonymized name of the streamer.		✓	✓	
Title	The title of the streams as configurated by the streamer			✓	
Language	The language in which the streamer is talking in during the stream			✓	

2.2 How many instances are there in total (of each type, if appropriate)?

For the data sets "top_category", "recommended_streams", "gambling_category", and "streamer_information" there are 10, 9, 5 and max 50 observations, respectively, available per 5 minutes. After running the data collection for 56 hours we achieved a total number of 6611, 6038, 3334, 33326 observations respectively.

2.3 Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set? If the dataset is a sample, then what is the larger set? Is the sample representative of the larger set (e.g., geographic coverage)? If so, please describe how this representativeness was validated/verified (e.g., to mitigate algorithmic interference). If it is not representative of the larger set, please describe why not (e.g., to cover a more diverse range of instances, because instances were withheld or unavailable).

- The data set "gambling_category" will be scraped in its entirety and does not require sub sampling. It encompasses all gambling categories on the site and does not change.
- The data set "recommended_streams" will be scraped in its entirety and does not require sub sampling. However, because they are recommended they may be influenced by an internet user's history. We made sure to use selenium to scrape the data, so there is no history or cookies that interfere with the data. However, there might be algorithmic interference in what streams Kick recommended for people to watch.
- The data set "top_category" was sampled on the available top live categories displayed on the main page. How many categories are displayed is based on the size of the browser. For example, with a zoomed-out browser (contr -, to -150) there are ten categories shown (see Image 2.6). Whilst with a zoomed-in browser (contr +, to 150) there are six categories shown (see Image 2.7). During data collection 10 categories were scraped each time. There are of course more than 10 categories available within Kick. However, we decided to only scrape the available categories on the main page, since most people will probably decide what they are going to watch based on what is available on the home page, when they don't have an idea on what/who they are looking for beforehand. Furthermore, the top 10 live categories are the most watched at the time of the data extraction, which is what we are interested in. Thus, we believe that validity was preserved.

Image 2.6 The zoomed-out web browser of Kick's main page

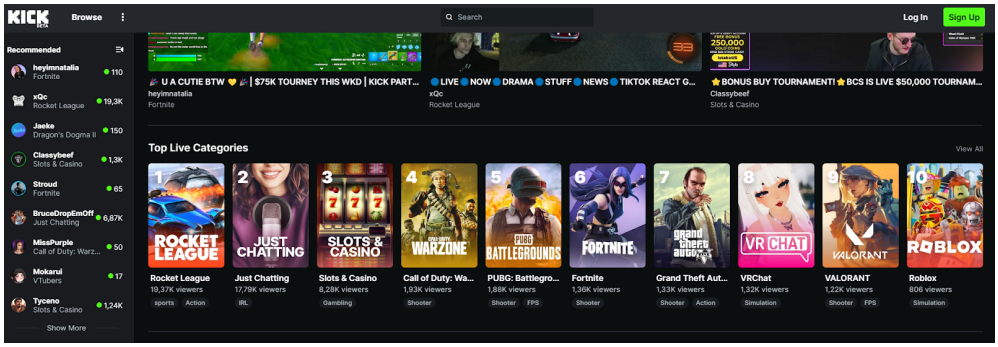
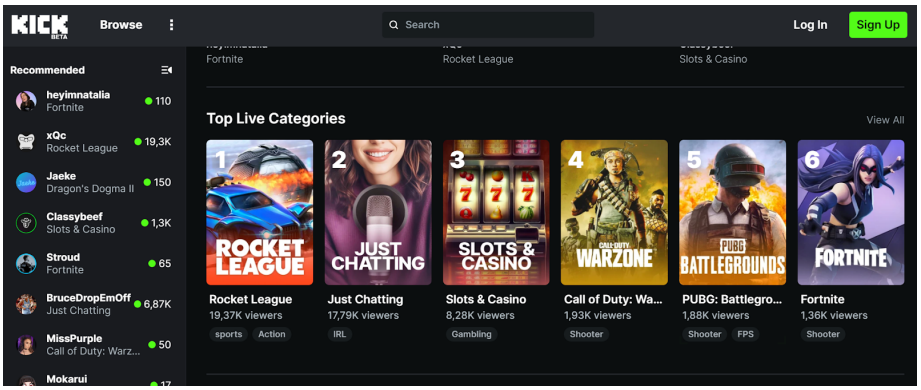


Image 2.7 The zoomed-in web browser of Kick's main page



- The last data set, "streamer_information", does have a subsample of a maximum of 50 observations in the order from high to low viewers. The number of streams can add up to well over 250 during peak hours but after 50 streams the viewership drops a lot with streams having less than 10 viewers at a time. Therefore, we decided to only sample the highest viewed 50 streams. We believe that this represents the larger pool of all the streams available within the gambling category and therefore also preserves the validity of the data extraction. Additionally, to provide researchers the opportunity to cross-reference the streamer data against recommended streams the decision was made to capture the most viewed streams over the lesser viewed ones since, most of the time, streams with a viewership higher than 10 viewers are shown in the recommended streams.

2.4 At what frequency was the data collected?

Data was collected every 5 minutes. Currently, the HTML of Kick allows for structural retrieval of the tables in which the information is nested. This means that the time it takes to collect the data is only 20 seconds for the 3 'shorter' scrapers and 40 seconds for the 'longer' scraper (streamer_information). The difference in time is due to the fact that the longer scraper has to load in 50 streamers. Thus, it takes longer to fully complete the scraping.

2.5 What data does each instance consist of? "Raw" data (e.g., unprocessed text or images) or features/operationalized variables? In either case, please provide a description.

As previously described in 2.1 all raw data will consist of strings that hold numeric, factor and string data. During the collection, these strings are separated and placed into their respective relevant variable columns. However, there is one exception concerning usernames. Given that the steamers usernames fall under the protection of privacy regulations, they were recorded with a safe untraceable directory using salted hashes. Please see table 2 in section 2.1 for a detailed description of each variable.

