Data Management and Visualization Project

#CANTSLEEP Insomnia and Covid-19: an analysis of the causes of sleep disturbances in a pandemic

Vittoria Porta (matricola 861784), Alessandro Vincenzi (matricola 800608),Fabio Pasquale Lombardi (matricola 860550)

Università degli Studi di Milano Bicocca
Corso di Laurea Magistrale in Data Science
Professori A.Maurino, F.Cabitza

Table of contents

Table of figures

Introduction

COVID-19 is the infectious disease caused by the most recently discovered coronavirus, it was unknown before the outbreak began in Wuhan, China, in December 2019. The World Health Organization announced it as a new pandemic that is affecting many countries globally, in fact the disease spreads very quickly person to person through small droplets from the nose or mouth, which are expelled when a person with COVID-19 coughs, sneezes, or speaks. There have been an overall of 6,040,609 confirmed cases including 370,657[1] deaths.

The ongoing epidemic has radically changed the lives of people around the world, starting from the long lockdown periods that some countries have undergone, up to the substantial change in deep-rooted and daily habits. From one day to the next, the whole world had to become aware that even natural actions such as going to the workplace or going shopping could have been a danger to themselves and others.

Since the epidemic reached the western countries, the world seems to have stopped: even the largest cities, the giants of the world economy have surrendered and declared the lockdown, impressing an almost apocalyptic scenario on people's minds.

During the worst quarantine periods, where direct contact with anyone was considered dangerous, people of all ages approached the digital world of **social networks**: a safe space where they can express their opinions, their feelings and their fears.

The latest data show that the number of internet users and social media users around the world have both increased by more than 300 million over the past twelve months[2].

Insomnia and sleep disturbances

"Coronavirus, social distancing, and acute insomnia: How to avoid chronic sleep problems before they get started" was the Harvard T.H. Chan School of Public Health online forum where a series of of weekly sessions addressing the emotional and psychological effects of the pandemic were published[3].

---

[1] updated at 11:11am CEST, 1 June 2020, Wordl Health Organization website https://covid19.who.int/?gclid=EAIaIQobChMIuIK2v7vg6QIVDMqyCh2WvAP3EAAYASAAEgJIbfD_BwE

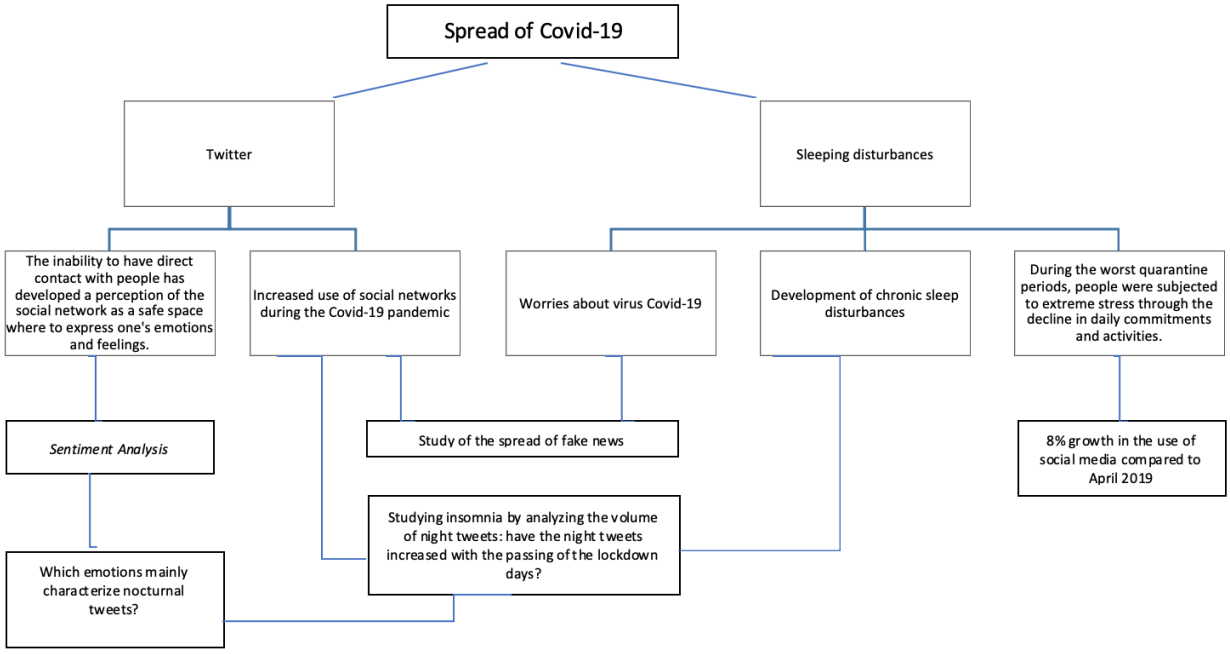
[2] DataReportal analysis indicates that 4.57 billion people now use the internet, an increase of more than 7 percent since this time last year. Social media users are growing even faster, up by more than 8 percent since April 2019 to reach 3.81 billion today. https://datareportal.com/reports/digital-2020-april-global-statshot

[3] The Harvard Gazete, Healt and Medicine, Insomnia in a pandemic. https://news.harvard.edu/gazette/story/2020/04/sleep-problems-becoming-risk-factor-as-pandemic-continues/

Experts say that during the quarantine periods, people were subjected to extreme stress caused by the sharp decrease in the commitments and activities that they used to face during the day. This change in habits has led the population to suffer from insomnia and consequently to develop depression and anger.

The aim of the project

The main Idea:



For the reasons above, the general aim of the project was analyzed the world of **Twitter** making assumptions and trying to answer specific questions based on the assumption that most of people suffered of sleep disturbances during the quarantine period:

- Has the volume of night tweets increased with the passing of the days in the worst months of the pandemic?

If that were the case, people would tweet more at night since they can't sleep.

- Which emotions mainly characterized night tweets?

In order to carry out the project and answer research questions we show the technologies that have been used.

Data sources

Since the project is focused on people's thoughts and fears during the pandemic, social network analysis is the most suitable type of analysis for the purpose. It consists in the study of interactions and communications between people on different trend themes, in fact, in these "virtual places" users feel free to be able to express all their emotions, finding comfort and support from other users.

In the pandemic period, Twitter was one of the social networks that had the greatest increase in use and for this reason it was chosen as the **primary source** of data for the project.

For the second part of the project, it was deemed useful to extract data from **additional sources** to complete the analysis and correctly answer research questions.

For the data enrichment part (treated in chapter "Data enrichment and transformation") were used three different data sources:

- The alphabetical list of all countries and capitals of the world[4]
- countries.csv dataset[5]
- List of U.S. state abbreviations[6]

For the data transformation part was used the list of tz database time zones[7].

Finally, it was useful in terms of visualization, to focus attention on tweets from the United States of America (a place where Twitter is known to be one of the most used social networks) and for this purpose an additional collection of tweets was integrated[8].

Data extraction

First of all we needed to create a **Twitter Application** and in order to do that we also required a twitter account in order to be able to extract data from the website we used Twitter API's granted by Twitter. Once the registration was accepted, Twitter gave us four important values that allowed us to consume the information: Consumer Key (API Key), Consumer Secret(API Secret), Access Token and Access Token Secret.

To extract data from the primary source (Twitter) has been used the library **Tweepy** for Python.

[4] available at https://www.countries-ofthe-world.com/capitals-of-the-world.html
[5] available at https://developers.google.com/publicdata/docs/canonical/countries_csv
[6] available at https://en.wikipedia.org/wiki/List_of_U.S._state_abbreviations
[7] available at https://en.wikipedia.org/wiki/List_of_tz_database_time_zones
[8] available at https://github.com/mykabir/COVID19/tree/master/data

This is a Python library for accessing the Twitter API, it permits you to extract real time tweets from all over the world through a Python code.
We created a class "listener" which was useful for listen the streaming of the tweets and then it was decided to extract tweets in *extended mode*, which means that all the tweets' documents were taken completed of all the available fields.

```
while True:

    try:
        auth = OAuthHandler(ckey, csecret)
        auth.set_access_token(atoken, asecret)
        twitterStream = Stream(auth, listener())
        tweepy.Stream(auth, listener, tweet_mode='extended')
        twitterStream.filter(track=['covid','coronavirus','wuhanvirus','#coronavir.
us','#stayhome','virus'],languages=['en'])
```

Through the function *filter* we were able to take only the tweets in English and with reference to the following keywords and hashtags: covid, coronavirus, wuhanvirus, #coronavirus, #stayhome, virus.
This filtering operation allowed us to ensure, in the first instance, a better **data quality**, avoiding to consider tweets that did not refer to the main research topic.

Data ingestion

Apache Kafka is a distributed streaming platform used to build real-time streaming data pipelines that reliably get data between systems or applications.
This technology is fast, scalable, durable and is able to takes continual streams of data from input topics, performs some processing on this input, and produces continual streams of data to output topics.
In order to link Python and Apache Kafka we imported **PyKafka** library and afterwards we create our **Kafka Topic** "Twitterdata2".

```
bin/zookeeper-server-start.sh config/zookeeper.properties
bin/kafka-server-start.sh config/server.properties
bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --
partitions 1 --topic twitterdata2

### consuming
bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --
topic twitterdata2 --from-beginning
```

A Kafka topic is a category (a logical representation of an object) to which records are stored and published; all Kafka records are organized into topics. Producer applications write data to topics and consumer applications read from topics.

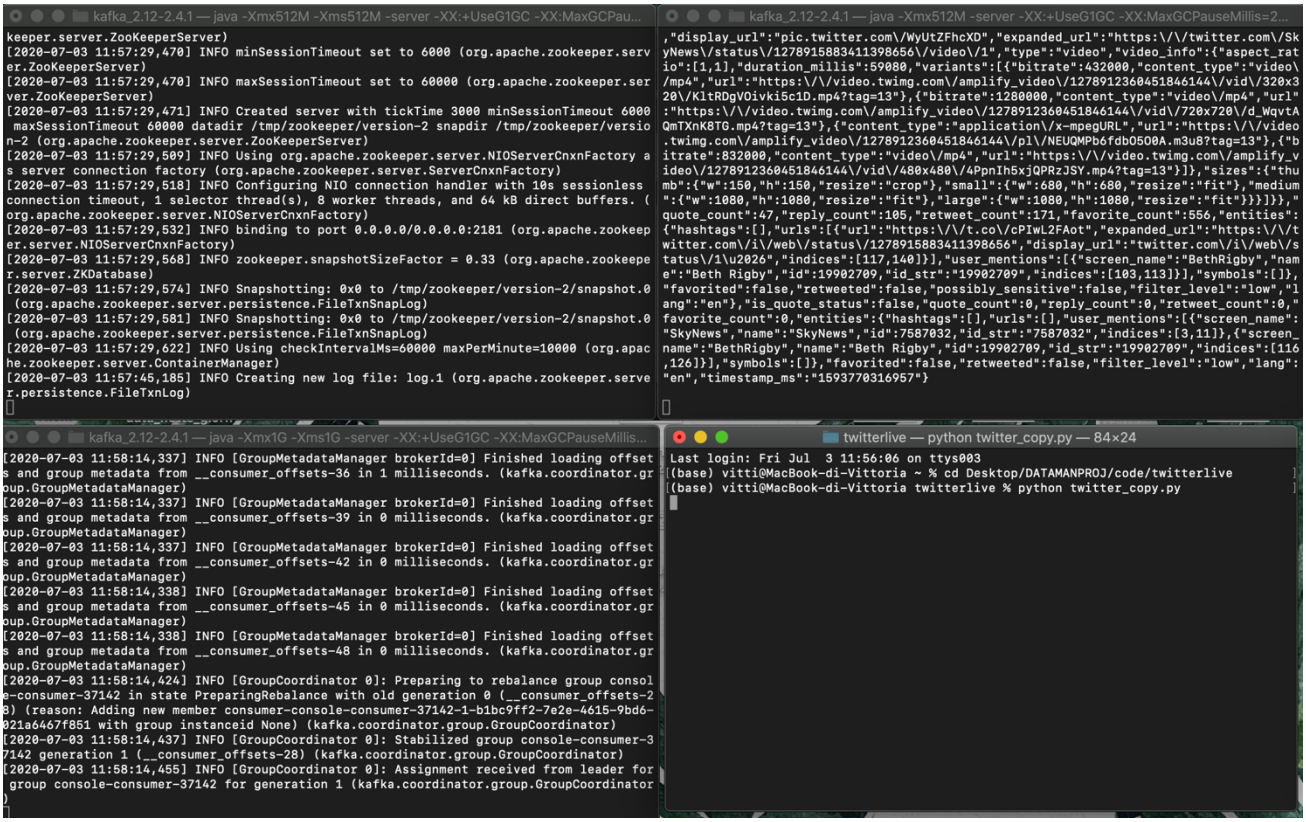
In simple terms, our Twitterdata2 topic contained all the Tweets.

