YouTube al tempo del Covid-19

Un'analisi dei video in tendenza

Gabriele Celeri, Federico Luzzi, Marco Peracchi, Christian Uccheddu

Introduzione e obiettivi

Negli ultimi mesi il Covid-19 ha avuto un impatto notevole sulle vite di tutti noi. L'obiettivo di questo lavoro è capire se la piattaforma YouTube sia stata influenzata dalla presenza di questo virus che ha costretto a rimanere nelle proprie case gran parte della popolazione mondiale. Per valutare l'impatto della pandemia e della quarantena obbligatoria, abbiamo confrontato i video in tendenza di Youtube nel periodo Dicembre-Gennaio rispetto Marzo-Maggio, con l'obiettivo di verificare se i contenuti presenti sulla piattaforma e la loro tipologia siano cambiati. Inoltre, ci siamo preposti, come secondo obiettivo, di verificare se l'andamento dei contagi, e delle notizie in merito, specialmente se negative, influenzassero in alcun modo la fruizione di video riguardanti il Covid-19.

Keyword: Covid-19, YouTube

Scelta degli strumenti

Per rispondere alle nostre domande di ricerca dobbiamo comprendere su quali delle tre "V" della Data Management avremmo dovuto concentrare i nostri sforzi. La conclusione a cui siamo giunti si focalizza sulla *Volume* e *Variety*. Il trattamento di grandi quantità di dati è risultato fondamentale, in quanto tramite le API di Youtube è stato possibile ricavare una discreta quantità di informazioni riguardanti i video in tendenza ($\sim 3Gb$). I dati riguardanti la pandemia sono stati ricavati in formato *csv*, di conseguenza l'integrazione con i dati di Youtube, in formato *json*, è stata fondamentale.

Per la gestione dei dati da Youtube ci siamo affidati al software MongoDB, sul quale sono stati caricati i dati mediante script python. Le API di Youtube non permettevano di scegliere il periodo, ma fornivano i dati in tempo reale, per questo abbiamo utilizzato Apache Kafka per avere i dati salvati per poi poterli caricare su MongoDB.

Raccolta dati

La raccolta dati è basata su due fonti principali: YouTube e [fonte/i covid-19].

Youtube API

I video in tendenza su Youtube variano ogni 15 minuti circa (fonte: `https://support.google.com/youtube/answer/7239739?hl=it`). Tuttavia questo non significa che cambino effettivamente i contenuti presenti, infatti solitamente si assiste solo a qualche video che scompare dalle tendenze o viceversa. Va specificato inoltre che il numero assoluto di video in tendenza è di circa 200, con qualche calo durante la notte più rilevante.

Le API di Youtube ci hanno permesso di raccogliere i dati necessari riguardanti i video in tendenza, con delle limitazioni sul numero di richieste gratuite che si potessero fare mediante il Google Developer.
Sono state affrontate due sessioni di *scraping* con metodi leggermente differenti:

1. dal 23 dicembre 2019 al 5 gennaio 2020 (raccolta ogni 30 minuti)
2. dal 18 marzo 2020 al 6 maggio 2020 (raccolta ogni 6 ore)

Abbiamo deciso di raccogliere dati delle tendenze dei seguenti paesi:
Italia, USA, Regno Unito, India, Germania, Canada, Francia, Corea del sud, Russia, Giappone, Brasile, Messico

Prima sessione

L'idea che ha guidato la fase iniziale del progetto era quella di comprendere come l'algoritmo delle tendenze di Youtube sceglie i video, utilizzando caratteristiche come visualizzazioni, likes, dislikes e commenti. La raccolta dati ha seguito il seguente algoritmo, messo in pratica dallo script *scraper_timed.py*:

Algorithm 1: Scraping youtube → csv

Data: country - insieme dei paesi prescelti; videos - insieme di video scaricati da un determinato paese

```
for every 30 minutes do
    foreach country do
        videos = APIrequest(max(50 video), country)
        while tendency videos are finished do
            videos += APIrequest(max(50 video), country)
        videos_fix = arrangeData(videos)
        saveToCsv(videos_fix)
        videos = ∅
```

Nota: per ogni richiesta API è possibile scaricare un massimo di 50 video, inoltre ogni paese ha un numero di video differente (solitamente 150-200).