2.6 Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)? For example, if you submit multiple data files, can they be merged meaningfully? If so, please describe how these relationships are made explicit.

- The data set "streamer_information" contains anonymized streamer names (streamer_secret) that are deterministic. This allows the data set to be merged with the "recommended_streams" data set. To ensure no row-misalignment occurs these observations can be cross-referenced against the total number of viewers and the timestamp of collection.
- Due to the inclusion of category, and subcategory tasks, data can also be merged based on these categories. For example, the top live categories data set can be merged with the gambling categories.
- Additionally, because there is data available from trial scrapers the opportunity arises to merge observations within data set versions. This is possible due to the timestamp, and the deterministic streamer names.

2.7 Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)? If it links to or relies on external resources, a) are there guarantees that they will exist, and remain constant, over time; b) are there official archival versions of the complete dataset (i.e., including the external resources as they existed at the time the dataset was created); c) are there any restrictions (e.g., licenses, fees) associated with any of the external resources that might apply to a future user? Please provide descriptions of all external resources and any restrictions associated with them, as well as links or other access points, as appropriate.

The current data sets are entirely self-contained. There are no redirections, dependencies or links within the observations.

2.8 In collecting this data, how did you ensure research validity is balanced with the technical feasibility of collecting the data as well as any associated legal or ethical risks?

Both technical feasibility, and legal and ethical risks weigh heavily in the data collection design in contrast to validity. Because the collected data is about people doing things, several layers of protection have to be installed to ensure privacy of streamers. For example, anonymizing streamer names. However, to maintain internal validity, the streamer names have to be anonymized consistently. For example, a streamer name with the name XYZ, should always get the same anonymized value 123, and not 123, 678, and 345. Yet, some concern does remain with regards to a stream title. To ensure that researchers can investigate title patterns with regards to bonuses and trigger words, the titles are left untouched. This means that all content provided by a streamer in the title is not censored in any way. This means that any offensive or otherwise sensitive information remains in the title content of the stream. Though this is not ideal, it does allow researchers to also investigate these harmful behaviors and their impact in online platforms to perhaps even improve platform safety. Hence, the choice was made to leave stream titles as is, and validity is still heavily respected. In contrast, validity trade-offs were made for the sake of technical feasibility. Ideal collection would contain minute by minute observations. Yet, we decided that it was unlikely that streamer and viewer behavior would change so drastically on a minute-by-minute time frame. Additionally, because we run four scrapers, we wanted to ensure that the scraper does not crash due to overlapping scrapping

commands. Therefore, instead of a minute-by-minute window, we decided on a five-minute window and to compensate for potential data loss we would increase the duration of the collection.

2.9 Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor patient confidentiality, data that includes the content of individuals' non-public communications)? If so, please provide a description.

The dataset contains anonymized usernames. To ensure privacy of the streamers the usernames are being anonymized with the method salted hashes. Usernames are considered confidential because it is possible to trace it back to a person and their personal information. The remainder of the data should not contain personal information. However, the exception is stream titles. According to the terms and conditions of Kick, streamers are not allowed to use offensive, harmful or personal information in their streams in any form. However, streamers do occasionally breach these rules and put that information in their title names. For example, there are observations where the titles contain names of other streamers that joined the stream.

2.10 Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety? If so, please describe why.

The collected data sets are viable to contain content that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety. This is because the streaming titles were not transformed during the collection. As a result, the title names may hold content that requires trigger warnings. The decision was made to leave the titles as they were because they allowed researchers unique research opportunities. First, the current data allows the opportunity to investigate the effect of mentioned giveaways. Second, they also allow researchers to investigate potential title patterns that may increase viewership. Last, given that these titles exist, the data set allows researchers to investigate watch patterns in response to these titles and may help policymakers improve their online policies.

2.11 Does the dataset relate to people? If not, you may skip the remaining questions in this section.

Given that the data set is related to streamers the next section addresses any concern on the subject.

2.12 Does the dataset identify any subpopulations (e.g., by age, gender)? If so, please describe how these subpopulations are identified and provide a description of their respective distributions within the dataset.

At best the current dataset is only able to differentiate between languages and streamer preferences among the general population of people with internet access. While there are age restriction warnings these are not necessarily enforced. Therefore, no other traceable sub population elements are contained in the data.

2.13 Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset? If so, please describe how.

Prior to anonymizing the streamer names these names could be traced back to individuals. However, these names are anonymized within the scraper and no actual names are stored in any database. Therefore, this information cannot be traced back to any individuals. While additional arguments can be made for the title of the stream, this title is not saved after streaming. This means that titles cannot be traced back to streamers, and thus individuals. No other traceable information is collected and thus of no concern.

2.14 Does the dataset contain data that might be considered sensitive in any way (e.g., data that reveals racial or ethnic origins, sexual orientations, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)? If so, please provide a description.

While the collected data does not directly contain personally sensitive information, it may contain untraceable personal statements or harmful opinions. Because streamers are allowed to write their own title, they may put harmful or personal content in their stream titles. The scraping code is altered to ensure the streamer names are untraceable, and thus the origins of these statements are untraceable. However, the chance titles contain personal opinions is still present.

3. Data inspection

3.1 Please provide meaningful summary statistics and plots. For example, the number of units per entity, means/SD for continuous variables, or frequency distributions for categorical variables. This part of the documentation is intended to illustrate (the richness of) the collected data.

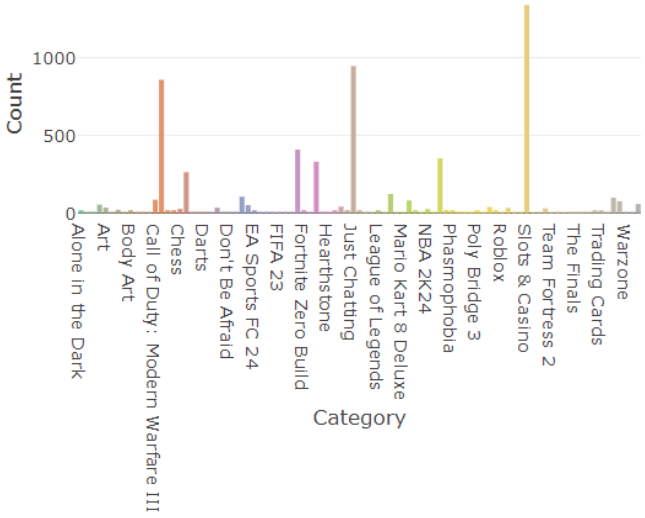
Please run our shiny data application if you would like to see more plots and make some plots yourself to get a better understanding of the data.