At the same time we were consuming data from Twitter we wanted to produced them to our Kafka topic.

```
1. def get_kafka_client():
2.     return KafkaClient('127.0.0.1:9092')
```

We run Kafka locally on our machine and not on some clouds.

```
1. \class listener(StreamListener):
2.
3.     def on_data(listener, data):
4.         try:
5.             client = get_kafka_client()
6.             topic = client.topics['twitterdata2']
7.             producer = topic.get_sync_producer()
8.             producer.produce(data.encode('ascii'))
```



Connection to MongoDB

MongoDB is a non-relational, document-oriented DBMS. Classified as a NoSQL type database, MongoDB moves away from the traditional structure based on relational database tables in favor of JSON-style documents with dynamic schema (MongoDB calls the BSON format), making the

integration of data of some types of applications easier and faster. We chose to use this technology because extracted tweets were in a JSON format.

In order to connect Python to MongoDB we used **Pymongo** library in Python: we created a database called "streaming_tweet" and a collection called "streaming_1".

In order to easily manage data extracted from Twitter in MongoDB we use **Studio 3T** tool, which is a professional GUI available for all the operating system, that helps to explore data faster with features like query building, data exploration, aggregation and data comparison, import/export.

The data extraction and collection phase lasted from 11 April 2020 to 16 May 2020, reaching a volume of 2.02 GiB (extracted 3.96 GB).

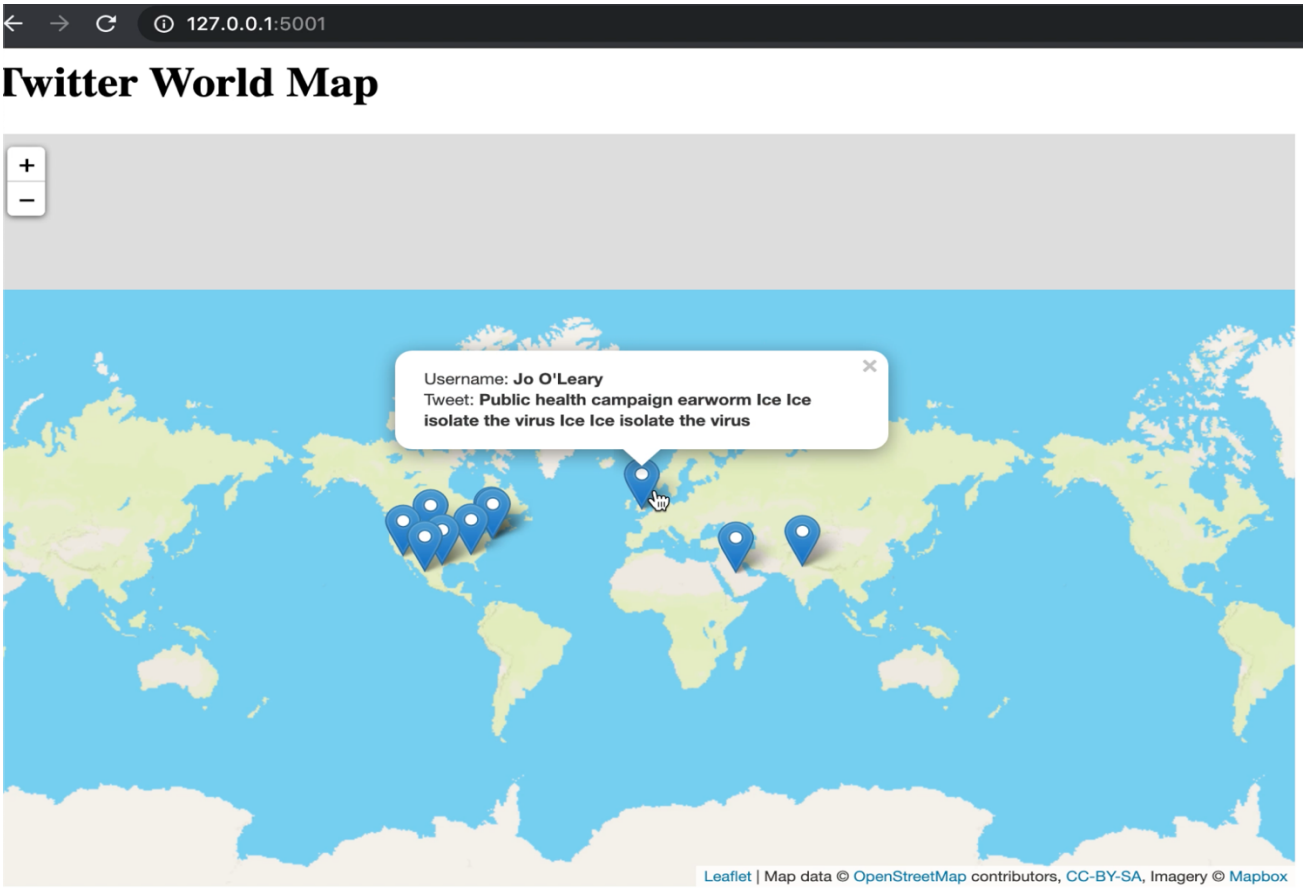
Preliminary analysis: real time

Since the data extraction process would take a long time, we wanted to have a tool that could visualize the data we were streaming in real time, allowing us to constantly check the quality of the data visually. For this reason we have created two interactive dashes using the Python dash library: the first one was a Live Twitter Map, which allowed us to check in a interactive map all the tweets that were collected and from which countries they were from. By clicking on the points, then, it was possible to view the text of the tweet as well as the username.

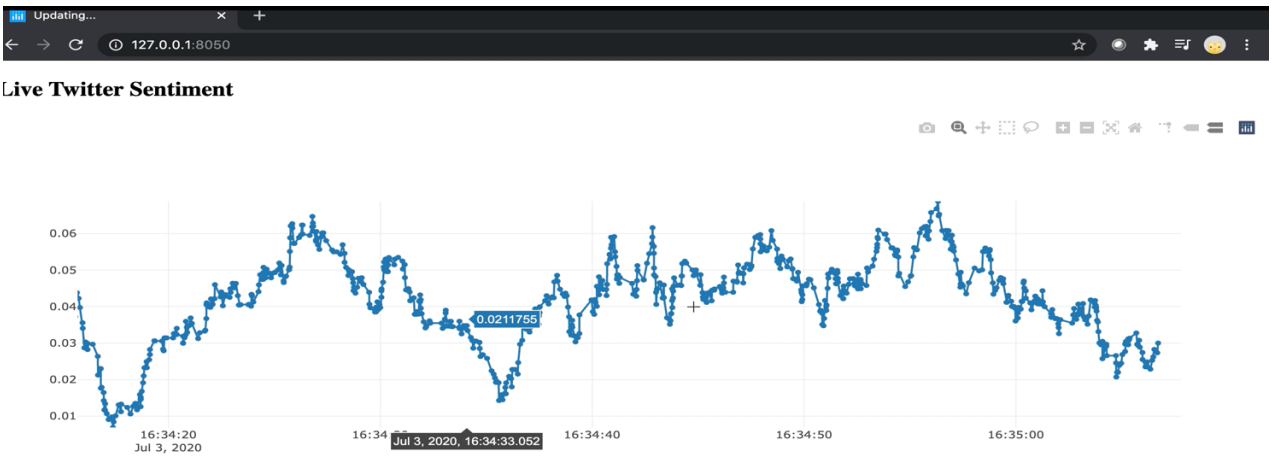
The application route developed with the **Flask** library for python was directly connected to the Kafka client, in such a way that we were able to spin up our consumer that loops through all the messages in the Kakfa topic.

```
1. @app.route('/topic/<topicname>')
2. def get_messages(topicname):
3.     client = get_kafka_client()
4.     def events():
5.         for i in client.topics[topicname].get_simple_consumer():
6.             yield 'data:{0}\n\n'.format(i.value.decode())
7.     return (Response(events(), mimetype="text/event-stream"))
```

To create the map we used Leaflet, which is a leading open-source JavaScript library for mobile-friendly interactive maps.



The second interactive dash that we created for a preliminary analysis was an interactive **real time sentiment line chart**: to calculate polarity score for sentiment we used vaderSentiment Python library (Valence Aware Dictionary and sEntiment Reasoner) which is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. It returns a compound score, that is computed by summing the valence scores of each word in the lexicon and then normalized to be between -1 (most extreme negative) and +1 (most extreme positive). Consequentially we were able to plot a line chart which updated in real time showing the polarity for all the streamed tweets.



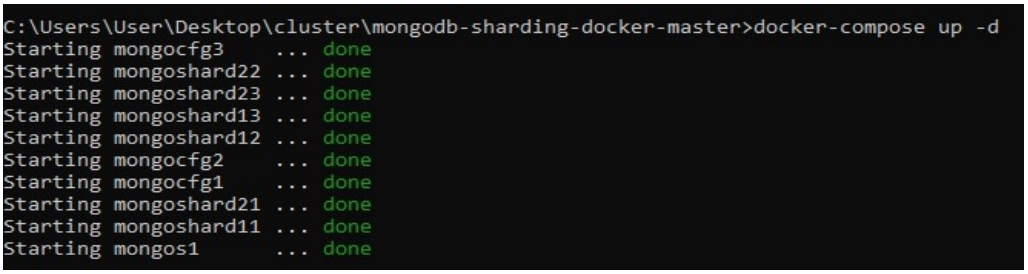
As we run the real time dashboards during the streaming period we created two demo in order to show how they works. They are available at this link:
https://drive.google.com/drive/folders/1T8rJVkNA05zm8P_yzRKrGoVWSA0IR4nG?usp=sharing

Sharded architecture

Since we collected a large volume of data, a sharded architecture has been built with the help of **Docker**: it is an open-source project that automates the deployment of applications within software containers. Using Docker to create and manage containers can simplify the creation of distributed systems, allowing different applications or processes to work autonomously on the same physical machine or on different virtual machines.
A hashed shard method was used for "_id" which allows for horizontal scaling.

The architecture[9] that has been deployed has:

- 2 Shard clusters in replica set (3 nodes for each cluster)
  - mongoshard11, mongoshard12, mongoshard13, mongoshard21, mongoshard22, mongoshard23
- Config cluster in replica set (3 nodes)
  - mongocfg1, mongocfg2, mongocfg3
- 1 Router mongos instance
  - mongos1



[9] First, the collection was saved in MongoDB, after which, using the --mongorestore command, it has been reinserted into a new sharded collection.