I dati vengono salvati in formato *csv* secondo il seguente schema:

Attributo	**Descrizione**
timestamp	data, ora e minuto della nostra rilevazione
video_id	identificativo unico del video
title	nome del video per esteso
publishedAt	data di pubblicazione
channelId	identificativo unico del canale che ha pubblicato il video
channelTitle	nome del canale per esteso
categoryId	identificativo unico della categoria
trending_date	data in cui il video è in tendenza
tags	stringa contenente i tag usati, separati dal carattere "|"
view_count	numero di visualizzazioni
likes	numero di like (mi piace)
dislikes	numero di dislike (non mi piace)
comment_count	numero di commenti sotto il video
thumbnail_link	url all'immagine di copertina del video
comments_disabled	booleano che dichiara se i commenti sono disabilitati
ratings_disabled	booleano che dichiara se i like/dislike sono disabilitati
description	descrizione del video

Tabella 1: schema degli attributi dei dati csv

I dati così raccolti sono salvati nella cartella [INSERIRE NOME CARTELLA]

Seconda sessione

Lo scoppio della pandemia da Covid-19 ci ha permesso di cambiare approccio e domande di ricerca, cercando di valutare come la pandemia abbia influenzato Youtube. Con l'esperienza della presa dati precedente abbiamo cambiato metodo di raccolta, preferendo immagazzinare i dati direttamente, attraverso una pipeline, in un database MongoDB.

Inoltre abbiamo effettuato cambiamenti all'algoritmo di scraping, preferendo effettuare le richieste API ogni sei ore, per un totale di quattro rilevazioni al giorno, invece che ogni 30 minuti come prevedeva precedentemente lo schema.

A scopo didattico abbiamo deciso di implementare una piccola pipeline Kafka in cui dividiamo la fase di raccolta dati (*scraper_producer.py*) e la fase di immagazzinamento (*scraper_consumer.py*).

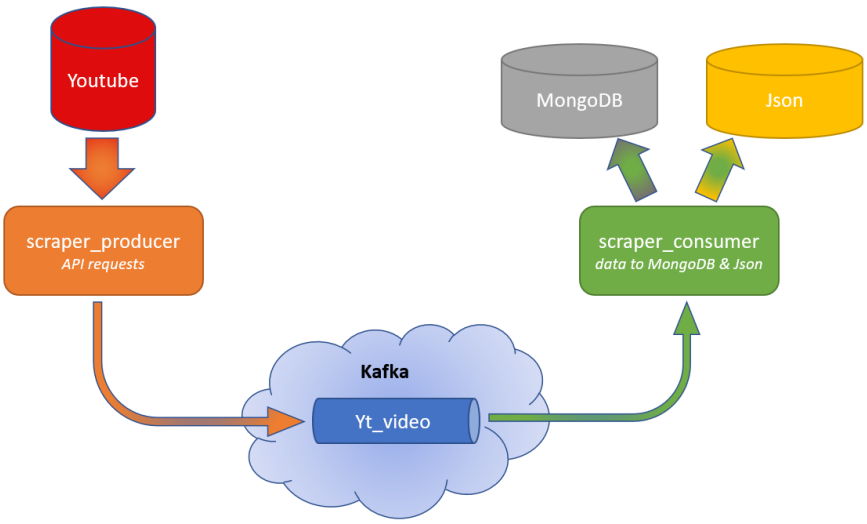

Figura 1: pipeline di raccolta dati - seconda sessione

Di seguito si può vedere nel dettaglio il procedimento dei due algoritmi:

Algorithm 2: scraper producer

Data: country - insieme dei paesi prescelti; videos - insieme di video scaricati da un determinato paese; KafkaProducer(data, channel) - sends data to channel kafka

```
for every 6 hours do
    foreach country do
        videos = APIrequest(max(50 video), country)
        KafkaProducer(videos, yt_video)
        while tendency videos are finished do
            videos = APIrequest(max(50 video), country)
            KafkaProducer(videos, yt_video)
```

Algorithm 3: scraper consumer

Data: KafkaConsumer(channel) - receives data from channel kafka

```
while loop do
    if KafkaConsumer(yt_video) ≠ ∅ then
        videos = KafkaConsumer(yt_video)
        videos_fix = arrangeData(videos)
        saveToJson(videos_fix)
        saveToMongoDB(videos_fix)
```

I dati così raccolti sono stati salvati in formato json nella cartella [INSERIRE NOME CARTELLA].