After data collection from date 20-03-2024 at 22:00 until 23-03-2024 at 06:10, four raw json files were generated. These json files contain information on 854 different streamers, 54 categories, 73 subcategories and 91 different games and topics. Streamers typically only play in one category and streams are available in 22 languages. The streamer with the highest number of views has 70,048.40 views on average with a standard deviation of 22,678.70 viewers. Viewers watch patterns are very dynamic. This is partially because the data captures the entire development of viewer count growth of a stream (see Image 3.1)

Image 3.1: Viewercount for a single stream

The data set "top_category", collected from the main page, has 612 observations that were collected in a fifty-six-hour period. There are four columns in total that have no missing values. On average, there are 17,990 viewers per stream with a standard deviation of 32,329.98. The least number of views at any point during the collection for a given category is 711, and the maximum is 235,700. The categories "In Real Life" and "Slots & Casino" are most frequently shown, and the average recommended category has 7,962.655 viewers with a standard deviation of 11,465.08. See Table 3.1 for the descriptives.

Image 3.2: All displayed Subcategories per Frequency



The data set "recommended_streams" has a total of 6,093 observations that were collected in a fifty-six-hour period. There are four columns in total that have no missing values. On average streamers have 3,393.00 viewers, with a standard deviation of 9,590.814 viewers. Across all the categories there is data on 354 different streams. "Slots & Casino" is the most frequently recommended category. Given that this category takes up 96 percent of the viewers in the gambling category this may not be as surprising. What is surprising is that regardless of the fact that Kick has over 75 categories, only 23 categories are recommended.

The data set "gambling_category" has a total of 3,335 observations that were collected in a fifty-six-hour period. There are three columns in total that have no missing values. The maximum number of viewers per category is 151,800 for the category "Slots & Casino", while the minimum is zero for all the remaining categories but poker. Within the category gambling, slot and casino streams make up 96 percent of the total streams, and the most commonly spoken language is English (67 percent) (Table 3.4). When keeping this in mind the spikes in viewership during the night times (CET, Amsterdam) come as no surprise (see Image 3.2). Additionally, the category "Sports Betting" spikes during American day times, and is quiet during other times which allows for unique and strong viewership data.

The last data set, "streamer_information", collected from the gambling page, has a total of 33,327 observations that were collected in a fifty-six-hour period. There are six columns in total that have no missing values. The maximum number of viewers for a given stream within gambling at any point during the collection time is 109,700 while the minimum is 5. On average there are 716.7 viewers with a standard deviation of 3,838.782. In total there are 854 different streamers. More than ninety percent of the streamers broadcast in only 1 category while the remaining 10 percent stream only a maximum of 2 categories. When grouping the average amount of viewers per streamer we see that the top 10 streamers account for 12 percent of the total amount of gambling viewers, and that these streamers gain viewers more quickly compared to less popular streamers.