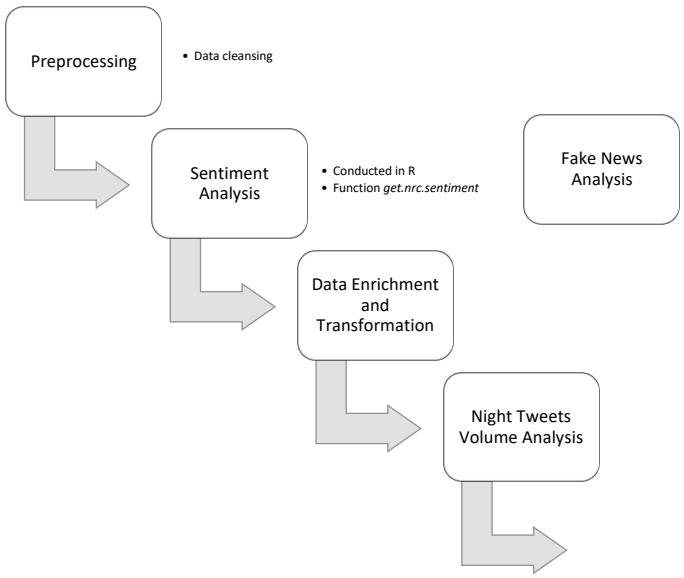
```
mongos> db.TwitterCollection.getShardDistribution()

Shard shard1 at shard1/mongoshard11:27017,mongoshard12:27017,mongoshard13:27017
 data : 1GiB docs : 174078 chunks : 31
 estimated data per chunk : 33.11MiB
 estimated docs per chunk : 5615

Shard shard2 at shard2/mongoshard21:27017,mongoshard22:27017,mongoshard23:27017
 data : 1.02GiB docs : 178055 chunks : 31
 estimated data per chunk : 33.87MiB
 estimated docs per chunk : 5743

Totals
 data : 2.02GiB docs : 352133 chunks : 62
 Shard shard1 contains 49.43% data, 49.43% docs in cluster, avg obj size on shard : 6KiB
 Shard shard2 contains 50.56% data, 50.56% docs in cluster, avg obj size on shard : 6KiB
```

Analysis



Data enrichment and transformation

To precisely answer the research questions and investigating the changes in night Tweets volume, we needed to enrich our data with external information: first of all, the fundamental field we were looking at were the “user.location” field and the “created.at” field. As long as we collected tweets characterized by the “extended” mode, we collected several raw data, that on one hand, did not contain the user's location (probably because they were produced by computer) and on the other, they contained non-existent or non-standard locations (for example NYC, instead of New York City). To perform the match between the "user.location" field of the original document and the standardized localization, we have taken as reference external datasets[10]:

[10] References in the paragraph "Data sources" pg.7

Secondly we manage the problem of the exact time of the production of the tweet: extracting twitter data returns the time of creation of the tweets as UTC time, for this reason we needed to convert the time into local time based on the place of origin of the users.

The data we were looking for had to correspond to all the states of the world, with their respective timezone names, capitals, state codes, the geographic coordinates of the states (latitude and longitude) and other important cities.

Subsequently, the dataset was expanded with additional cities among the most important and populous in the world to have a wider coverage, which will be discussed once the key problems and functions necessary for the resolution of the latter are presented.

Pre-processing and missing values treatment

The Pre-processing phase is an essential phase that uses data mining techniques to clean and transform the data available in a more suitable and agile format for subsequent analyzes.

First of all, using R Software, we conducted the **data filtering** phase, that is the process of choosing a smaller part of data set excluding repetitive or unnecessary observations and using that subset for analysis.

Starting from 1432 columns, they are summarized in the 7 most important variables for our research, which are divided into:

- 'created_at' : date in UTC referred to the moment of the tweet extraction
- 'id' = identification code of the user who sent the tweet, consisting of 25 characters or the integer representation of the unique identifier for this Tweet
- 'id_str' : same as 'id', but with with the particularity of being in string format
- 'text' : tweet text or decoded retweet in UTF-8.
- 'user.screen_name' : username
- 'user.location' : potential and presumed location of origin of the sender
- 'timestamp_ms' : a digital record of the time of occurrence of a particular event representing the date in UTC format referring to the moment of extraction of the tweet.

As for the columns, the data filtering process is carried out also for the rows, in fact duplicates are discarded, in particular those rows that share both the text of the tweet and the identification code-id, and the rows are deleted with the empty or missing tweet.

After dimensionality reduction phase is completed, the **data cleaning** process is applied through a function to clean tweet text and their localities of origin from all those elements that do not bring added value to the text and therefore are not significant.

Such as, retweet, that are Tweet repost presented with the prefix 'RT' followed by 'at sign' (that is this symbol @) and by user name of original author, punctuation, numbers, html links, unnecessary spaces, emojis, control characters (NPC, a code point that does not represent a written symbol) and other special characters, after all the characters are transformed into lower-case letters.

```
clean_tweet <- function (tweets) {
  clean_tweets = gsub('(RT|via)((?:\\b\\W*@\\w+)+)', '', tweets) # remove retweet entities    clean_tweets = gsub('@\\w+', '', clean_tweets) # remove @ word
  clean_tweets = gsub('[[:punct:]]', '', clean_tweets) # remove punctuation
  clean_tweets = gsub('[[:digit:]]', '', clean_tweets) # remove numbers
  clean_tweets = gsub('[ \t]{2,}', '', clean_tweets) # remove unnecessary spaces
  clean_tweets = gsub('^\\s+|\\s+$', '', clean_tweets) # remove non-space whitespace
  clean_tweets = gsub('[[:cntrl:]]', '', clean_tweets) # remove control characters

  clean_tweets = gsub('http\\w+', '', clean_tweets) # remove html links
  clean_tweets = gsub('https\\w+', '', clean_tweets) #remove another type of link

  clean_tweets = gsub('<.*>', '', enc2native(clean_tweets)) # remove emojis or special characters
  clean_tweets = tolower(clean_tweets) # lower case letters
  clean_tweets
}
```

A crucial step belonging to the pre-processing phase consists in expelling from the dataset all those lines that have missing values and therefore do not bring an essential added value for adequate research on the entire population.

This step is exactly the Missing Value detection and is applied in the analysis on the 'user.location' column.

Table 1

Variable Name	Rows number with NA	% rows with NA
user.location	117860	34.8 %

In order to transform the "user.location" field and matching that with two new variables (extracted from the external dataset, "Parole Chiave" and "timestate", we filtered only the columns that referred to the location and for each of them we counted the absolute frequency and the relative percentage frequency through SQL queries to choose which variable was the most appropriate to study.

This process is essential to derive the local date of the tweet at the time of its extraction, because the functions that will be used for this aim recognize a well-defined and outlined set of time zones, the same list that was present in the 'timestate' dataset.

Table 2

Variable Name	Absolute Frequency	Relative Percentage Frequency
user.location	225753	66.18%
user.time_zone	0	0%
geo	146	0.04%
coordinates	146	0.04%
place	2576	0.75%

Among the proposed variables, the best choice is 'user.location' but the percentages suggest that large audience have neither a location nor location enabled cell phones. These Tweets not show up via any publicly accessible twitter location search method and this represents a gap in the data.

Moreover the matching of a location in a user's profile to a location in the real world may not be entirely accurate and can cause conflicts to be examined.

So we wondered if there had been an event that triggered this dynamic and we found out that on 18 June 2019 Twitter announces that it removes the ability to tag user's precise location giving more power to the user in fact from that moment he/she could claim to be in any state.

In effect, the peculiarity of this variable 'user.location' is that it does not follow an orderly logic but it presents itself as a single city, a single state code, a single state, a combination of the previous ones, a merger of terms without spaces or, in the worst case, a fancy name. This introduces a lot of noise into the data.

Data quality assessment

To analyse the locations, we also assessed the presence of homonyms (same name for different concepts) that could give rise to any conflicts, both between states, between cities and between state-code.

Table 3

Conflict Variable	Homonym String	Conflict String 1	Conflict String 2
State	Georgia	Georgia, USA	Georgia, Europe
Cities	London	United Kingdom	Ontario
	Paris	France	Texas
	Toledo	Spain	Ohio
	Venice	Italy	Florida & California

	Naples	Italy	Florida
State Code	AL	Albania	Alabama
	AR	Argentina	Arkansas
	AZ	Arizona	Azerbaijan
	CA	Canada	California
	CO	Colombia	Colorado
	DE	Delaware	Germany
	GA	Gabon	Georgia
	ID	Idaho	Indonesia
	IL	Illinois	Israel
	MD	Maryland	Moldova
	ME	Maine	Montenegro
	MN	Minnesota	Mongolia
	MO	Macau	Missouri
	MT	Malta	Montana
	NE	Nebraska	Niger
	PA	Panama	Pennsylvania
	SC	Seychelles	South Carolina
	SD	Sudan	South Dakota
	TN	Tennessee	Tunis

In order to avoid these types of conflicts, in cases whose subjects are states and cities, the strings have been modified by adding the name of the continent to the states and the name of the country to the cities knowing that we would use the **R "grep"** function in such a way as to minimize the creation of inaccuracies.

While in cases whose subjects are state code the only way we thought was to go is to discard the state codes for both states, even if this meant losing a significant slice of correctly classified data.

In this way priority was given to the quality of the classification rather than the quantity of matched locations.

Therefore by evaluating these circumstances, functions such as Jaccard Distance and Edit Distance or Levenshtein Distance which measures the number of primary edits that would need to be made to transform one string into another were excluded because they become ineffective with very dissimilar strings.

Hence we have chosen to use a more elastic and flexible function such as 'grep' which takes in input a regular expression as first argument and an input vector as second argument and it returns a new vector with the indices of the elements in the input vector which could be partially matched with the regular expression.

The regular expression is formed initially by a list of states and cities for each state and secondly by a list of state code searched at the end of each string.

Subsequently, the match model has been expanded also using the 'state code' as it has been noticed that various locations are exactly composed of these state codes, for which the 'match' function has

been exploited, which combines identical strings for number and order of characters, to identify additional locations.

Whenever the algorithm finds valid matches it adds the identified status and the corresponding time zone in two new columns called 'status' and 't_state'.

To find out how many states had been matched the percentage of classified locations was calculated with respect to the column total and it amounts to about 71.4% of all locations.

It turns out to be a good result because, thanks to the 'grep' function, the error according to which a location is mistakenly matched with a different one or similarity is found between a location and a fantasy one is minimized.

Although it is a high percentage, it is necessary to reflect on the fact that on the opposite side we lose almost 30% of the information caused mainly by the freedoms imposed by Twitter about the location of the tweet, but also in a small part by an acceptable inefficiency of the algorithm.

To verify what has just been said and therefore appropriate and relevant outputs are provided, a random sample of 20 rows is extracted, consisting of 'user.location', 'status' and 't_state' to confirm or deny the presence of errors and inaccuracies.

Proceeding in the development and analysis of the dataset, we return to address the original problem, namely the conversion of the timestamp in possession with that in a different time zone.

For this request, we start from the removal of the last 3 characters of the timestamp that provide the milliseconds, not useful for processing, then the timestamp is transformed into date time, formed by 'year-month-day hour: minute: second' and finally it converts the datetime according to the connected timezone.

Then we obtained from the datetime column additional information such as the single date ('year-month-day') and the time, useful to distinguish the night from the day.

```
1. ###Convert UTC into correct local time zone
2. library(anytime)
3. library(lubridate)
4. td1$timestamp_ms=anytime(as.numeric(substr(td1$timestamp_ms,start = 1,stop = 10)),
   tz="UTC")
5. td1$t_state=as.character(unlist(td1$t_state))
6. td1$timestamp_local=lapply(seq(length(td1$timestamp_ms)), function(x) {format(with
   _tz(td1$timestamp_ms[x], td1$t_state[x]))})
```

We had to identified the night in the period that begins at midnight and ends at 5:59 in the morning, while the day starts from 6 and ends at 23:59.

Sentiment analysis

Among the various Sentiment Analysis algorithms we had preferred the function '**get_nrc_sentiment'** which calls the NRC sentiment dictionary to calculate the presence of eight different emotions and their corresponding valence, which however has its strengths and weaknesses.