Nota: abbiamo scelto di salvare i dati in formato Json in modo che fosse facile caricarli in un nuovo server mongoDB e renderli più trasportabili. Per il caricamento vedi lo script: *json_to_mongo.py*

Lo schema logico dei documenti json è sostanzialmente lo stesso di quello dei csv visto precedentemente in *Tabella 1* ma con due variazioni:

- **tags** immagazzinati come array di stringhe

  Ad esempio:

  tags: ["covid-19", "quarantena"]

- informazioni relative ai likes, dislikes, view_count e comment_count innestati in un oggetto **statistics**

  Ad esempio:

  statistics: {view_count: 14235, likes: 513, dislikes: 34, comment_count: 254 }

Dati Covid-19

I dati relativi ai Covid-19 sono stati estrapolati da questo link. I dati vengono estrapolati in formato *csv*; dopo averli puliti essi contengono le seguenti informazioni:

Attributo	Descrizione
iso_code	codice unico riferito al paese
location	nome esteso del paese
date	data della rilevazione
total_cases	numero totale dei casi
new_cases	nuovi casi registrati in quella giornata.

Tabella 2: Schema degli attributi dei dati sul Covid-19.

Qualità dati

Avendo una grande mole di dati ci siamo dovuti anche occupare di verificare la pulizia o meno del nostro dataset. Per prima cosa abbiamo notato che non c'era presenza di Missing Values, questo è stato di per sé una grande fortuna poiché di solito il Missing Replacement è una operazione molto lunga. Ci sono state però due problematiche principali che abbiamo dovuto affrontare:

- Ridondanza: Questo problema è nato dal fatto che i trending di Youtube sono stati presi ogni mezz'ora, con il risultato di avere molti dati simili riguardanti le stesse fasce della giornata.
- Buchi: Questo problema è nato dal fatto che Google non permette di avanzare troppe richieste per la presa dati nella stessa giornata, per arginare questo problema abbiamo utilizzato 3 chiavi diverse con cui mandare le richieste a Google in modo tale da coprire la maggior parte della giornata. Ci sono stati però dei punti in cui la presa dati non è riuscita. Il metodo che abbiamo utilizzato per riempire quei buchi è stato quello di duplicare i dati della presa dati precedente. Questo metodo è stato possibile poiché la presa dati è stat effettuata in intervalli di tempo in cui i dati non vengono modificati significativamente.

Integrazione dati

Per poter rispondere alle nostre domande di ricerca abbiamo dovuto effettuare un'integrazione tra i dati dei video di Youtube e i dati relativi ai contagi giornalieri di covid-19. Il merge l'abbiamo effettuato prima del loro caricamento su MongoDB. Lo script che implementa la nostra integrazione è *merge_to_mongo.py*.

Sostanzialmente applichiamo un'**integrazione temporale** considerando, per ciascun video, il **timestamp** della rilevazione e il paese in cui il video è presente (**country_name**). Ricerchiamo il corrispettivo (timestamp, country_name) nella base dati covid e generiamo un nuovo documento con tutte le informazioni del video di Youtube aggiungendo un sotto documento *covid* con tutti i dati relativi ai contagi avvenuti quel giorno in quello stato. A questo punto i documenti vengono caricati su MongoDB.

Un problema riscontrato durante l'integrazione è stato che la rilevazione viene effettuata con il fuso orario di Greenwich GMT, però ciascun paese ha uno o più fusi orari differenti. Pertanto abbiamo corretto i timestamp in base al fuso orario della capitale di ciascun paese. Nel dettaglio visualizzare il file *country_name.json*.