Image 3.3: Viewership per Gambling Category

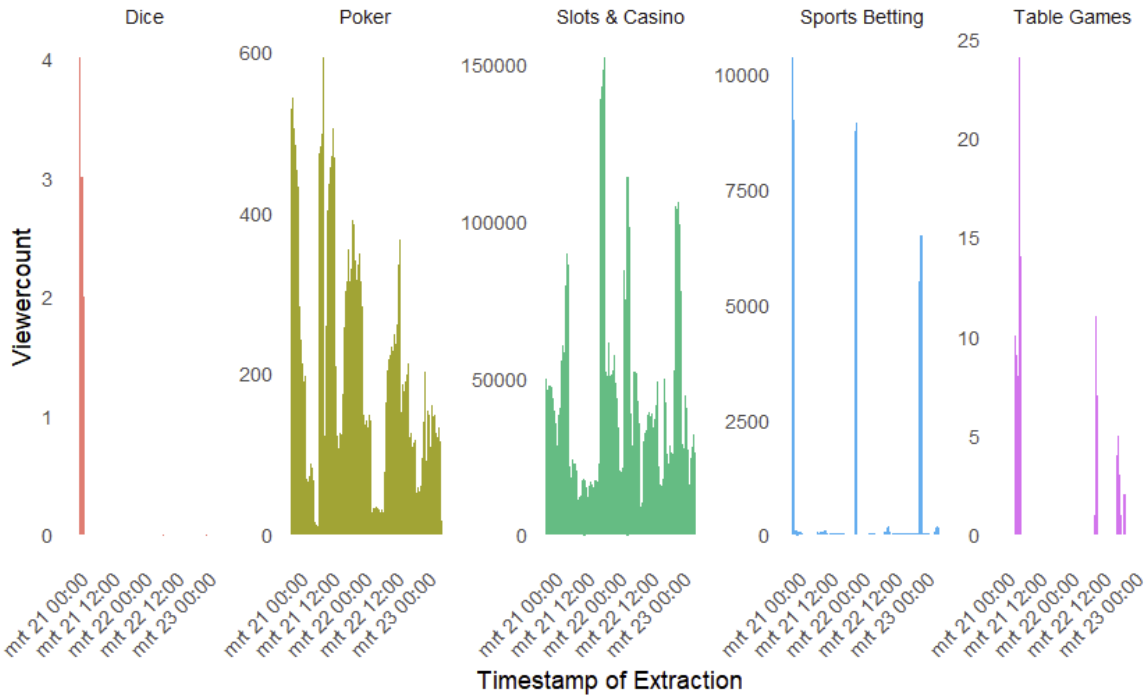


Table 3.1 The descriptives for the data set Top_Category

	Top Categories on Main Page
Global	
Collection - Date	*56 hours*
Total Number of Observations	*6,612*
Total Number of Columns	*4*
Number of Missing Observations	*0*
Viewer Descriptives	
Maximum Number of Viewers	*235,700*
Minimum Number of Viewers	*711*
Mean & SD Number of Viewers	*17,990; 32,329.98*
Category Descriptives	
Most Frequent Category	*IRL (925)*
Least Frequent Category	*ActionComedy (1); Hack and SlashRPG (1); HorrorAction (1); SimulationOpen World (1)*
Most Frequent Subcategory	*Just Chatting (668)*
Least Frequent Subcategory	*Euro Truck Simulator 2 (1); Journey to the End (1); Resident Evil Village (1); SpongeBob SquarePants: Battle For Bikini Bottom (1); The Finals (1); Titan Quest (1)*
Mean & SD amount of Viewercount per category	*8,540.23; 11,065.81*
Mean & SD amount of Streams per subcategory	*7,962.66; 11,465.08*

Table 3.2 The descriptives for the data set Recommended_Streams

	Recommended Streamers, Gambling
Global	
Collection - Date	*56 hours*
Total Number of Observations	*6,039*
Total Number of Columns	*4*
Number of Missing Observations	*0*
Viewer Descriptives	
Maximum Number of Viewers	*126,600*
Minimum Number of Viewers	*0*
Mean & SD Number of Viewers	*3,393; 9,590.81*
Streamer Descriptives	
Total Number of Streamers	*354*
Mean & SD Number of Viewercount per streamer	*698.18; 4,014.76*
Category Descriptives	
Most Frequent Category	*Slots & Casino (1342)*
Least Frequent Sub Category	*Crypto & Trading (1) Don't Scream (1) Farlight 84 (1) FIFA 23 (1) Poker (1) Roblox (1) Sports (1)*

Table 3.3 The descriptives for the data set Recommended_Category

	Recommended Categories, Gambling
Global	
Collection - Date	*56 hours*
Total Number of Observations	*3,335*
Total Number of Columns	*3*
Number of Missing Observations	*0*
Viewer Descriptives	
Maximum Number of Viewers	*151,800*
Minimum Number of Viewers	*0*
Mean & SD Number of Viewers	*7,850; 20,352.22*
Category Descriptives	
Most Frequent Category Displayed	*All (667)*
Least Frequent Category Displayed	*All (667)*
Mean & SD amount of Viewercount per category	*7,849.91; 17,264.48*

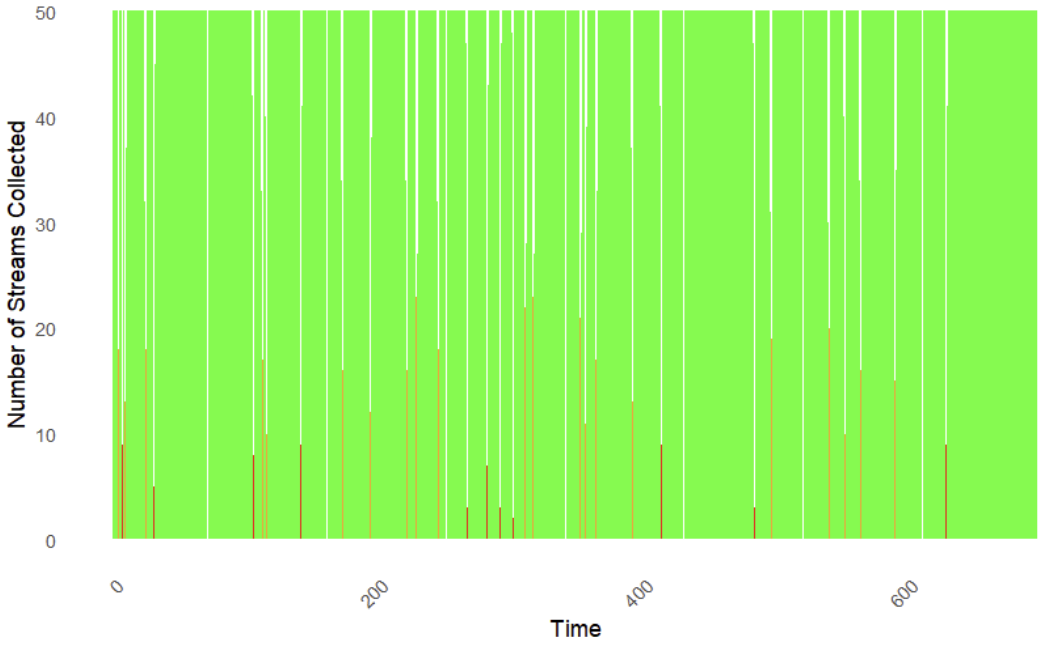
T*able 3.4: The descriptives for the data set Streamer_Info*

	Streamers, Gambling
Global	
Collection - Date	*56 hours*
Total Number of Observations	*33,327*
Total Number of Columns	*6*
Number of Missing Observations	*0*
Viewer Descriptives	
Maximum Number of Viewers	*109,700*
Minimum Number of Viewers	*5*
Mean & SD Number of Viewers	*716.7; 3,838.78*
Streamer Descriptives	
Total Number of Streamers	*854*
Mean & SD Number of Viewer-count per streamer	*595.53; 3,761.03*
Most Frequent Language	*English (22557)*
Least Frequent Language	*lietuvių kalbi (1)*
Category Descriptives	
Most Frequent Category	*Slots & Casino (32101)*
Least Frequent Category	*Table games (10)*
Mean & SD amount of Viewer-count per category	*407.45; 402.23*

3.2 Is any information missing from individual instances? If so, please describe why this information is missing (e.g., because it was unavailable). This does not include intentionally removed information but might include, e.g., redacted text.