It offers a rich variety of emotions, such as:

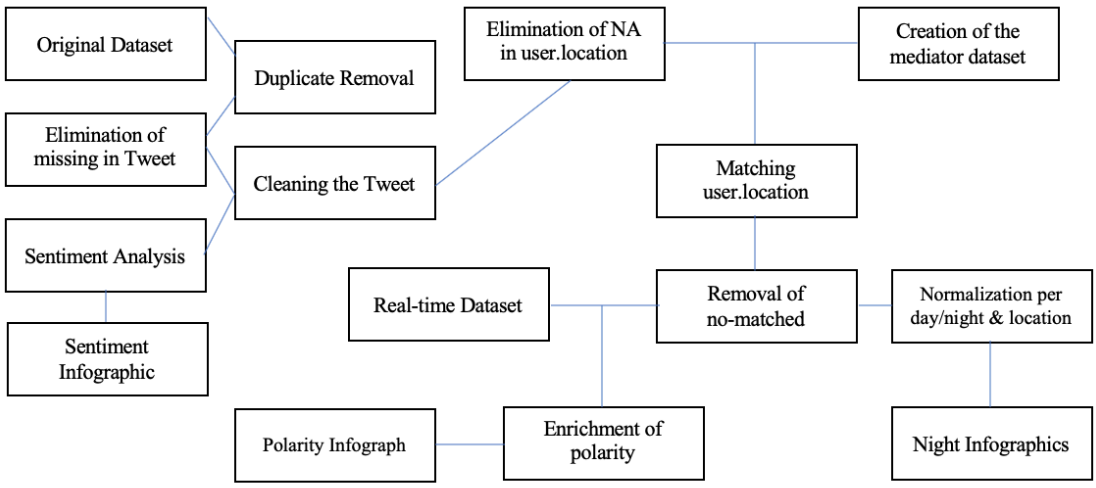
- anger
- anticipation
- disgust,
- fear
- joy
- sadness
- surprise
- trust

and it further provides the "negative" and "positive" polarity, on the other hand for a big data of our size it is computationally onerous.

Once the sentiment algorithm has been implemented, the 10 new columns about the emotions and polarity are added to the dataset and the relative percentage frequencies.

Than the rows by date are aggregated revealing which feelings were the most common during the Covid-19 pandemic.

Summarizing what has been said so far, a process flow diagram is presented to a whole all the steps that led from the original dataset to all its branches until reaching the ultimate goal, such as the infographic representation.



Retweet analysis

During the course of the project we became interested in various aspects of the world of twitter: such as, retweets, which represent the diffusion of information. This phase of the project was mainly carried out in R Studio.

With regard to retweets, the analysis was focused on the creation of tables designed to represent the spreading of information. The starting point was the extraction of the necessary data from the original collection.

The interesting fields for this analysis are: created_at, id, text, user.screen_name, user.location, extended_tweet.full_text. In order:

- Created_at is the field in which there is the date and time when the tweet was published and coincides with the time it was extracted. A number of changes have been made to this field to accommodate the format
- Id is a unique code for tweets
- Text is the field containing the text of the tweet, this field was very useful for obtaining the name of the account that was retweeted even if it has some limits: the text contained within it is truncated to 140 characters, to obtain the text complete you must use another field (extended_tweet.full_text)
- User.screen_name is the field in which the name of the account that published the tweet appears
- User.location is the field in which the location where the tweet was posted is present, this field has also undergone changes during the preparation of the table

Extended_tweet.full_text is the field containing the full text, this field has many empty lines. This feature was very useful for locating retweets.

At the same time, was created a .csv file *ad hoc* in order to standardize the location of the tweets: this file contains the list of all the states and all the federal states belonging to the USA; each state has its time zone, an identification code and a collection of keywords. The keywords were used to standardize the location of the tweets.

It was therefore impossible to carry out a precise analysis if for each location because, as we said before, there were several strings that were difficult to identify.

In addition, we created two functions that are able to clean tweets'text from useless punctuation, symbols and emojis.

Once all the necessary material was loaded, the data frame extracted from the original collection was divided into two subsets: a part containing the instances with the extended_tweet.full_text, and the other one that presents empty strings in this field.

```
solo_ex <- locrt %>% filter(extended_tweet.full_text != "")
min140 <- locrt %>% filter(extended_tweet.full_text == "")
```

After creating these two subsets it was important to consider tweets that had less that 140 characters in text:

```
sss <- grep('^RT', min140$text)
daun <- min140$text[-sss]
unione <- min140 %>% filter(text %in% daun)
```

The instances that presented 'RT' at the beginning of the text, were separated from those that did not have it, so it was possible to isolate the retweets from the 'original' tweets. The instances that had less that 140 characters in text were merged with the first subset created previously.

```
locrt <- rbind(solo_ex, unione)
semmm <- min140$text[sss]
semi <- min140 %>% filter(text %in% semmm)
```

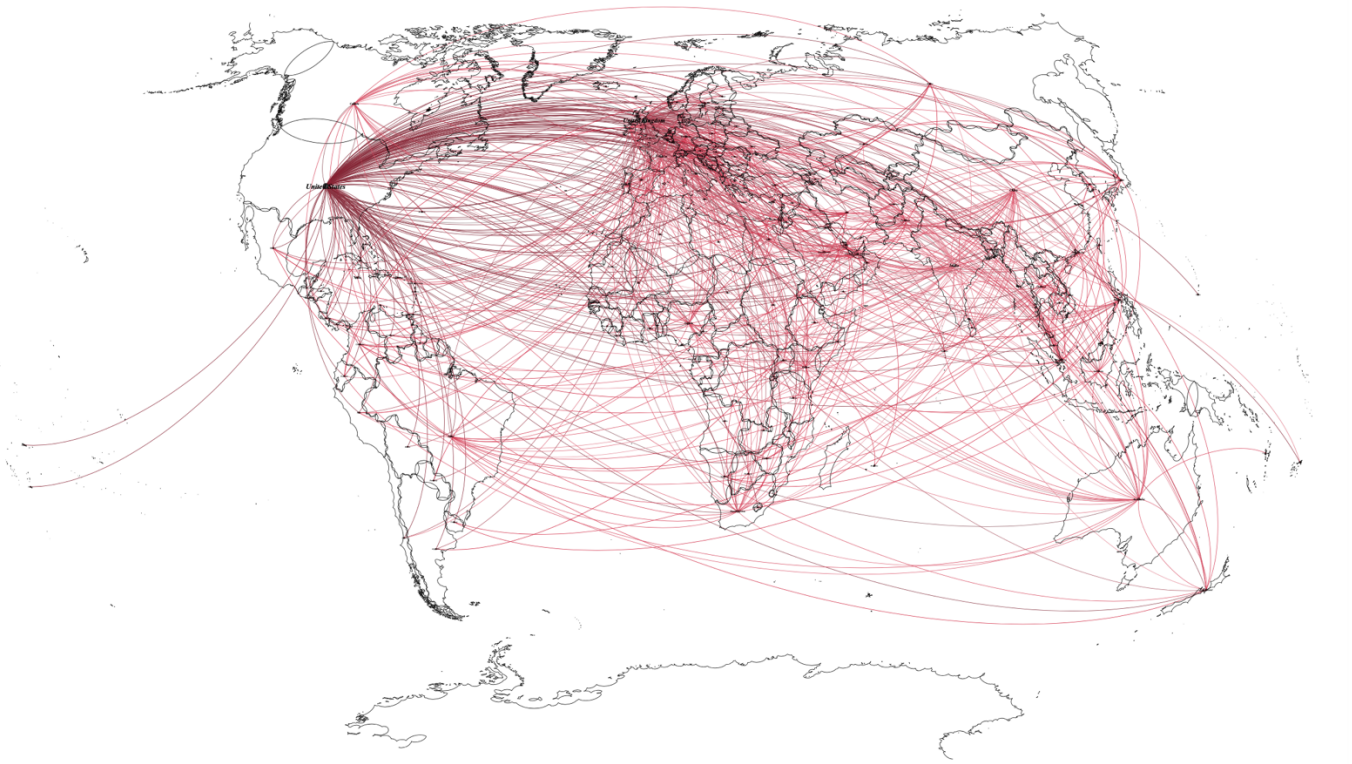
At this time we obtained to different tables, one showing tweets and the other one showing the retweets. To extract the username we used Res: it was noticed that the name of the account retweeted in the text always appeared in this way *"RT @ Mckeon92: What the…."* So the split was applied in the presence of the "@" and in the presence of the ":".
During the second phase, we standardized the location of the tweets in both tables. The keywords in the table described above are scanned and when a match is found the standardized name that refers to that location is entered. In the table with 'original' tweets, 72% of the rows were classified with the standardized location, the remaining instances were removed; in the retweets table the 70% of the locations have been identified and the unclassified instances have been removed.

To format the dates we used the library of R **lubricated**: it has a function that is able to automatically format the dates. The original date format was inconsistent so the result of the process process will be inserted in the 'created_at' column which now has the appropriate format. The two tables created at this point can be merged on the column that has the name of the account that has been retweeted. The final table at this point will have:

- Date: the same as 'created_at' column.
- Stato_retweetato: username that has been retweeted

- Loc_statoretweetato: location of the account that has been retweeted
- Ha_retweetato: username of who retweeted
- Loc_haretweetato: location of the username who retweeted

In order to plot the graph of tweet and retweet we used **Gephi** which is is an open source software for the analysis and visualization of social networks, written in Java and based on the NetBeans platform.



We have encountered some difficulties in using the software, as it is very complicated: in the future it might be interesting to analyze the connections of the graph deeply.

Fake News Analysis

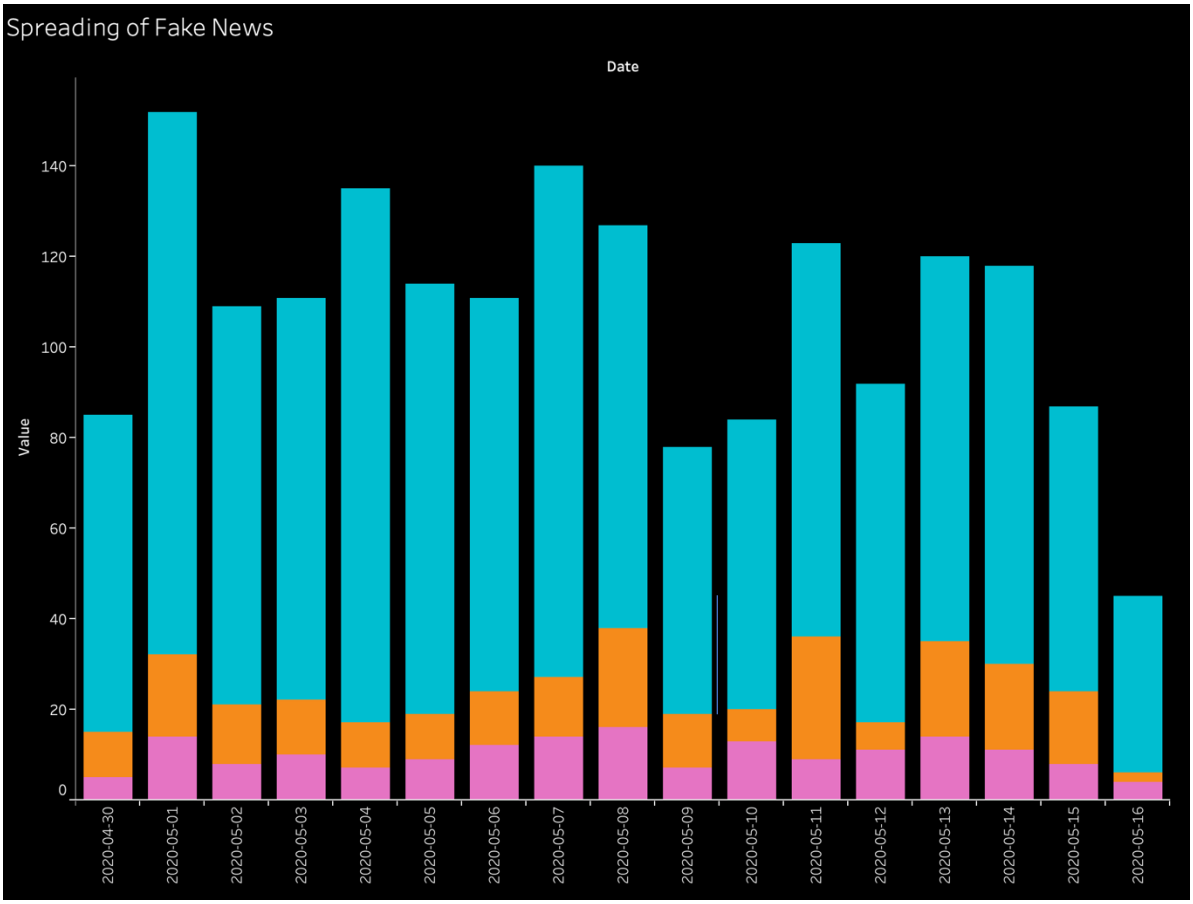
During the pandemic period, the spread of fake news on social networks was particularly pervasive: the spread of fake news was probably one of the major causes of stress and concern. The study of the spread of fake news seemed necessary to be analyzed. The fake news concerning the Covid-19 virus can refer to three main categories: the conspiracy (or the belief that the virus was somehow the result of the failure of diplomacy between states); false homemade treatments and transmission. The analysis was conducted in order to search the tweet texts of the keywords that referred to one or more categories, in particular the keywords that were chosen are the following:

- For what concern the conspiracy:
  "fault", "blame", "plandemic", "laboratories", "lab", "labs", "created", "create"
- For what concern the treatments:
  "bleach", "steroids", "water", "beard", "garlic", "oranges", "lemon", "milk", "chilli"
- For what concern the transmission:
  c("pets", "hair", "clothing", "sole", "shoes", "ticks")

The result obtained is a table with the following columns: created_at (creation date), text, status (location), conspiracy, treatment, transmission. At this point it was decided to group by date and location by adding the occurrences for each category of fake news.

```
1. finalfn <- semifinalfn %>% group_by(created_at, status) %>% summarise(complotto = sum(compl
   otto), cure = sum(cure), trasmissione = sum(trasmissione)).
2. faketotviz <- finalfn %>% gather(key = "Tipo_FakeNews", value = "Occorrenze", complotto:tra
   smissione)
```

For the purpose of visualization, a further modification has been made to the table, i.e. the category name has become a label with the value associated with it alongside, this is possible in a single line of code exploiting the potential of the tidyr package, containing the function gather ().



Normalization as resizing method

We have chosen to perform the normalization of the data in order to display through a choropletic map how normalized tweet number evolves along a territory considering the bias according to which that regions with bigger sizes tend to have a bigger weight in the map interpretation.

In choosing which type of normalization we wanted to implement, we preferred the standardization because it scales the data by leading the mean to 0 and the standard deviation to 1, so as a standard distribution.

One of the reason why we select standardization is because this type of transformation is more versatile in the presence of outliers compared to alternative standardization techniques, such as min-max scaling but also because we did not know the distribution followed by our data.

In our specific case, we thought that the preferable action was to standardize both by location and by date because we did not want to show the variation between one state and another one on the single day but rather the change that occurred in the single state along the entire analysis period.

Another significant focus that we wanted to valorize is to create a clear separation between night and day and for this reason standardization has been applied both for the population of night tweets and for daytime tweets.

The procedure provides to calculate the range between each value (x) and the arithmetic mean (μ) in relation to the standard deviation (σ). The formula is then expressed as: $\frac{x_i - \mu}{\sigma}$

During the development of this project we read about a similar study on Internet, so we decided to use the same data and visualize them.

This work is present in the following link: https://github.com/mykabir/COVID19

As explained, "*his main objective is to explore the psychology and behavior of the societies at large which can assist in managing the economic and social crisis during the ongoing pandemic as well as the after-effects of it. In this paper, we describe the CoronaVis Twitter dataset (focused on United States) that we have been collecting from early March 2020.*[11]*"*

The data has been collected from 5 March 2020 until 24 April 2020 and they have been saved in csv files with the following structure:

- Tweet_id: Unique ID of a tweet
- created_at: Creation time of a tweet.
- loc: State level user location

[11] CoronaVis: A Real-time COVID-19 Tweets Analyzer https://arxiv.org/abs/2004.13932

- text: Processed tweet text. All the text are in small letters, non-English characters and few stop words are removed
- user_id: Pseudo user id
- verified: Denotes whether the tweet post is verified or not (1 or 0). 0 → Not Verified. 1 → Verified

In order to analyze the dataset we carried out the study using Python.

We implemented the data enrichment to the new dataset in order to increase the data information by adding the name of the state and the time zone name.

In an initial exploration of the data, we found the presence of numerous countries not belonging to the US, so they have been excluded. The non-American identified States are: not US territory (XX), Puerto Rico (PR), Guam (GU), Jakarta (ID), Germany (DE) while the American States superfluous for the study are: Northern Mariana Islands (MP), Hawaii (HI), Alaska (AK).

Then the dates in UTC are converted into local dates using the time zone name and after the time, the date and merging between them are extracted from them.

Moreover, starting from the hour, the night we divided from the day according to the same criterion previously reported, that is: 0 means the night period that starts from 00:00 and ends at 5:59, instead 1 means the daytime period that starts at 6:00 and ends at 23:59.

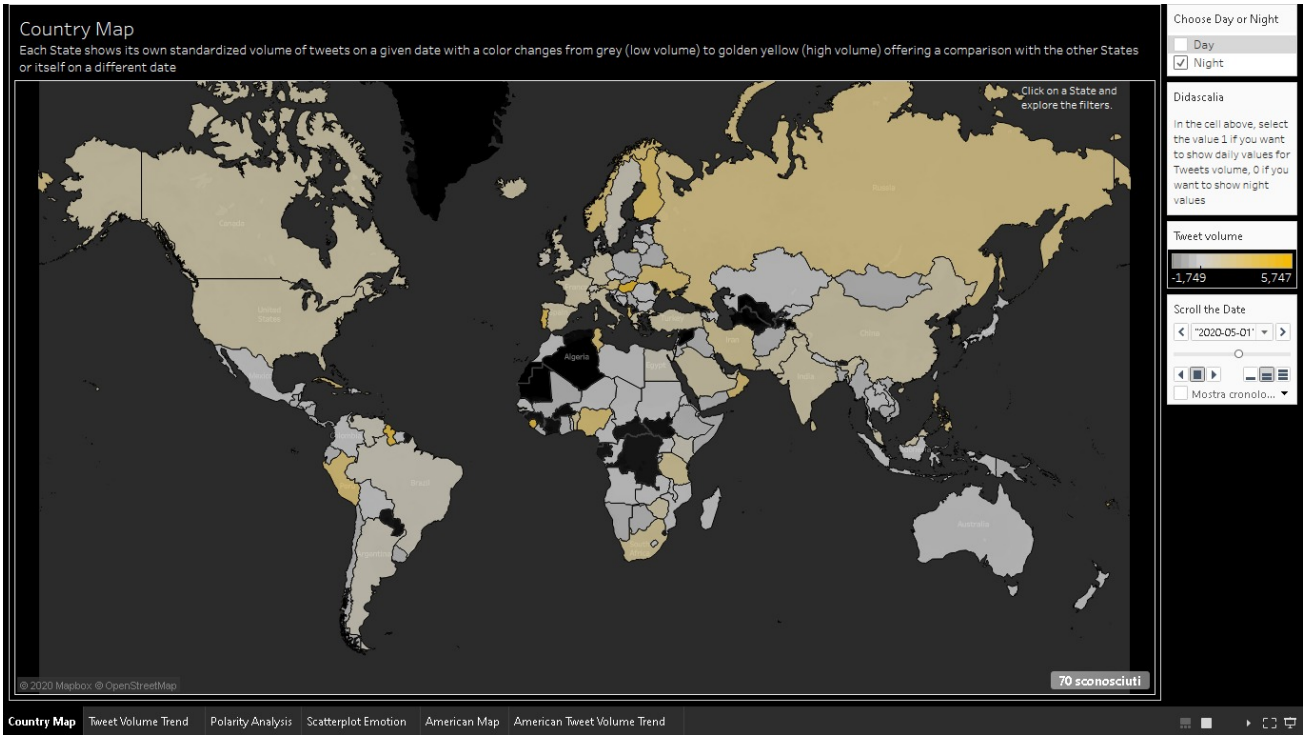
Data Visualization

Once the complete datasets were obtained, it was possible to view the data and verify the hypotheses made through the research questions.

In addition to the real-time interactive dashboards ("Real Time Analysis") The tools used for the visualizations were Tableau and Gephy[12].

[12] Look at page 22

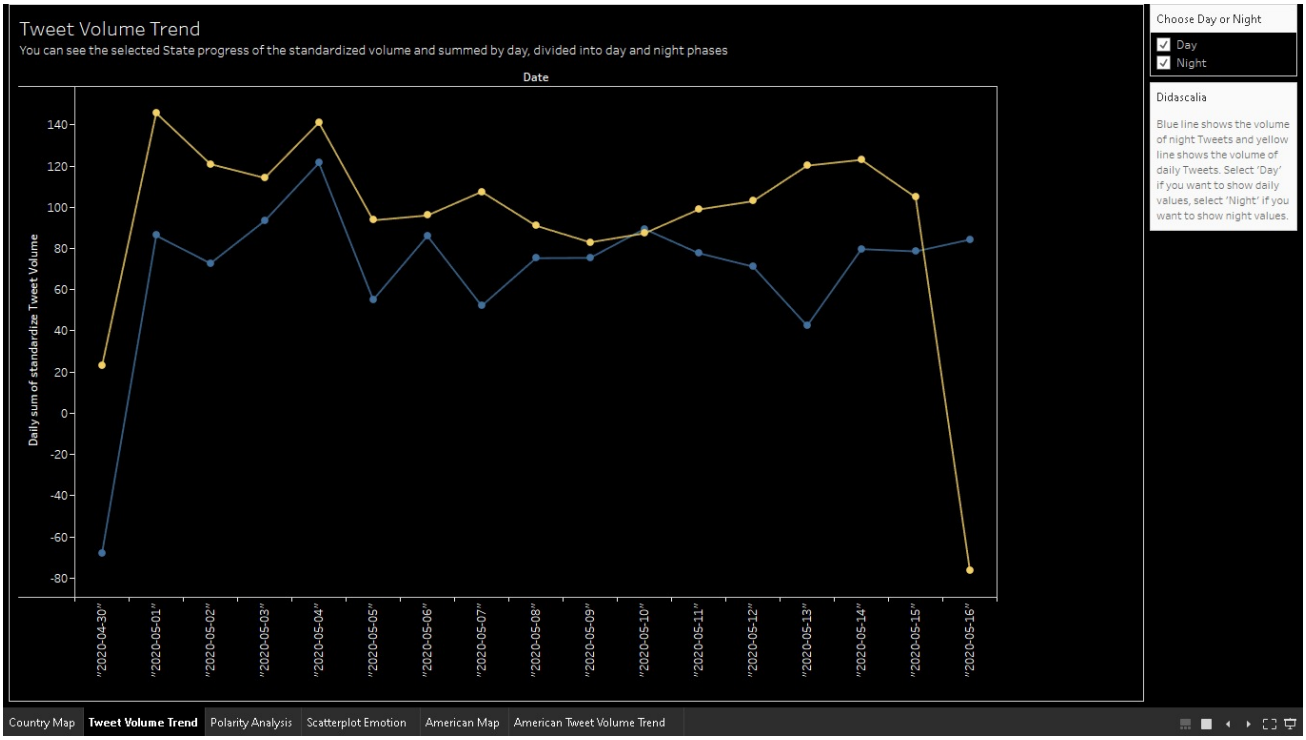
Tableau



Choropleth World Map

After the standardization of the data, was created a choropletic map, that is a thematic map in which the areas are coloured according to the standardized volume of tweets of the specific State selected. Unlike the classic choropletic maps, this one has a time slider with automatic reproduction that allows you to view the chromatic variation indicating an increase or decrease in tweets'volume.