L'operazione di integrazione complessivamente richiede:

- senza sharding: 1419 s (23,6 min circa)
- con sharding: 1431 s (24 min circa)

Espressione regolare

Per poter distinguere quali video possono essere considerati legati al covid-19 o meno, abbiamo definito la seguente espressione regolare:

```
/(corona|covid|virus|pandemi[aec]|epidemi[aec]|tampon[ei]*|sierologico|mascherin[ae]|코로나 바
이러스|fase\s*(2|due)|iorestoacasa|stayathome|lockdown|[qc]uar[ae]nt[äae]i*n
[ea]|कोरोनावाइरस|वेरेंटाइन|massisolation|distanziamento\s*sociale|social\s*distancing|감염병 세계적 유행
|パンデミック|コロナウイルス|सर्वव्यापी महामारी|मतबदिआपी भतांभारी|пандемия|коронавирус|social\s*distancing|
distanciamiento\s*social|코로나|कोविड|वेंडिड|vaccin[oe]*|isolamento|intensiv[ao]|assembrament[io]|
guant[oi]|dpi|disinfettante|swabs|emergenza|emergency|droplets*|aerosol|isolation|intensive
\s*care|crowd|gloves*|disinfectant|감염병 유행|완충기|마스크|나는 집에있어|폐쇄|사회적 거리두기|백신|모임|비상 사
태|비말|범 혈증|écouvillon|masques*|restealamaison|confin[ae]mento*|distanciation\s*sociale|
soins\s*intensifs|rassemblements|désinfectant|urgence|gouttelettes|飛沫|タンポン|マスケリン|封鎖|
人混みを避ける|ワクチン|隔離|集会|集中治療|緊急|बूंदें|फाहे|मास्क|तॉकडाउन|सोशल डिस्टन्सिंग|टीका|गहन देखभाल|समारोहों|आपातकालीन|
gotas|cotonetes|m[áa]scaras|ficoemcasa|vac[iu]na|reuni[õo]n*es|emerg[êe]ncia|капли|тампоны|
маски|карантин|социальное\s*дистанцирование|вакцина|интенсивная\s*терапия|сходы|
чрезвычайное\s*происшествие|hisopos|mequedoencasa|cierre|Tröpfchen|Tupfer|Masken|
bleibezuHause|Ausgangssperre|soziale\s*Distanzierung|Impfstoff|Intensivstation|
Versammlungen|Notfall|건강\s*격리|検疫|संगरोध|[kK]арантин|hand|man[io]s*|소유|手|[Pp]уки|Hände|
mãos)/i
```

Abbiamo cercato di includere tutte le parole legate al contesto del Covid-19 e la loro traduzione in tutte le lingue dei paesi del nostro studio.

Questa espressione regolare l'abbiamo applicata sia analizzando i titoli dei video, sia analizzando i tags scelti per descrivere il video. L'applicazione è avvenuta tramite le seguenti query:

1. setting tutti video come non covid (sia per titolo che per tags):

   db.video_merge.update({},{$set : {covid_tags : false, covid_title : false}},{multi : true})

2. setting true tutti i video che hanno almeno un tag che rispetta l'espressione regolare (REGEX):

   db.video_merge.update({tags : {$in : [REGEX]}}, {$set : {covid_tags: true}}, {multi : true})

3. setting true tutti i video il cui titolo rispetta l'espressione regolare (REGEX):

   db.video_merge.update({title : {$in : [REGEX]}}, {$set : {covid_title: true}}, {multi : true})

Scalabilità dell'algoritmo

Visto che una delle V che abbiamo deciso di usare concernenti i Big Data è stata quella relativa al volume dobbiamo occuparci di vedere come si comporta la nostra elaborazione dati con volumi di dati sempre crescenti. Abbiamo deciso di implementare lo **Sharding** su MongoDB. In particolare, abbiamo deciso di costruire 3 shard replica group formati da 3 repliche dello shard in modo da garantire la ridondanza nel caso di guasti e la frammentazione per rendere più efficienti le query. La chiave di sharding scelta è quella del campo "**country_name**" perché le query per rispondere alle nostre domande vengono fatte sul singolo paese. Questo approccio è stato pensato anche nell'ottica di dividere la grande mole di dati in server disposti in ciascun paese.