There are no missing values in any of the data sets. The only form of missing instances are in the streamer_information dataset, because data is collected on a maximum of 50 streams, but does not always contain 50 observations for a given moment in time. For example, at 22:24:35 on 21-03-2024 there are only 2 live streams (see Image 3.4). Given that data is collected with a frequency of five minutes, and that observations are traceable across collections, these missing instances are unlikely to significantly impact the quality of the data.

Image 3.4: The number of observations collected per collection round.



4. Collection Process

4.1 Can you describe your technical extraction plan in such a way that another researcher or team could replicate your data collection process?

The extraction plan was to extract the data of Kick.com via a selenium web browser to ensure that there are no biases from our end in extracting the data. The biases in question are created by cookies or other preferences that could have been saved in our own web browsers. However, we could not find a way to overcome potential algorithmic biases. When the selenium web browser is created the top live categories on the homepage of Kick.com are going to be scraped (image 2.1). This happens via continuous cycling through all the different 'div's' within a higher class with the help of BeautifulSoup. The method was chosen to mitigate risk of path breakage (e.g., the path towards an object or value is changed and the scraper breaks down), and therefore increases internal validity. Additionally, because the layering of HTML elements was not always consistent this div cycling improved the efficiency of scraping.

Additionally, a second selenium web browser is opened to the gambling section of Kick.com. This decision was made for two reasons. First, it ensures the scraper bypasses CloudFlare. And second, it increases the speed of collection. On the gambling page three different objects are scraped: the different gambling categories, the recommended streamers on the left of the website, and the top fifty streamers sorted by high-low viewers (images 2.2-2.5). Again, the BeautifulSoup method that utilizes div's to cycle through is used.

All the scraped information of Kick.com will be stored into 4 different json files due to the varying length of the observations. (The top live categories has 10 observations, the recommended streams has 9

observations, the gambling categories has 5 observations, and lastly the gambling streamers has a maximum of 50 observations). These json files will be updated on a loop of 5 minutes.

4.2 Why did you choose a particular data extraction technology over others (for example, why did you opt for Selenium over BeautifulSoup for website scraping, or a specific package instead of self-coded requests for APIs)?

Given that there were no legal API's available at the time of collection, the choice to scrape data came naturally. When building the scraper, the package selenium was used to capture the html of Kick.com. This is because selenium is better equipped to handle the dynamics of the site and to bypass the CloudFlare protection of the site. Once the HTML was captured the decision was made to jump to the package BeautifulSoup because it handles raw HTML searching better compared to selenium.

4.3 If you encountered technical challenges during the scaling of your data collection, how did you resolve them? Please provide a clear explanation of the debugging process.

As we have stated in chapter 4.1, some classes were very strangely put together. An example of such a class is: "grid-fill place-content-stretch items-stretch gap-4 py-2 transition-all duration-300 md:overflow-hidden mobile-column [&>*]:!mb-2 [&>*]:lg:!mb-0 mobile-row". This clearly shows that there is a lot of possibility for the path to break since a small change class would result in the scraper to fail. Thus, we made the decision to find a coherent class that contains the information we needed to extract. Within this class we will cycle through all the 'div's' that are available and find the corresponding div that will extract the data we are after. For the streamer_info scraper not all valuable information was found within the div's thus we had to look elsewhere for this information. Some of the classes for the streamers were coherent to use to scrape information so we used those. Furthermore, some information was also stored in a 'span' so we decided to also cycle through them. So, for the streamer_info scraper we used a mix of div's, span's, and coherent classes to extract our data.

Next, the storage of the json files was reevaluated several times. The first and initial scraper was built in a way that made the script sensitive to row misalignment and included all four scrapers simultaneously. The next script that separated the scrapers had trouble unnesting the retrieved data from its respective soup object. Eventually, four final scrapers were written that saved each row in the nested list to the json and thus evaded the nested list problem.

Last, we originally decided to schedule our scraper with the help of windows task scheduler, which is a program that lets the user schedule tasks for the computer to perform. However, this came with some hiccups. First off, the scraper did not work and continuously broke down when asked to run. We came to the conclusion that this was due to Google Chrome being outdated which luckily was an easy fix. Second off, the task scheduler would stop at random after a couple of hours of running the scraper. We believed the cause of this was due to the fact that the computer was perhaps not 'strong' enough to continuously perform 5 minutes of for four different scrapers. So, we decided to change the interval to 10 minutes. Still, this only solved the problem partially since it would still sometimes skip intervals and therefore lose out on data that could have been scraped. Thus, in conclusion the scraper did not work properly. Therefore, we decided to come up with a new way to schedule the loop. We did this inside of our python scripts for

the scrapers. And to make sure the loop could be handled and every interval could be captured we decided to rent a cloud computer via Google Cloud. This luckily solved all the problems we encountered since the scraper ran without trouble in loops of 5 minutes.

4.4 What technical obstacles did you face during the data extraction process, and how did you overcome them?

Currently Kick is protected by CloudFlare which means that it intercepts users visits to the original site and redirects them via CloudFlare. While this improves the protection of Kick.com, it makes scraping increasingly difficult. For example, CloudFlare identified the quick clicking behavior of the scraper as non-human and then automatically blocked the scraper by asking for "validation of humanity". By installing strategically placed 3 second sleep-timers to simulate human clicking behavior this first obstacle was overcome. Furthermore, we also used the undetected chromedriver instead of the 'normal' chromedriver to make sure CloudFlare did not instantly recognize us a 'robot'.

The last problem was the saving of the json files. Due to the usage of BeautifulSoup the observations were unnested from the HTML as stored into an empty list. However, when running the scraper these lists remained nested within the json files. This problem was solved by writing the list items separately to the json file.

4.5 What measures or monitoring systems were in place to ensure and validate the quality of the extracted data? Can you describe how these monitoring systems functioned?

Due to the fact that we loop our scrapers within their respective python files, we failed to program a notification of some sort to notify that the scraper broke down. Therefore, the only measure of monitoring was visual inspection of the cloud environment the scrapers were operating in. This was done by copying the current json files when the monitoring took place. These json files were then examined to check if everything was still functioning. The check consisted of comparing the 'timestamp_of_extraction' to make sure that the scraper ran every five minutes, and looking at the number of extractions to make sure that the scraper still captured every observation and wrote them all to the json files.

For the task scheduler, two data monitoring systems were put in place. The first measure was an automated email message sent to the owner of the scraping computer. The data collection was executed via the windows task scheduler. However, if the task scheduler failed to execute its scraping task a command window was left open and the task scheduler failed to execute any other scheduled tasks. Yet, within the planning of the task scheduler there is an option to email the administrator if a task is failing. This allows the administrator to be automatically notified of trouble. Sadly, this task scheduler email notification is not completely reliable, therefore a second monitoring measure was put in place. This second method is simple human visual inspection. At random intervals the administrator checks if the scraper is still successfully running. While this is not the most efficient and advanced method of monitoring, it is the most reliable.

4.6 Can you specify the infrastructure you used for the deployment and execution of your data collection?

Originally, we executed the data collection on a 2017 HP windows laptop with an i5 processor. The computer was wiped of all data and programs with the exception of anaconda, python, google chrome and task scheduler. After installing the dependent packages, a task scheduler was set up that initiated four tasks that indicated the execution of the four python scraping scripts at a respective ten by two-minute interval. More specifically, each task scheduled referred to a .bat file saved in a chosen folder, that called on the anaconda script path to activate.bat, and then called a .py script in a chosen folder. In this .py script is the actual scraper. So, a .bat file was opened, the activate.bat function was called, a python script was executed. This happens every ten minutes for each of the four scrapers at a two-minute window (e.g., 12:00, 12:02, 12:04: 12:06; 12:10, 12:12, 12:14, 12:16). This ten-minute window enabled the laptop to execute the commands within the limits of its relatively outdated specs. However, as discussed in chapter 4.3 this did not work fluently and there were still a lot of hiccups.

So, eventually we rented a cloud computer via Google Cloud (Image 4.1 and 4.2). This enabled us to effectively run the scrapers, without having to worry about our own computers running the scrapers. We made sure to download python and move the scrapers to the cloud environment. After the set up was finalized we ran the scraper in four different command windows that were timed to be started approximately 1 minute apart from each other. This was done to make sure the cloud computer did not get overloaded with requests. Since the longest scraper takes 40 seconds to run the computer had time to finish the scraper before starting another. So, eventually the first scraper was started at around 22:12, the following scraper was activated at 22:13, then the following at 22:14, and the last one at 22:15, which allows the first scraper to run again at 22:17, etc. So, with this method we also preserved our original plan to extract data every 5 minutes.

Image 4.1: The virtual machine in Google Cloud

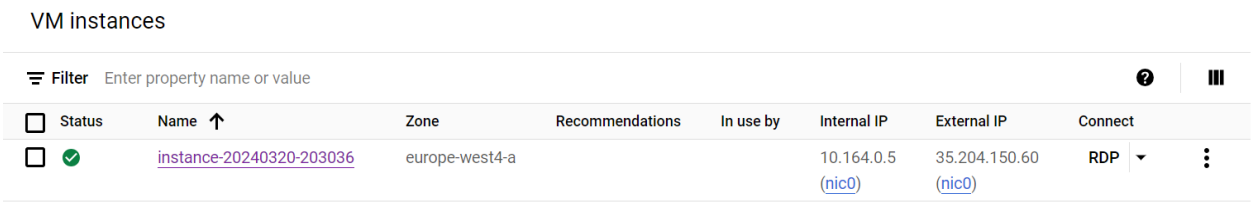
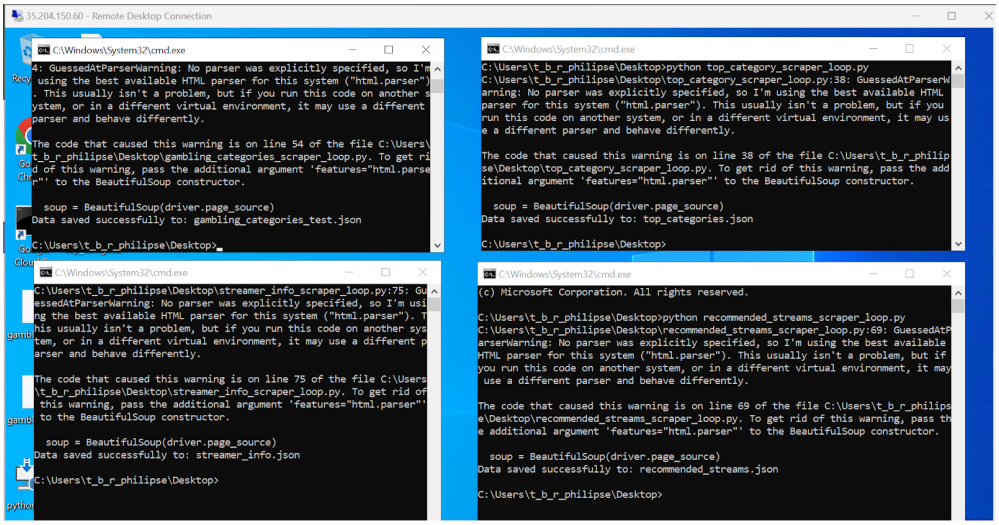


Image 4.2: Cloud computer environment



4.7 Over what timeframe was the data collected? Does this timeframe match the creation timeframe of the data associated with the instances (e.g., recent crawl of old news articles)? If not, please describe the timeframe in which the data associated with the instances was created. Please provide meaningful summaries, possibly referencing timestamps from log files.

The original attempt to run our scraper with the help of windows task scheduler started on 22-03-2024 at 10:00 and ended and restarted multiple times before deciding to pull the plug on 23-03-2024 at 12:00 and fully focus on the Google Cloud environment.

The data collection was started on Wednesday March 21 2024, at 22:12:19., and lasted until Saturday 22 March 2024, at 06:34:05. There is a slight difference in the start time and end time since we coded the scraper to loop for 56 hours. This is due to the fact that one scraper encountered a problem with starting up, which was solved with simply restarting the scraper. Therefore, this scraper started 20 minutes later and ended 20 minutes later.

4.8 Where was the data stored during the collection process? Detail any specific storage mechanisms or locations you utilized.

Each scraper python file was written so that each time new information is scraped it is written to an existing json file. This means that every instance of the scraper all data is saved, and should the scraper crash all observations until that point are stored at the designated json file. Given that there are four independent scrapers executed, four different json files are written independently of each other. These raw json files are stored on the cloud computer that runs the scrapers. *With regard to the extraction with the help of windows task scheduler, we made sure to write the raw json files to the same location the bat files and py scripts are located because this enables the administrator to limit the number of open windows should a scraper break down, and improve the efficiency of on the fly-computations.*

Additionally, the decision was made to upload the raw json files to Google Drive on a daily basis. This enabled all team members to have access to the data and check for any collection discrepancies should the administrator have missed something. Ultimately, we only kept the latest version of the raw json files since keeping all versions did not make sense.

After collection the raw json files were used for transformation. When the transformation is completed, the transformed files are stored in the same Google Drive location as the raw json files.

4.9 Who was involved in the data collection process (e.g., students, crowd workers, contractors), and how were they compensated (e.g., how much were crowd workers paid)?

The team consists of five students at Tilburg University that did not get paid for their work. However, we do have to give credits where credits are due since we received valuable help and support from our professor, H. Datta.

4.10 Were any ethical review processes conducted (e.g., by an institutional review board)? If so, please provide a description of these review processes, including the outcomes, as well as a link or other access point to any supporting documentation.

There were no ethical review processes conducted by for example an institutional review board. However, we did consider the ethicality of the scraper we were planning to make with our professor with regard to privacy, when it is ethical to extract data, and what kind of data is ethical to extract.

4.11 Does the dataset relate to people? If not, you may skip the remaining questions in this section.

The created data sets relate to people. More specifically, they relate to the streamers on Kick.

4.12 Did you collect the data from the individuals in question directly, or obtain it via third parties or other sources (e.g., websites)?

The streamer data was collected directly from the website. No third parties or other sources were called upon.

4.13 Were the individuals in question notified about the data collection? If so, please describe (or show with screenshots or other information) how notice was provided, and provide a link or other access point to, or other- wise reproduce, the exact language of the notification itself.

The streamers were not informed about the data collection. This is because personal information was anonymized during the collection process.

4.14 Did the individuals in question consent to the collection and use of their data? If so, please describe (or show with screenshots or other information) how consent was requested and provided, and provide a link or other access point to, or otherwise reproduce, the exact language to which the individuals consented.

Because streamers were not notified of the collection, there is no written consent. However, the terms and conditions of Kick state that all streamers must respect confidentiality and privacy rules, but streamers are not protected by the platform themselves.

4.15 If consent was obtained, were the consenting individuals provided with a mechanism to revoke their consent in the future or for certain uses? If so, please provide a description, as well as a link or other access point to the mechanism (if appropriate).

N/A

4.16 Has an analysis of the potential impact of the dataset and its use on data subjects (e.g., a data protection impact analysis) been conducted? If so, please provide a description of this analysis, including the outcomes, as well as a link or other access point to any supporting documentation.

No analysis was conducted. This is because the deterministic secret streamer names are not directly traceable to the actual streamers, and the impact on individuals is unclear.

5. Preprocessing, cleaning, labeling

5.1 Did you perform any pre-processing during the data extraction process? If yes, please provide specific examples and explain the reasoning behind each on-the-fly pre-processing step.

As a pre-process we label all our different variables within our scrapers. So, this ensured that we knew what each different variable meant and made it easy for others to also understand the data they are using. Next, we used deterministic username salted hashes transformations to respect the privacy of streamers. Last, a time stamp was generated for each round of collected and rendered to CET. CET was chosen over other time zones because the collection happens in a CET time zone and allowed us to easily understand when the scraper was most efficient during trials. This is the only pre-processing we believe that needed to be done to ensure valuable, coherent data sets.

5.2 After collecting the data, what additional pre-processing steps were undertaken?

After the data was collected the json files were cleaned in r and saved as .csv files. During this cleaning process it was ensured that:

- All columns that enabled merging had the same column name.
- The format of all columns was transformed to either: numeric, character, factor, or date-time.
- That the column viewercount had an easily computable value by removing the string “viewers”. Additionally, if there was a “K” that it was substituted for a multiplication of 1000.

After the computation, the data files were saved separately. The goal of these steps was to improve the handling of the data without changing the content of the observations.

5.3 Were any measures implemented to ensure privacy, such as anonymizing user data? Please describe the methods used.

To ensure the privacy of Kick's streamers, we anonymized streamer names during the data collection process. We generated hashed streamer names using python's hashlib package. Only the hashed streamer names were only stored in the respective json output files, to ensure that the same streamer name received the same anonymized hash name to track their respective behavior.

5.4 How did you address and clean out any implausible or erroneous observations in the dataset?

There were no apparent implausible or erroneous observations. All values were completed, categories were logically distributed and so are viewercounts. The only remarkable matter is that at times the scrapper appears to have failed to collect all available streamer_information data. This resulted in a sudden drop of streams. However, this took place only a handful of times (Image 3.2). The decision was made to leave those missing observations as is and not fill them with stream patterns of surrounding observations (e.g. taking the mean of viewercount). This is because the user of the data set should be able to decide how to deal with these missing observations themselves (e.g., they want to take a roll out mean instead of a simple mean).

5.5 Did you modify the data structure for long-term storage, like rearranging the dataset or renaming columns for clarity? If so, provide details on these changes and their rationale.

For the data set "top_category" the "timestamp of extraction" label was reformatted to "timestamp_of_extraction". This is because the other data sets were formatted as such and unifying this format allows the users of the data set to easily loop through and merge data sets based on the timestamp. Additionally, all viewercount columns were mutated to only show a numeric value. This entailed removing the string "viewers". But also replacing the letter "K" with a multiplication of 1000. This allowed for an easily usable column. No other columns were touched.

5.6 What potential threats or biases could arise from your pre-processing steps? Please elaborate on any risks associated with the modifications made to the data and how they might impact the dataset's integrity or utility.

It is unlikely that any threats or biases arose due to these steps. However, the bias that may arise is labeling bias, anonymization bias, and formatting bias. Labeling bias may arise because the labeling of the variables within the scrapers could limit the interpretation of the variables. For example, the columns named "category" and "subcategories" may be wrongly interpreted by data users and they might get the levels wrong. E.g., world of warcraft (game in subcategory) is part of world of warcraft (subcategory category) and belongs to the category gaming. Despite the deterministic salted hashes, anonymization bias may still arise. This is because, while these hashes still use a randomly generated hash, it may still be partially dependent on the original characteristics of the username. Though we have been unable to discover a pattern ourselves yet, does not mean it is not there. Last, the formatting of the viewercount may result in

some formatting bias. All viewercounts below a thousand show the exact number of viewers (e.g., 276). However, when viewership exceeds a thousand the number of viewers is rounded to the nearest thousand. (e.g., 2947 becomes 3000K). This is a partial loss of data but is also subjected to kick's rounding algorithm. By substituting the K for a multiplication of 1000, the assumption of kick's algorithm is continued across the column. This way of data wrangling may cause underlying biased patterns that may not be initially visible or realized by the user of the data.

5.7 Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)? If so, please provide a link or other access point to the "raw" data.

We saved the 'raw' data in the four different json.files with regard to their scraper, which are all stored and kept safe for future usage. The link to the 'raw' data is shown below:

https://drive.google.com/drive/folders/14zXwqtMzzy2eKbgvackKn_nxYQtn2_r-?usp=sharing

In this link are also all the other used materials to generate, process, clean and analyze the data.

5.8 Is the software used to preprocess/clean/label the instances available? If so, please provide a link or other access point.

Yes, we used R studio to preprocess/clean/label the data, which is a free and open-source program to write R in. These cleaned files can also be found in the same link as in provided in chapter 5.7.

6. Uses

6.1 Has the dataset been used for any tasks already? If so, please provide a description.

The data sets have not been used and are not publicly available due to Kick scraper preferences and the CloudFlare protection.

6.2 Is there a repository that links to any or all papers or systems that use the dataset? If so, please provide a link or other access point.

N/A

6.3 What (other) tasks / research projects could the dataset be used for? Provide a set of potential research questions or ideas for research projects.

The data sets can be used for looking into the evolution of a new streaming service or looking into the changes in gambling viewership and how gambling is promoted on a streaming platform. It can also be

used to compare Kick to their close competitors, so a potential research question can be "How does gambling viewership on Kick evolve compared to the streaming platform Twitch?".

In addition, since the scraped data contains the total number of views in the gambling categories, future research can answer the question "To what extent are the gambling streams watched compared to the In Real Life streams, and how is this relationship influenced by the time of the day". This can be considered interesting because Kick's most watched category in the 1st quarter of 2023 was gambling while In Real Life streams was the second most watched category on the platform (Statista, 2024).

Next to that, potential research may determine the relationship between the language of the gambling streamers and their viewers in combination with the time of the day. Analyzing this previous data can be interesting to see any connection between time zones in different countries and their viewership. However, it should be taken into account that some languages are spoken on different sides of the world, such as Spanish being spoken in Latin-America but also in Spain. And that English is a worldwide language and is not restricted to English speaking countries.

Lastly, research can address the difference in recommended streams on Kick compared to Twitch. This can be done by checking the variation in categories being recommended but also the viewership of the recommended streams. Potential results could determine that Kick shows less streams where video games are played but a higher proportion of In Real Life and gambling streams than Twitch.

We do have to address the fact that our scraper ran less days than we originally planned. So, if the scrapers could run for a longer period of time the empirical power would be way greater, since the current data sets only have the information of 3 days. But imagine what it would look like if it would be able to run for a year. We like to believe that there would be trends that can be discovered with such data sets.

6.4 Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses? For example, is there anything that a future user might need to know to avoid uses that could result in unfair treatment of individuals or groups (e.g., stereotyping, quality of service issues) or other undesirable harms (e.g., financial harms, legal risks) If so, please provide a description. Is there anything a future user could do to mitigate these undesirable harms?

In general, future data users are unlikely to be able to harm. However, the data set does provide the opportunity to discriminate based on language and title usage. While this is partially mitigated by the deterministic secret streamer names user should remain vigilant. This is especially the case for title content. Because the titles are generated by streamers themselves, streamers can put in all kinds of information that they would like. This included personal information. While the terms and conditions of kick state that streamers are not allowed to share personal, harmful or otherwise sensitive information, these rules are not always respected. This means that users of the data set may find themselves with data that could harm streamers. A way to handle this problem is to use a text analysis dictionary that identifies harmful words

and filters them out. Additionally, users should remain mindful before publishing graphs or chunks of information.

6.5 Are there tasks for which the dataset should not be used? If so, please provide a description.

The data set should not be used for investigating the viewercount on Kick in Turkey, Greece and China. This is due to the fact that these countries have blocked or partially blocked Kick. In Turkey, the government has blocked gambling from Kick. Kick has therefore made a version without gambling for the people in Turkey to still use their platform. In China and Greece Kick is blocked entirely.

References

1. Boegershausen, J., Datta, H., Borah, A., & Stephen, A. T. (2022). Fields of gold: Scraping web data for marketing insights. *Journal of Marketing*, *86*(5), 1-20.
2. Gerken, B. T. (2023, June 8). Twitch scraps ad changes after streamers leave platform. *bbc.com*. https://www.bbc.com/news/technology-65834521
3. *Twitch Safety Center*. (n.d.). https://safety.twitch.tv/s/article/Community-Guidelines?language=en_US#15IllegalActivity
4. Statista. (2024b, February 13). *Hours watched on most popular Kick content categories Q1 2023*. https://www.statista.com/statistics/1409660/top-live-streaming-categories-kick-hours-watched/