There is also a distinction between night and day which provides a significant amount of additional information.

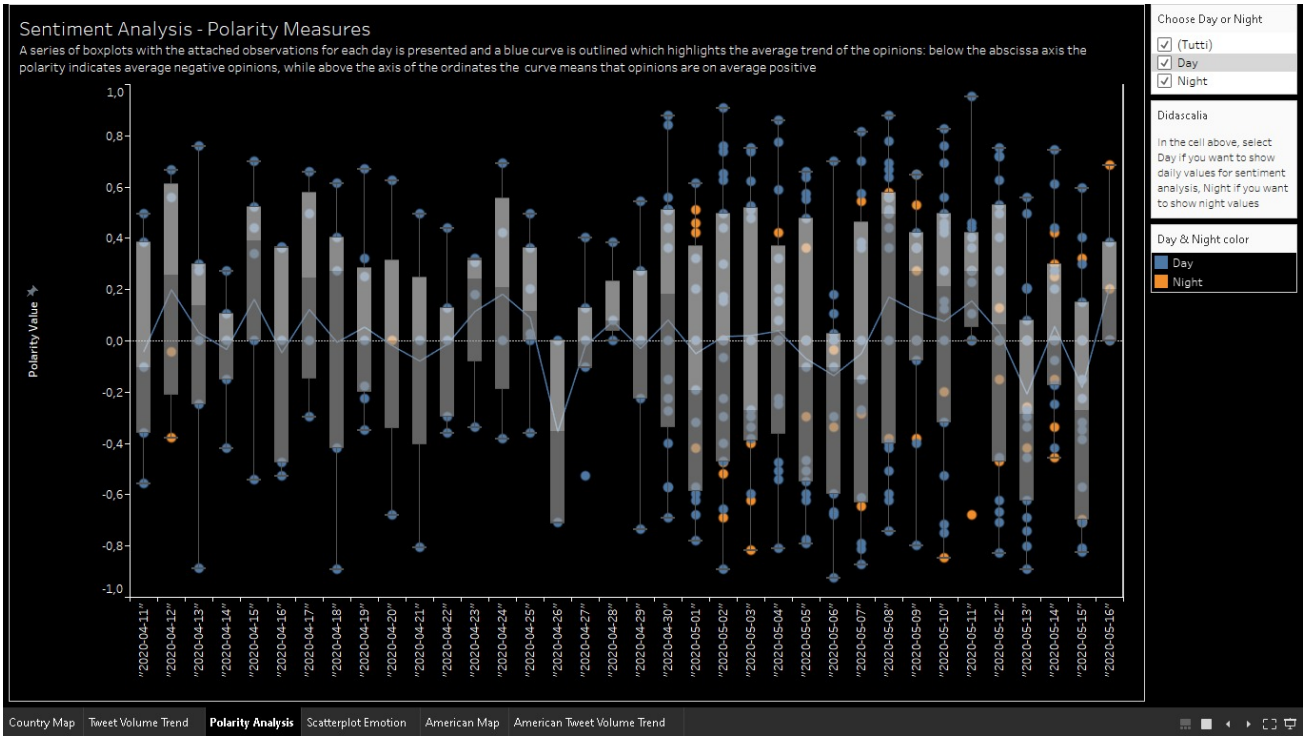


Linechart – Volume Tweet Trend

Since the comparison between very distant dates can be difficult, we decided to produce one single linechart for each state that provides temporal information.

In fact, once a certain State has been chosen and clicked, you will be directed to a page consisting of a linechart that shows the trend of the volume of both night and daytime tweets along the timeframe that starts from April 30 , 2020 to May 16, 2020.

Thanks to it we were able to answer our first question: "*have the night tweets increased with the passing of the lockdown days?"*



Boxplot – Polarity Analysis

In addition to the linechart, another essential infographic were producted, in order to partially answer the second research question: *"what emotions mainly characterize tweets?"*

The observations are the polarity values of each tweet, detected in real time and arranged in a graph composed of a series of boxplots with annexed observations divided by color into night and daytime. Furthermore, to know the daily evolution of polarity, we have outlined and linked the averages of each boxplot obtaining an overall overview of the thinking of each country.

Sentiment Scatterplot as a Quadrant Matrix

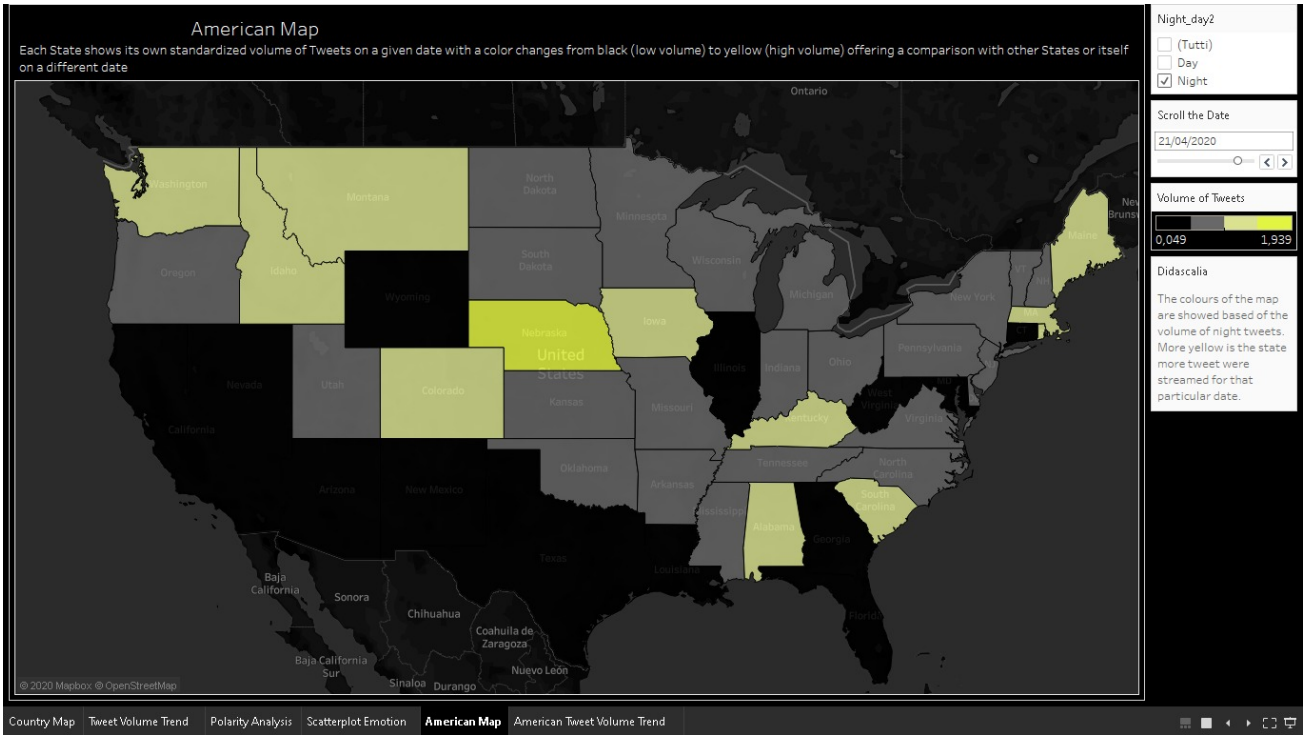
During numerous reflections on which method could be useful to visualize the most widespread emotions in the globe, connected to the Covid-19 pandemic, a scatterplot was used. Although it does not seem the most suitable infographic at first, with a more accurate evaluation it will be possible to clearly perceive some elements that with other visualizations it would not have been possible.

We examined the eight feelings in question by cataloguing the negative ones from the positive ones and the similar ones from the opposite ones; then we weighed their possible combinations to create 4 uniform pairs. Two of them, were assigned to the X axis and were calculated as the sum of trust and surprise subtracted fear and disgust, while the other two couples were assigned to theY axis as the sum of joy and anticipation subtracted from the sadness and anger. The two indexes created, calculated for each state, offer us the opportunity to classify and highlight at a glance which states were mainly characterized by negative emotions in the first quadrant and which by positive emotions in the third quadrant. We also distinguish those particular states that fall into a dualistic area where only negative or only positive emotions are not privileged but are mixed together.

Finally, we have marked each continent with a color, that can be discerned for a specific trend if viewed individually.

The measure scale adopted in both the X and Y axes is the symmetric logarithmic one for obtaining a more evenly distributed data points and less isolated extremes.

Scrolling through the list of continents, there are no singular patterns or trends, but the observations are approximately distributed homogeneously in all the quadrants assuming that the variances are uniform with each other.

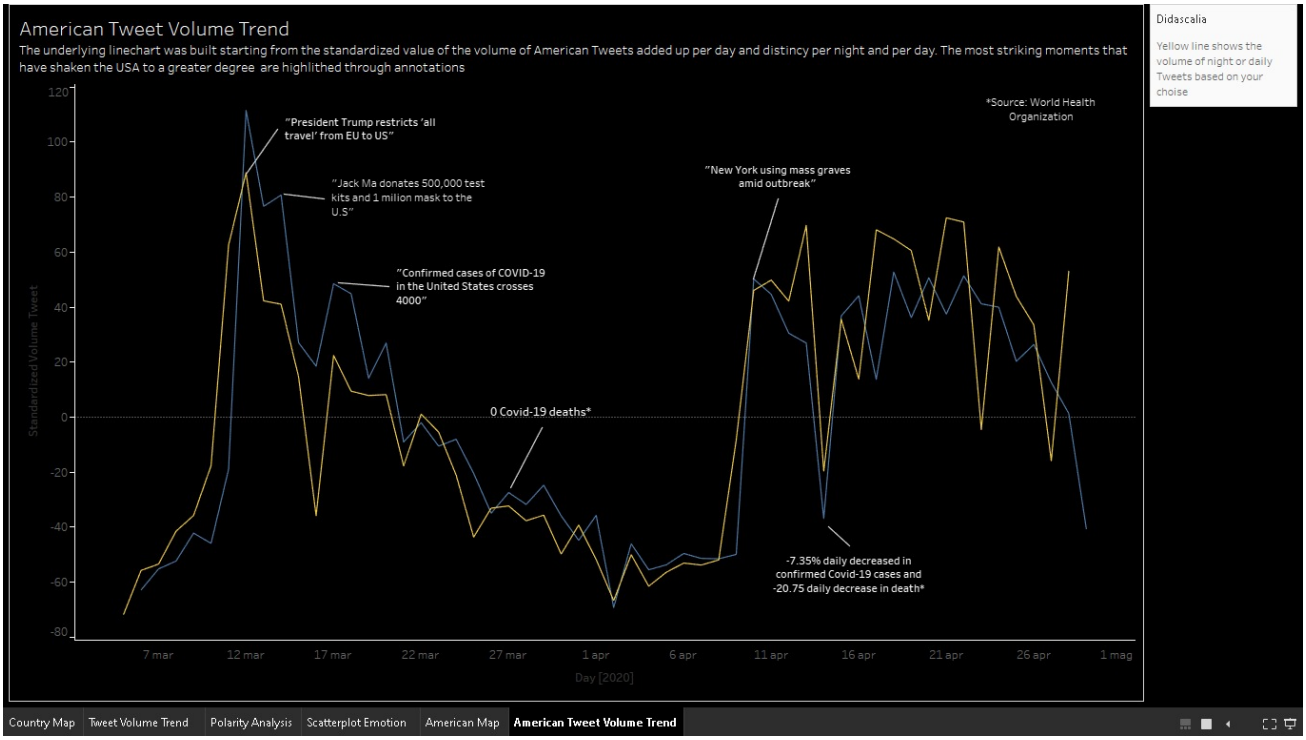


Choropleth USA map

Following the dynamics already developed in the first map, a choropletic map of all the states of the United States of America is presented.

Also this time the color changes from state to state, when the date is scrolled, thus highlighting both the liveliest and most active subjects and the most dull and inactive ones.

When the color becomes brighter, it means that a greater volume of tweets originated on the specified day in the concerned state, than both the other states in the moments in which they were characterized by a darker and deeper color and of itself during the whole period of time.



Linechart – American Tweet Volume Trend

In order to have a visualization that had a greater visual impact of the color change, we created a line-chart that re-configured the American tweet volume trend from March 6, 2020 to April 29, 2020, thinking that a line or otherwise a curve would provide a higher quality information content.

As in any proposed infographic, we give the opportunity to choose between night and / or day, in order to leave the viewer more freedom of decision.

The breakout highlights an initial peak and absolute maximum on March 12, 2020, a fateful day for the USA because the President Trump restricts 'all travel' from EU to US. There are also numerous relative maxima consisting of steep growths followed by deep falls, as if to signify that the individual traumatizing events trigger a common need for belonging, sharing, which arouses curiosity but also a tangle of emotions and feelings, but as an immediate effect and not protracted over time.

Other important events that have occurred are:

14 March, 2020 → Jack Ma donates 500 thousand test kits and one million mask to the U.S.

17 March, 2020 → Confirmed cases of COVID-19 in the United States crosses 4000.

27 March, 2020 → 0 deaths source: World Health Organization.

10 April, 2020 → New York uses mass graves amid outbreak.

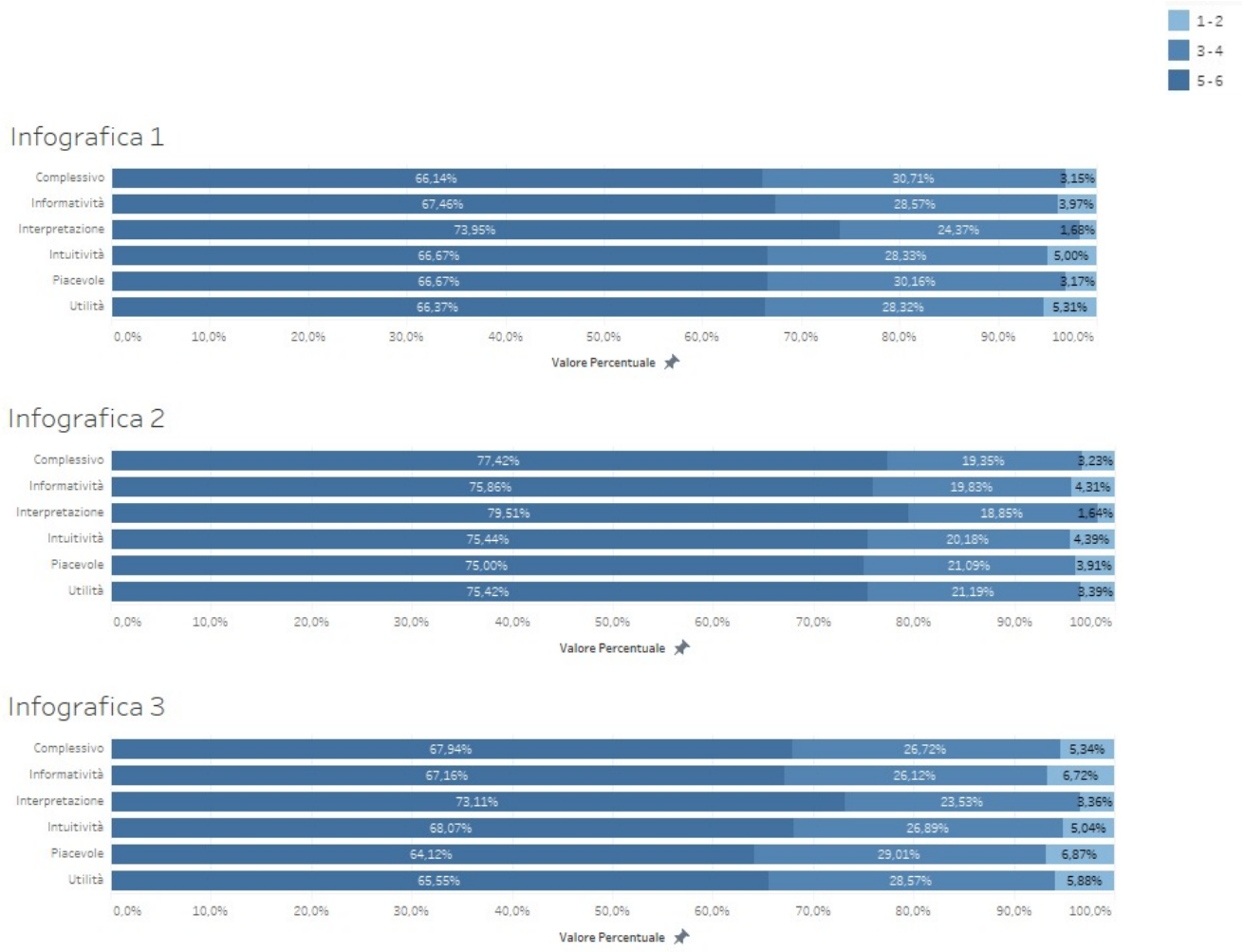
14 April, 2020 → -7.35% daily decrease in confirmed cases and -20.75% daily decreased in death source: World Health Organization.

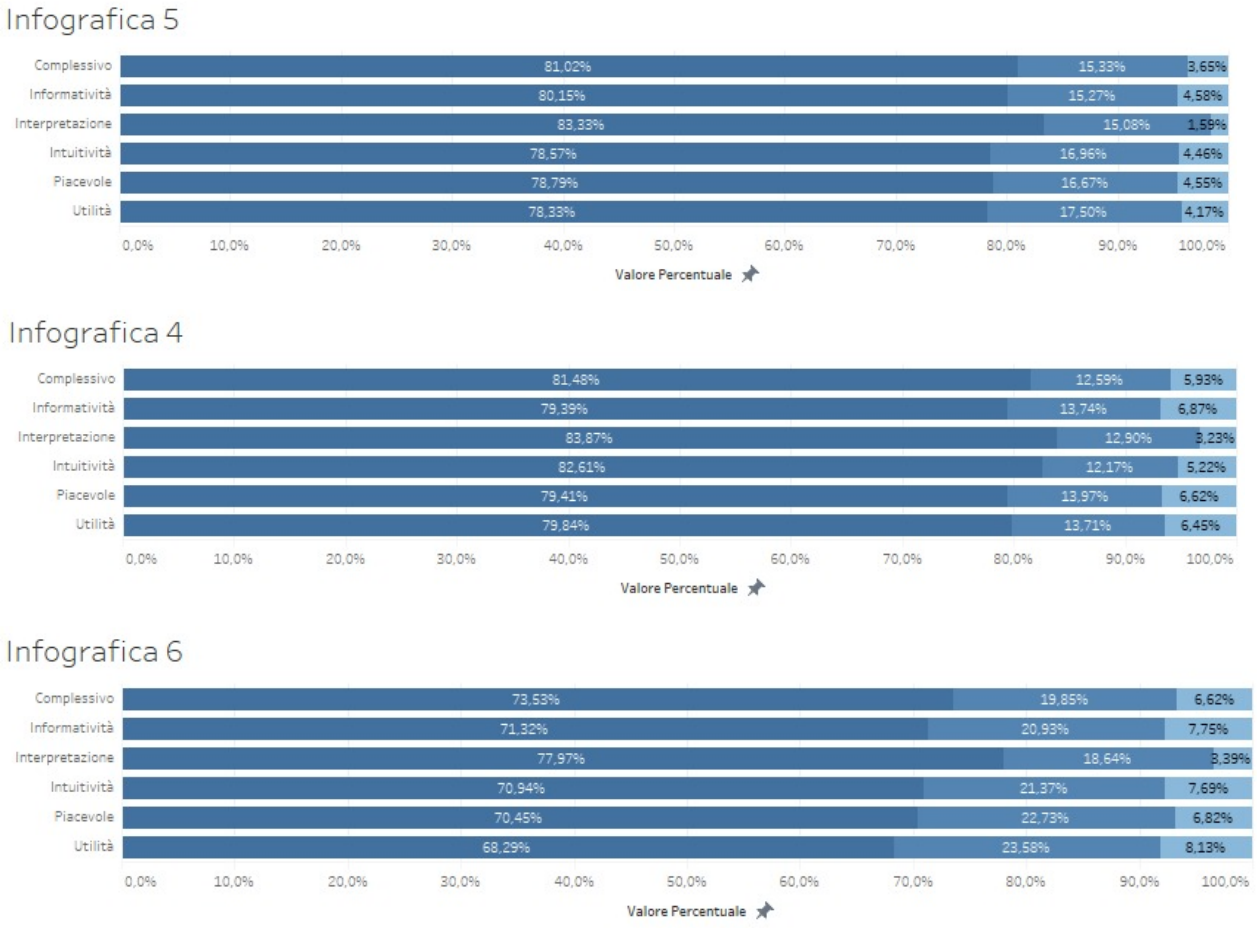
Test of infographic

Psyhometric Test

The test consists of a module for each infographic. The cathegories subjected to evaluation (with a grade from 1 to 6) are: utility, intuitiveness, clarity, informativeness, pleasantness and overall evaluation. The test sample was 26 people.

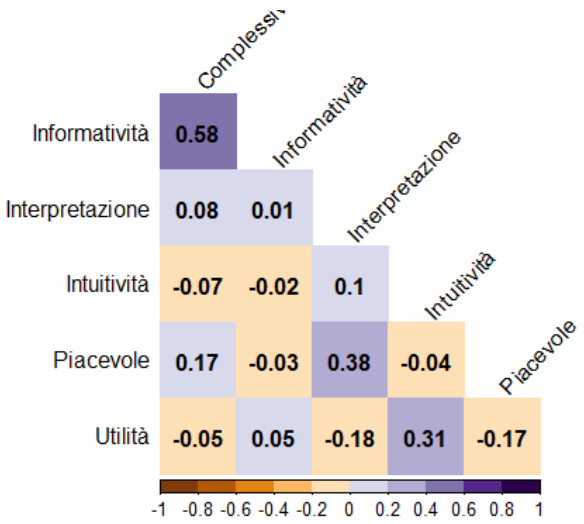
The test results are summarized in the Stacked Bar Charts, where they are shown the answers divided into 3 classes (1-2, 3-4, 5-6)



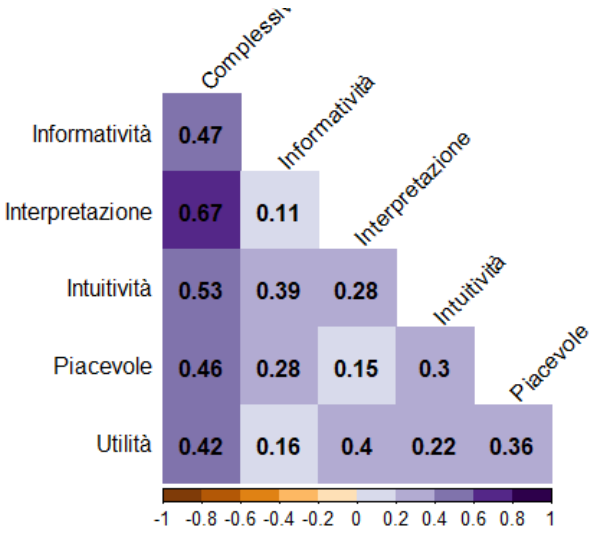


Corrplots

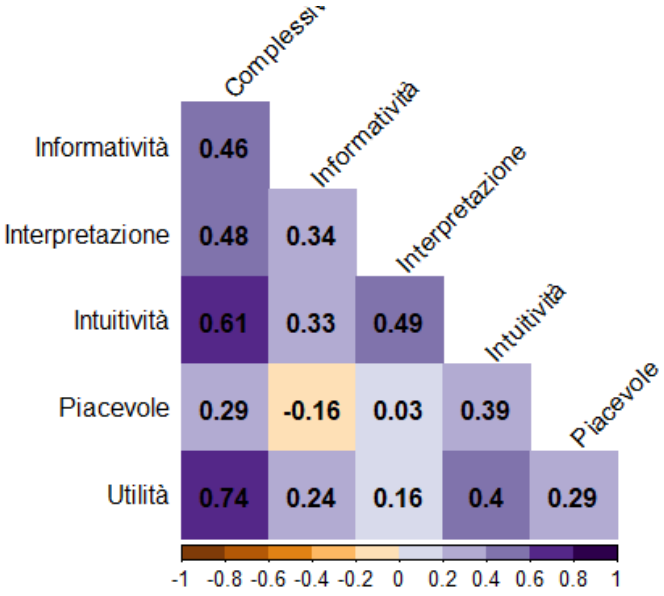
To analyse the relationships between the responses to the surveys, was conducted an analysis on the linear correlations between the various characteristics that were tested. The results are summarized in the following corrplots.



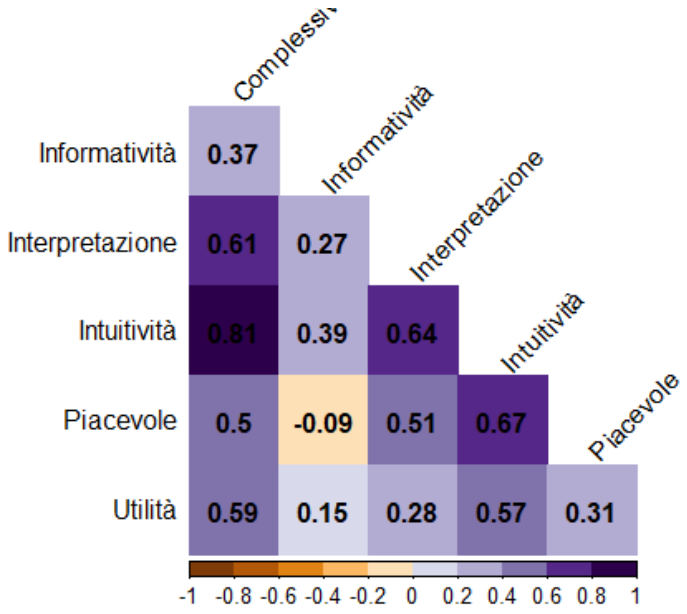
First infographic



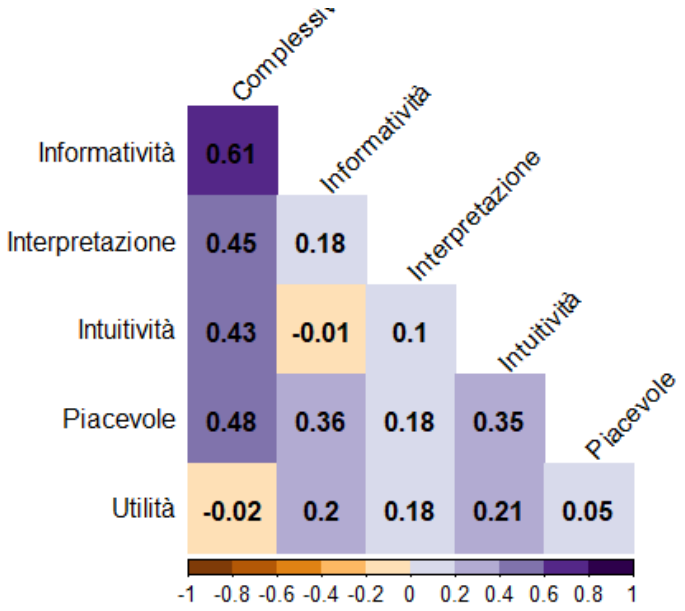
Second infographic



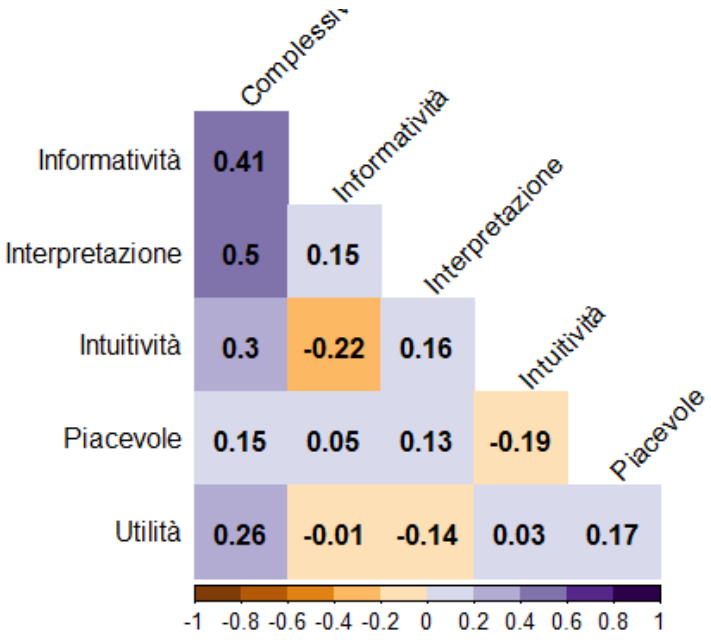
Third infographic



Fourth infographic



Fifth infographic



Sixth infographic

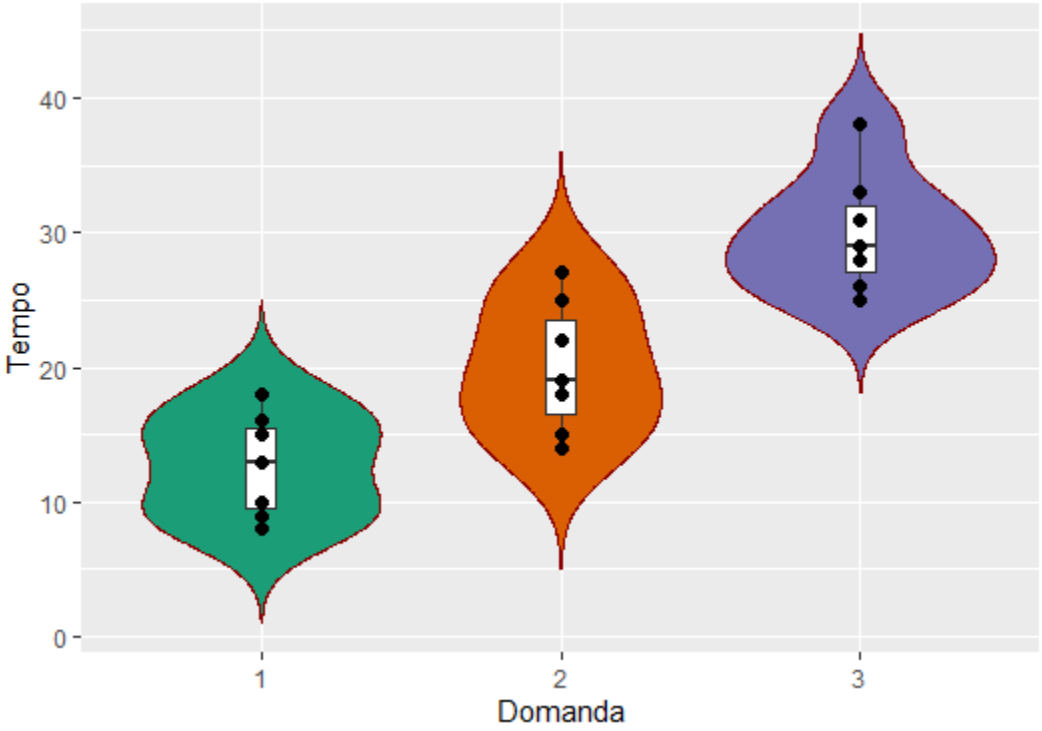
Qualitative test

In this phase 7 people were selected. Each of them answered three questions (regard the most impressive infographic). This phase is useful to understand if the visualizations transmitted what we wanted. The questions were:

- Which country recorded the highest volume of nightly tweets on May 12$^{th?}$ (1/1 point)
- Which is the most confident country? (1/1 point)

- On what day did China record the highest average polarity value?(1/1 point)

Both the response times and the error rate were calculated on the basis of the score of each question. Each person was able to answer correctly 3 questions. The response times was recorded variables between the different questions; the timing distribution is summarized in the following violin plots.



It is possible to notice how the third question has recorded longer response times, since the infographic showing the results of the polarity sentiment analysis is probably the most difficult to understand. This infographic is therefore more suitable for more experienced data analysis users.

Heuristic Test

For the heuristic test, 3 people were interviewed who reported the following considerations:

- It would be necessary to add more context information to make the views easier to understand.
-It is difficult to find a unique trend in the visualization that reports the results of the polarity sentiment analysis.
- It is difficult to understand the meaning of the axes in the visualization that shows complex sentiment analysis.

Conclusion and further analysis

The study was conducted on a scientific basis claiming that during the worst times of the Covid-19 pandemic people suffered from stress and sleep disturbances. Starting from these assumptions, we have tried to analyze if the volume of night tweets had increased in the various countries of the world. After extracting the data and applying some transformations to make them of a better quality, we visualized the results. In particular, we have chosen the United States of America as the country of focus, both because it is one of the states that mostly uses this social network and because it has suffered serious damage due to the pandemic. We managed through a line chart to verify the fact that there were actually spikes in the volume of night tweets in the event of several particular events.
As for the second research question, which referred to the emotions that characterized the night tweets, the results that were recorded are more disparate, more consistent patterns can be visualized but a deeper analysis and algorithms of more detailed sentiment analysis.
Surely this is an analysis that could have future developments: for example it could be interesting to correlate our results with the deaths of Covid-19.

Roles

Vittoria Porta: Data extraction, Data ingestion, Connection to Mongo DB, Preliminary Analysis: Real Time, Sharded Architecture, Report and Viz

Alessandro Vincenzi: Data enrichment and transformation Pre-processing & Missing, Data quality assessment ,Sentiment Analysis, Normalization, Report and Viz
Fabio Pasquale Lombardi: Retweet Analysis, Gephi connection and graph, Fake News Analysis, Report and Viz

Bibliography and References

"Generalized anxiety disorder, depressive symptoms and sleep quality during COVID-19 outbreak in China: a web-based cross-sectional survey" Y. Huang,N. Zhao, June 2020 https://doi.org/10.1016/j.psychres.2020.112954
CoronaVis: A Real-time COVID-19 Tweets Analyzer https://arxiv.org/abs/2004.13932
Twitter Visualization - Display Tweets on a Map in real time. https://www.youtube.com/watch?v=a5MZN9u-8LA
Live Twitter Sentiment Graph - Sentiment Analysis GUI with Dash and Python p.4 https://pythonprogramming.net/live-graph-twitter-sentiment-analysis-gui-dash-python/

Kafka Documentation https://kafka.apache.org/documentation/

MongoDB documentation https://docs.mongodb.com/

WHO Coronavirus Disease (COVID-19) Dashboard https://covid19.who.int/?gclid=EAIaIQobChMIoZe3qLq56gIVA9Z3Ch1WVgPyEAAYASABEgKPMPD_BwE