Segue lo schema logico applicato:

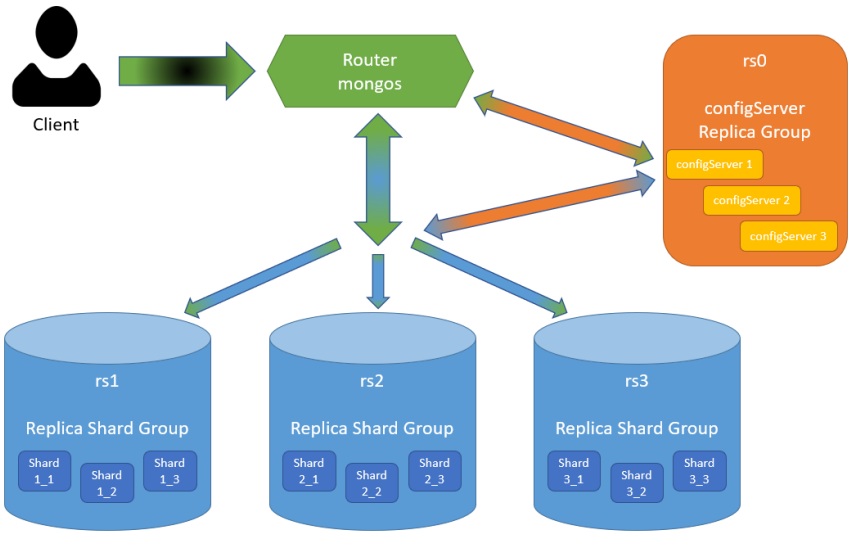

Figura 2: Pipeline sharding

Query

Abbiamo eseguito alcune query per testare il funzionamento del database con e senza sharding. In particolare nella cartella *query* si nota come la query 4, 5, 6 abbiano dei risultati interessanti.

In particolare:

- *query 4*: elenco video a tema covid della categoria intrattenimento d'Italia
- *query 5*: elenco video a tema covid della categoria intrattenimento e musica in Russia
- *query 6*: elenco video a tema covid della categoria news & politics in Italia e Germania

Per quanto riguarda tempo di esecuzione in millisecondi(*executionTimeMillis*):

Query	No Sharding	Sharding
Query 4	635	126
Query 5	802	199
Query 6	1119	538

Tabella 3: differenze tempi in millisecondi

Riguardo al numero di documenti esplorati per ottenere la risposta:

Query	No Sharding	Sharding
Query 4	444207	36950
Query 5	444207	28715
Query 6	444207	444207

Tabella 4: differenze numero di documenti esplorati

Si può notare come in particolare per la query 4 e 5 vengono esplorati un numero di video molto minore nel database con sharding rispetto a quello senza, questo si traduce poi nella marcato miglioramento dei tempi. La query 6 essendo che necessità informazioni su più di un paese esplora tutti i dati non sfruttando la divisione secondo shard keys.

Per il dettaglio vedere i file di risposta nella cartella *query*.

Visualizzazione

Per la scelta delle visualizzazione abbiamo dovuto pensare per prima cosa a delle domande di ricerca a cui rispondere. Le domande che ci siamo posti sono le seguenti:

- Variazione della tipologia di video in tendenza da periodo pre covid-19 a periodo covid-19

- È vero che la fruizione di video su Youtube riguardanti i contagi di Covid-19 segue l'andamento dei dati sull'epidemia?

Scelta features

Per rispondere in modo coerente alle nostre domande di ricerca abbiamo deciso di concentrarsi sulle seguenti features del nostro dataset.

- View Count
- Covid Title
- Covid Tags
- Trending Date
- Title
- Cases New

Scelta della visualizzazione

Per rispondere alle due domande di ricerca che ci siamo posti abbiamo deciso di utilizzare due infografiche diverse. La scelta di utilizzare due infografiche diverse è stata data dal fatto che le due domande sono incompatibili tra loro e quindi sarebbe stato impossibile utilizzarne una unica.

Prima infografica La prima infografica consiste nella combinazione di due diverse visualizzazioni:

- La prima visualizzazione è costituita da un lollipop chart temporale che rappresenta il numero video entrato in tendenza ogni giorno.
- La seconda riguarda è costituita da un bubble chart e riguarda il numero di video per ogni categoria entrato in tendenza quel giorno.

La combinazione di queste due visualizzazioni permette di capire se il contenuto dei video entrati in tendenza nel periodo pre Covid-19 e post Covid-19 differiscano significativamente. Per farlo è sufficiente selezionare due giorni contemporaneamente e vedere il cambiamento nelle bolle. L'esplorazione di questa infografica avviene per step in modo da avvicinare l'utente piano piano a tutte le informazioni che questa infografica può offrire. Di seguito una visione sommaria dell'infografica comprensiva dei contesti:

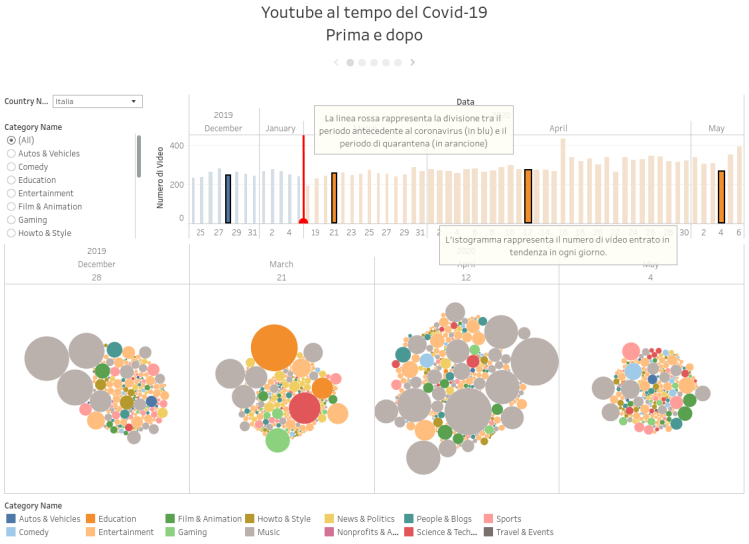

Figura 3: Prima infografica.

Seconda infografica Per quanto riguarda la risposta alla seconda domanda di ricerca abbiamo deciso di utilizzare una infografica composta da due visualizzazioni. In particolare:

- La prima visualizzazione è un misto tra un bar chart e un line chart temporale. In questa visualizzazione abbiamo fatto risaltare la differenza percentuale tra una rilevazione e la precedente. In particolare il line chart riguarda l'aumento percentuale del numero di video in tendenza relativi al Covid-19 mentre il bar chart riguarda l'aumento percentuale dei nuovi casi di Covid-19 rispetto al giorno precedente. Abbiamo utilizzato questo tipo di grafico in modo da poter vedere se la divulgazione negativa dei dati riguardanti il Covid-19 abbia influito sulla fruizione online dei video riguardanti lo stesso argomento. La scelta di creare due forme diverse per i due andamenti è motivata dal fatto che dai questionari è risultata più apprezzata per riconoscere le due variabili.

- La seconda visualizzazione è uno stacked bar chart che rappresenta il numero di video per categoria che riguardano il Covid-19 che può essere filtrata per giorno semplicemente interagendo con la prima visualizzazione.

La combinazione di queste due visualizzazioni consente di rispondere alla seconda domanda di ricerca che ci siamo posti precedentemente. Proponiamo di seguito una visione sommaria dell'infografica comprensiva dei contesti.

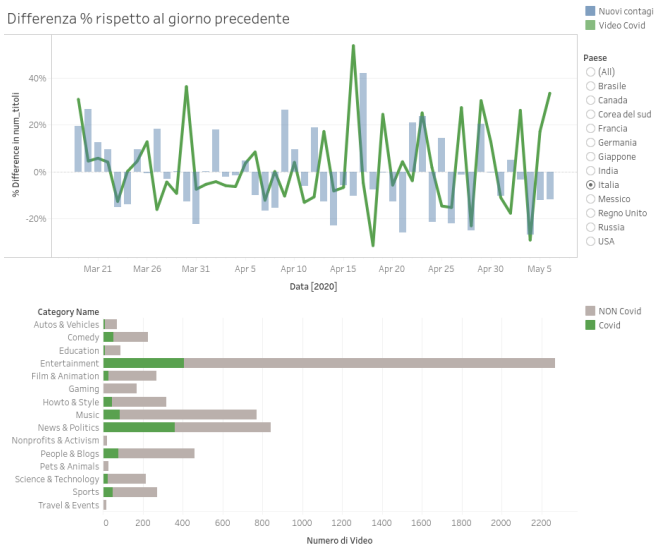

Figura 4: Seconda infografica.

Valutazione della qualità

La valutazione della qualità della nostra infografica si è articolata in tre macro passaggi:

User Test Durante questa fase ci siamo occupati di sottoporre la nostra infografica a 6 persone lasciando completa libertà di esplorazione. Le varie esplorazioni sono state registrate in modo da far sì che potessero emergere le diverse problematiche di cui non ci siamo accorti in fase di realizzazione delle infografiche. Esponiamo le problematiche emerse durante questa fase di valutazione e le correzioni applicate:

- *Problema 1:* Il fatto di avere due variabili sotto forma di linea nella prima infografica rende difficoltoso distinguerle nonostante il colore.
  Soluzione 1: Abbiamo deciso di assegnare ad una variabile la forma "linea" e all'altra variabile la forma "barra".

- *Problema 2:*
  Soluzione 2:

- *Problema 3:*
  Soluzione 3:

Risultati dei task Durante questa fase ci siamo occupati di sottoporre tre diverse richieste per ogni infografica a 24 utenti; questi task devono essere soddisfatti esplorando interattivamente l' infografica. In particolare i task da risolvere sono stati i seguenti:

1. Per la *prima infografica:*
    - Nella giornata del 20 marzo quale video ha avuto più visualizzazioni e a quale categoria appartiene? *Education, Recognizing handwashing ...*
    - Della categoria Entertainment quali sono i canali che hanno fatto più visualizzazioni a Natale e a Pasqua? *The Late Show with Corbin, Mr Beast*
    - Nella categoria education confronta il 29 dicembre e 21 marzo in Germania. Qual è il titolo dei video più visti per ciascun giorno?
2. Per la *seconda infografica:*
    - Trova la categoria che ha avuto più video Covid-19 il giorno 27 Marzo in Italia? *News and Politics*
    - Quanti video Covid-19 della categoria "People and Blogs" ci sono stati negli USA? *2*
    - Quanti nuovi contagi ha avuto la Corea del Sud il 12 Aprile? *62*

Ci siamo occupati di registrare i tempi in cui gli utenti riuscivano a completare questi obiettivi e abbiamo visualizzato questi record nei seguenti violin plot: Questa visualizzazione è utile per capire se le nostre infografiche sono troppo dispersive o riescono a centrare gli obietivi facilmente.

Questionari Per quanto riguarda quest'ultima fase ci siamo occupati di rivolgere un questionario della valutazione della qualità a 24 persone. In particolare il questionario è stato articolato nella seguente maniera:

- Come valuti la chiarezza dell' infografica?
- Come valuti l'utilità dell'infografica?
- Quanto valuti la bellezza dell'infografica?
- Come valuti l'intuitività dell'infografica?
- Quanto è stata informativa l'infografica?
- Come valuti complessivamente l'infografica?

Le risposte sono state registrate grazie al tool: Questionari di Google. Una volta registrate le risposte ci siamo occupati di vedere se la valutazione complessiva dell'infografica fosse coerente con una ricostruzione complessiva data dalla regressione con i coefficienti di Cabitza-Locoro. E' stato comodo usare un box plot per la registrazione di queste risposte poiché la media è un indicatore di tendenza centrale e non fornisce alcuna informazione sulla distribuzione di questi dati. Per quanto riguarda invece della coerenza della valutazione complessiva rispetto alla ricostruzione data dai coefficienti abbiamo avuto la seguente distribuzione: