People's Emotions Analysis while Watching YouTube Videos

Volodymyr Motyka[1], Yaroslav Stepaniak[1], Mariia Nasalska[1] and Victoria Vysotska[1,2]

[1] *Lviv Polytechnic National University, S. Bandera Street, 12, Lviv, 79013, Ukraine*
[2] *Osnabrück University, Friedrich-Janssen-Str. 1, Osnabrück, 49076, Germany*

Abstract
For analysis, a dataset containing information about videos from video hosting YouTube is selected, namely: title, video category, channel (author), number of views, number of likes, number of dislikes, date of video release. The purpose of the study was to analyze the state of people while watching videos on this platform. For this, various methods of visualization and data processing, smoothing methods, correlation and cluster analysis are used.

Keywords
Cluster analysis, correlation, smoothing, YouTube, like, dislike, emotion, sentiment analysis

1. Introduction

Nowadays, sharing information between people in different parts of the world is not a problem if there is access to the Internet. Social networks, messengers, video hosting have become an integral part of our lives. Now almost everything can be done without leaving home. In 2005, one of today's most popular video sharing platforms, YouTube, was created. The idea is simple: the ability to share video/audio with anyone, and most importantly, anyone can share and it's all free, and what's more, people now have the opportunity to earn on YouTube from the content they share using monetization and advertising. It allowed people to relax, because they had an analogue of television. But you can choose what you want to watch, you can watch news, comedies, documentaries and much more, it allowed development, because scientists can spread their knowledge not only in within the walls of the university, but all over the world, it allowed people to spread their thoughts to the masses. Since people are the main users of YouTube, how they feel when they watch the content is extremely important. If a person feels uncomfortable while watching a video (more than one), then he will obviously not want to watch the video sooner or later, and this can cause some commercial problems. In addition, YouTube is one of the sources of operational and current news today. The topic of our research is the emotions that people experience when watching content on YouTube.

2. Related works

Let's pay attention to the exact numbers and look at the statistics of the most popular social networks for July 2021. The data are taken from the resource [1] and shown in Fig. 1. The number of users is given in millions. As can be seen from the statistics, YouTube is the second most popular platform in the world after the social network Facebook. In addition, YouTube is the second most popular search engine after Google. More than two billion of its users, equivalent to nearly one-third of all Internet users, log in every month. However, that is not all. YouTube viewers watch more than a billion hours of video on its platform every day and are responsible for generating billions upon billions of views [2].

COLINS-2023: 7th International Conference on Computational Linguistics and Intelligent Systems, April 20–21, 2023, Kharkiv, Ukraine
EMAIL: volodymyr.motyka@lpnu.ua (V. Motyka), yaroslav.stepaniak@lpnu.ua (Y. Stepaniak), mariia.nasalska@lpnu.ua (M. Nasalska), victoria.a.vysotska@lpnu.ua (V. Vysotska)
ORCID 0009-0009-8086-7619 (V. Motyka), 0009-0007-3074-1132 (Y. Stepaniak), 0009-0008-1089-039X (M. Nasalska), 0000-0001-6417-3689 (V. Vysotska)

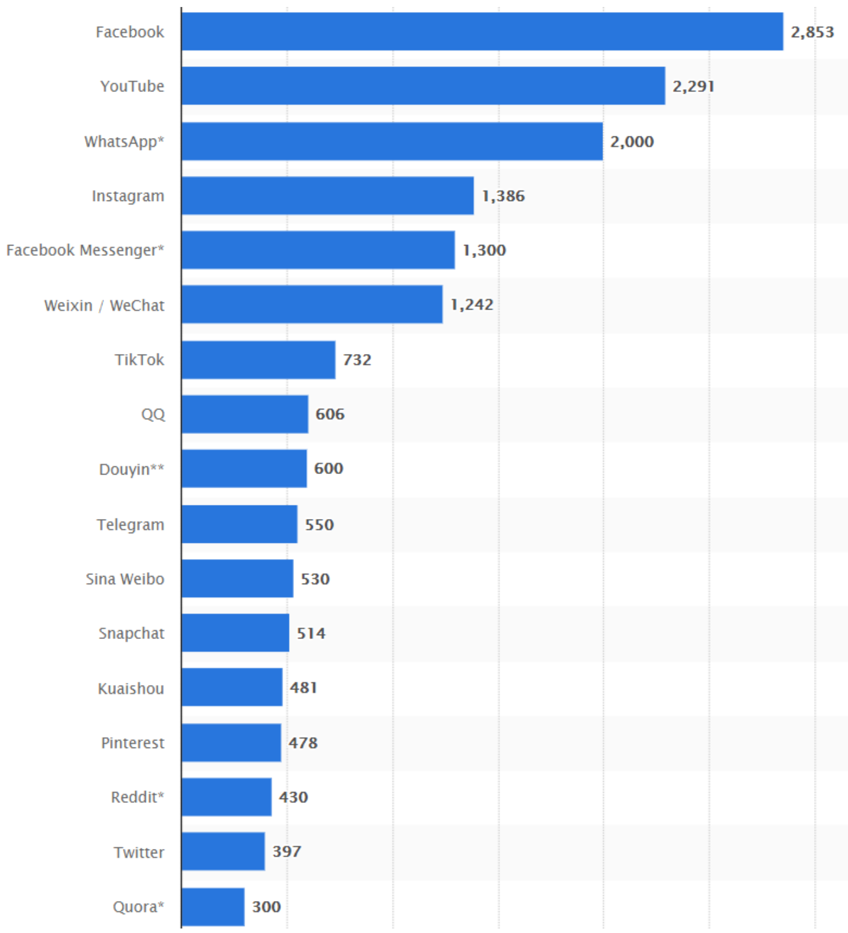
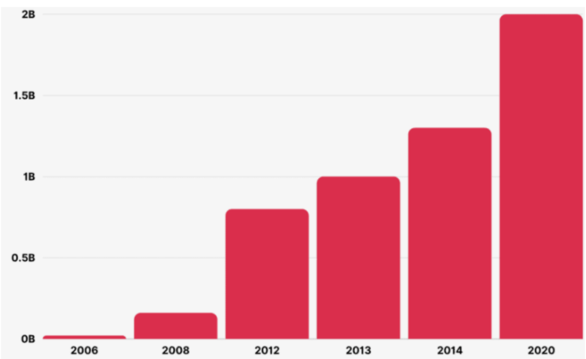


Figure 1: The most popular social networks in the world by active users for July 2021 [1]

Let's look at how many more users have become in recent years:



Country	Views	Population
USA	916 Billion	(~ 2767x total population)
India	503 billion	(~ 363x total population)
UK	391 billion	(~ 5750x total population)
Brazil	274 billion	(~ 1286x total population)
Thailand	207 billion	(~ 3000x total population)
Russia	207 billion	(~ 1428x total population)
South Korea	204 billion	(~ 4000x total population)
Spain	169 billion	(~ 3596x total population)
Japan	159 billion	(~ 1262x total population)
Canada	158 billion	(~ 4158x total population)

Figure 2: The number of YouTube users during its existence (in billions) and the number of video views sorted by country [2-3]

One of the reasons for the jump in popularity was changing the interface and adding new functions and opportunities for users, for example, users could rate not only videos, but also entire playlists, and when choosing a video, they were immediately shown the number of video views and its duration. All this affects the emotional state of users. The reason for the jump in popularity in 2020 was the pandemic of the coronavirus disease, as an extremely large number of people around the world began to work, study at home. This increased the amount of free time people have and they started using social platforms like YouTube more [3-5]. It is impossible not to note the number of video views on YouTube. As can be seen from Figure 2, views more than once exceed the population of countries, so there is certain content that people are ready to view more than once and more than twice.

3. Methods and materials

We will use the methods of visual presentation of data, smoothing, correlation method to perform the tasks. Methods of visual presentation of data - methods of presenting data in the form of graphs, charts and/or other subtypes of them (histograms, pie charts, etc.), time series, etc. Depending on the specific task, a specific method of data presentation will be used. We will implement these methods using Microsoft Power BI and/or R tools. Smoothing methods are used to reduce the influence of the random component (random fluctuations) in time series. They make it possible to obtain more "pure" values, which consist only of deterministic components. Some of the methods are aimed at highlighting some components, for example, a trend [6-8]. We will implement these methods using Microsoft Excel, R and/or Microsoft Power BI. Correlation method (Correlation - analysis) - a method of studying the interdependence of characteristics in the general population, which are random variables with a normal distribution [9-13] for different NLP-talks based on emotions recognizing and analysis [14-23].

4. Experiments

The source of the selected dataset: https://www.kaggle.com/ahmedmohamedmahrous/youtube-textsentiment?select=USvideos.csv. Let's open the dataset using R Studio:

<table>
<tr><th>video_id</th><th>title</th><th>channel_title</th><th>category_id</th><th>tags</th><th>views</th><th>likes</th><th>dislikes</th><th>comment_total</th><th>thumbnail_link</th><th>date</th></tr>
<tr><td>XpVt6Z1Gjjo</td><td>1 YEAR OF VLOGGING -- HOW LOGAN PAUL CHANGED YO...</td><td>Logan Paul Vlogs</td><td>24</td><td>logan paul vlog|logan paul|logan|paul|olym...</td><td>4394029</td><td>320053</td><td>5931</td><td>46245</td><td>https://i.ytimg.com/vi/XpVt6Z1Gjjo/default.jpg</td><td>13/09/2021</td></tr>
<tr><td>K4wEI5zhHB0</td><td>iPhone X â€" Introducing iPhone X â€" Apple</td><td>Apple</td><td>28</td><td>Apple|iPhone 10|iPhone Ten|iPhone|Portrait ...</td><td>7860119</td><td>185853</td><td>26679</td><td>0</td><td>https://i.ytimg.com/vi/K4wEI5zhHB0/default.jpg</td><td>13/09/2021</td></tr>
<tr><td>cLdxuaxaQwc</td><td>My Response</td><td>PewDiePie</td><td>22</td><td>[none]</td><td>5845909</td><td>576597</td><td>39774</td><td>170708</td><td>https://i.ytimg.com/vi/cLdxuaxaQwc/default.jpg</td><td>13/09/2021</td></tr>
<tr><td>WYYvHb03Eog</td><td>Apple iPhone X first look</td><td>The Verge</td><td>28</td><td>apple iphone x hands on|Apple iPhone X|iP...</td><td>2642103</td><td>24975</td><td>4542</td><td>12829</td><td>https://i.ytimg.com/vi/WYYvHb03Eog/default.jpg</td><td>13/09/2021</td></tr>
<tr><td>sjlHnJvXdQs</td><td>iPhone X (parody)</td><td>jacksfilms</td><td>23</td><td>jacksfilms|parody|parodies|iphone|iphone x|...</td><td>1168130</td><td>96666</td><td>568</td><td>6666</td><td>https://i.ytimg.com/vi/sjlHnJvXdQs/default.jpg</td><td>13/09/2021</td></tr>
<tr><td>cMKX2tE5Luk</td><td>The Disaster Artist | Official Trailer HD | A24</td><td>A24</td><td>1</td><td>a24|a24 films|a24 trailers|independent films|...</td><td>1311445</td><td>34507</td><td>544</td><td>3040</td><td>https://i.ytimg.com/vi/cMKX2tE5Luk/default.jpg</td><td>13/09/2021</td></tr>
<tr><td>8wNr-NQImFg</td><td>The Check In: HUD, Ben Carson and Hurricanes</td><td>Late Night with Seth Meyers</td><td>23</td><td>Late night|Seth Meyers|check in|hud|Ben Ca...</td><td>666169</td><td>9985</td><td>297</td><td>1071</td><td>https://i.ytimg.com/vi/8wNr-NQImFg/default.jpg</td><td>13/09/2021</td></tr>
<tr><td>_HTXMhKWqnA</td><td>iPhone X Impressions & Hands On!</td><td>Marques Brownlee</td><td>28</td><td>iPhone X|iphone x|iphone 10|iPhone X impr...</td><td>1728614</td><td>74062</td><td>2180</td><td>15297</td><td>https://i.ytimg.com/vi/_HTXMhKWqnA/default.jpg</td><td>13/09/2021</td></tr>
<tr><td>_ANP3HR1jsM</td><td>ATTACKED BY A POLICE DOG!!</td><td>RomanAtwoodVlogs</td><td>22</td><td>Roman Atwood|Roman|Atwood|roman atw...</td><td>1338533</td><td>69687</td><td>678</td><td>5643</td><td>https://i.ytimg.com/vi/_ANP3HR1jsM/default.jpg</td><td>13/09/2021</td></tr>
<tr><td>zgLtEob6X-Q</td><td>Honest Trailers - The Mummy (2017)</td><td>Screen Junkies</td><td>1</td><td>screenjunkies|screen junkies|screenjunkies n...</td><td>1056891</td><td>29943</td><td>878</td><td>4046</td><td>https://i.ytimg.com/vi/zgLtEob6X-Q/default.jpg</td><td>13/09/2021</td></tr>
<tr><td>Ayb_2qbZHm4</td><td>Honest College Tour</td><td>CollegeHumor</td><td>23</td><td>Collegehumor|CH originals|comedy|sketch ...</td><td>859289</td><td>34485</td><td>726</td><td>1914</td><td>https://i.ytimg.com/vi/Ayb_2qbZHm4/default.jpg</td><td>13/09/2021</td></tr>
<tr><td>CsdzfITXBVQ</td><td>Best Floyd Mayweather Interview | Awkward Puppets</td><td>Awkward Puppets</td><td>23</td><td>best floyd mayweather interview|awkward|p...</td><td>452477</td><td>28050</td><td>405</td><td>2745</td><td>https://i.ytimg.com/vi/CsdzfITXBVQ/default.jpg</td><td>13/09/2021</td></tr>
<tr><td>l864IBj7cgw</td><td>Jennifer Lawrence Challenges Jimmy to an Axe Throwing Co...</td><td>The Tonight Show Starring Jim...</td><td>23</td><td>The Tonight Show|Jimmy Fallon|Jennifer La...</td><td>258781</td><td>8085</td><td>303</td><td>726</td><td>https://i.ytimg.com/vi/l864IBj7cgw/default.jpg</td><td>13/09/2021</td></tr>
<tr><td>4MkC65emkG4</td><td>Hand In Hand A Benefit For Hurricane Relief | MTV</td><td>MTV</td><td>24</td><td>mtv|video|online|official|tv|television|watch|...</td><td>274358</td><td>9215</td><td>477</td><td>838</td><td>https://i.ytimg.com/vi/4MkC65emkG4/default.jpg</td><td>13/09/2021</td></tr>
<tr><td>vu_9muoxT50</td><td>Colin Cloud: Mind Reader Predicts Your Tweets - America's ...</td><td>America's Got Talent</td><td>24</td><td>America's Got Talent 2017|america's got tal...</td><td>473691</td><td>14740</td><td>415</td><td>1696</td><td>https://i.ytimg.com/vi/vu_9muoxT50/default.jpg</td><td>13/09/2021</td></tr>
<tr><td>1L7JFN7tQLs</td><td>iPhone X Hands on - Everything you need to know</td><td>Jonathan Morrison</td><td>28</td><td>Apple|iPhone X|iPhone 8|iPhone X unboxin...</td><td>514972</td><td>18936</td><td>641</td><td>3817</td><td>https://i.ytimg.com/vi/1L7JFN7tQLs/default.jpg</td><td>13/09/2021</td></tr>
<tr><td>ZQK1F0wz6z4</td><td>What Do You Want to Eat?!</td><td>Wong Fu Productions</td><td>24</td><td>panda|what should we eat|buzzfeed|comed...</td><td>282858</td><td>14870</td><td>300</td><td>1398</td><td>https://i.ytimg.com/vi/ZQK1F0wz6z4/default.jpg</td><td>13/09/2021</td></tr>
<tr><td>T_PuZBdT2iM</td><td>getting into a conversation in a language you don't actually ...</td><td>ProZD</td><td>1</td><td>skit|korean|language|conversation|esl|japan...</td><td>1582683</td><td>65749</td><td>1531</td><td>3598</td><td>https://i.ytimg.com/vi/T_PuZBdT2iM/default.jpg</td><td>13/09/2021</td></tr>
<tr><td>w8fAellnPns</td><td>Juicy Chicken Breast - You Suck at Cooking (episode 65)</td><td>You Suck At Cooking</td><td>26</td><td>how to|cooking|recipe|kitchen|chicken|chick...</td><td>479951</td><td>23945</td><td>640</td><td>1941</td><td>https://i.ytimg.com/vi/w8fAellnPns/default.jpg</td><td>13/09/2021</td></tr>
<tr><td>UCrBICYM0yM</td><td>Downsizing (2017) - Official Trailer - Paramount Pictures</td><td>Paramount Pictures</td><td>1</td><td>downsizing|preview|release date|official|dra...</td><td>2693468</td><td>7941</td><td>302</td><td>1432</td><td>https://i.ytimg.com/vi/UCrBICYM0yM/default.jpg</td><td>13/09/2021</td></tr>
</table>

Figure 3: Displaying the data of the selected dataset in RStudio

As can be seen from the dataset (Fig. 4), there are 11 columns with data.

- video_id – video id;
- title – video title;
- channel_title – the name of the channel that posted the video;
- category_id – the identifier of the category to which the video belongs within the YouTube platform;
- tags – "hashtags" used in the video;
- views – number of views;
- likes – number of likes;
- dislikes – number of dislikes;
- comment_total – number of comments;
- thumbnail_link – video link;
- date – the date of the presentation of the video.

We create a report table consisting of category_id and comment_total columns. Let's display data in Cartesian and polar coordinate systems using R tools:

```
#Decart system
ggplot(data = videos, aes(x=category_id, y= comment_total))+ geom_line()
#Polar coordinate system
ggplot(data = videos, aes(x=category_id))+geom_bar(width = 0.5)+coord_polar(theta = "x")
```

Let's define quantitative data: views, likes, dislikes, comment_total. Let's add the category_id column to them for more convenient further analysis. Let's calculate the quantitative characteristics by selecting the views data column, which characterizes the number of views of the corresponding video, using R:

- Sample size – the number of units in the sample: `sample_size<-nrow(videos)`
- Sample mean. We find using the built-in mean() method:

```
avg<-mean(videos$views, na.rm = FALSE)
```

- The median of the sample is the number that "divides" "in half" the ordered set of all the values of the sample, that is, the average value of the changing characteristic, which is contained in the middle of the series, placed in the order of increasing or decreasing of the characteristic. For this, we will use the median() method: `median_views<-median(videos$views, na.rm = FALSE)`
- Mode - the value that occurs most often in the sample. Since there is no built-in method for finding it in R, we will define our modes function:

```
modes <-function(v) {## modes function
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}
mode_views<-modes(videos$views)
```

- Sample size – the difference between the maximum and minimum value of the sample. To find the maximum and minimum, use the built-in methods max() and min():

```
range_views<-max(videos$views)-min(videos$views)
```

- Standard deviation - the amount of spread relative to the arithmetic mean. To find, we will use the built-in method sd(): `standart_deviation<-sd(videos$views)`
- Coefficient of variation – an indicator that determines the percentage ratio of the average deviation to the average value:

```
variation_coef<-sd(videos$views)*100/mean(videos$views, na.rm = FALSE)
```

- Asymmetry reflects the skewness of the distribution relative to the mode. Let's use the built-in skewness() method: `skewness_views<-skewness(videos$views)`
- The kurtosis coefficient characterizes the "steepness", that is, the steepness of the rise of the distribution curve compared to the normal curve. Let's use the kurtosis() method:

```
kurtosis_views<-kurtosis(videos$views)
```

- Standard error is the deviation of the sample from the actual mean. To find it, we will use the formula for calculating the standard error and the sd() method for calculating the standard deviation:

```
standart_error<-sd(videos$views)/sqrt(nrow(videos))
```

To find the number of intervals, we will use Sturges' formula, and to find the width of the interval - Scott's formula. Cumulative – a continuous curve is displayed graphically, which gives a more accurate result compared to a histogram. For construction, we will use the ecdf() function.

Finding the number of intervals and the interval width for the views attribute:

```
k<-1+log2(nrow(videos)) #Number of intervals
h<-3.5*sd(videos$views)*(nrow(videos))^(-1/3) #Interval width
```

Construction of a histogram:

```
hist(videos$views, breaks = k, xlab = "Views", main = "Histogram of views")
```

Construction of cumulata:

```
plot(ecdf(videos$views),xlim=c(0,2*10^7), main="Cumulate", xlab="Views", ylab = "Frequency",
     verticals = FALSE)
```

Finding the number of intervals and the interval width for the likes attribute:

```
k<-1+log2(nrow(videos)) #Number of intervals
h<-3.5*sd(videos$likes)*(nrow(videos))^(-1/3) #Interval width
```

Construction of a histogram:

```
hist(videos$likes, breaks = k, xlab = "likes", main = "Histogram of likes", xlim = c(0,8*10^5))
```

Construction of cumulata:

```
plot(ecdf(videos$likes), xlim=c(0,8*10^5), main="Cumulate", xlab="Likes", ylab = "Frequency",
     verticals = FALSE)
```

Finding the number of intervals and the interval width for the dislikes attribute:

```
k<-1+log2(nrow(videos)) #Number of intervals
h<-3.5*sd(videos$dislikes)*(nrow(videos))^(-1/3) #Interval width
```

Construction of a histogram:

```
hist(videos$likes, breaks = k, xlab = "Dislikes", main = "Histogram of likes", xlim = c(0,8*10^5))
```

Construction of cumulata:

```
plot(ecdf(videos$likes), xlim=c(0,8*10^5), main="Cumulate", xlab="Dislikes", ylab = "Frequency",
     verticals = FALSE)
```

Finding the number of intervals and the interval width for the category_id attribute:

```
k<-1+log2(nrow(videos)) #Number of intervals
```

```
h<-3.5*sd(videos$category_id)*(nrow(videos))^(-1/3) #Interval width
```

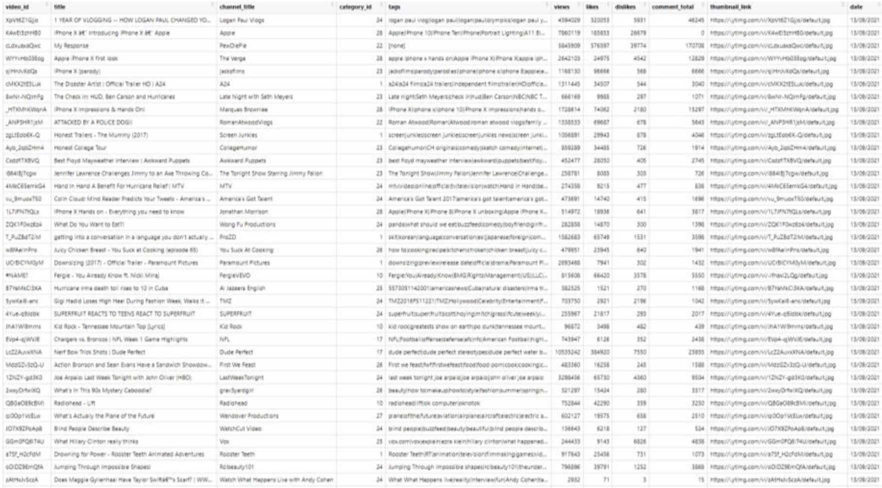
Construction of a histogram:

```
hist(videos$category_id, breaks = k, xlab = "Category", main = "Histogram of categories",
     xlim = c(0,8*10^5))
```

Construction of cumulata:

```
plot(ecdf(videos$category_id), main="Cumulate", xlab="Category", ylab = "Frequency")
```

Let's present the dataset in the form of a table:



video_id	title	channel_title	category_id	tags	views	likes	dislikes	comment_total	thumbnail_link	date

Figure 4: The selected dataset in the form of a table

Number of likes and dislikes depending on video categories:

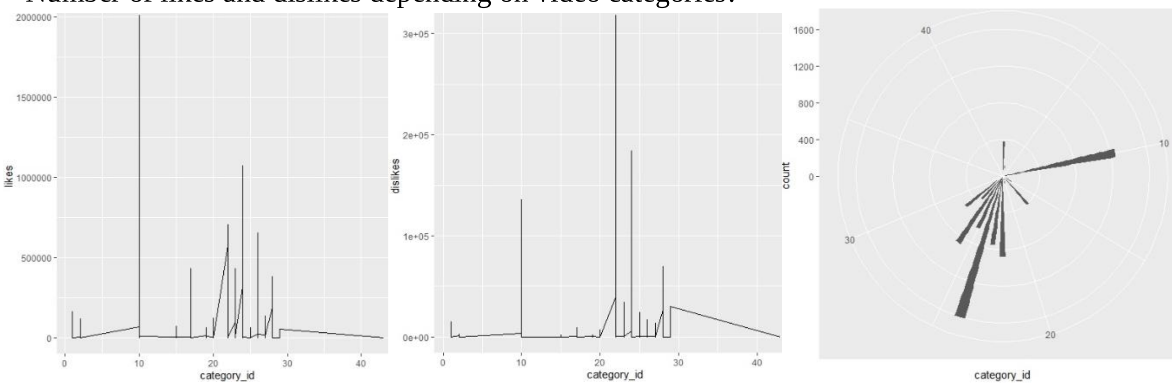


Figure 5: Graph of the number of likes and dislikes in the Cartesian and polar coordinate systems depending on the video category

Let's find the statistical parameters for the views attribute (Table 1).

Table 1
Statistical parameters for the views attribute

Name	Value
Sample size	7998
Selective average	939101.6
Median	308611.5
Mode	0
Sample size	41500672
Standard deviation	2147691
Coefficient of variation	228.10
Asymmetry coefficient	8.056
Kurtosis	99.265
Standard error	24014.92

After executing the code, we have histograms and corresponding cumulates:

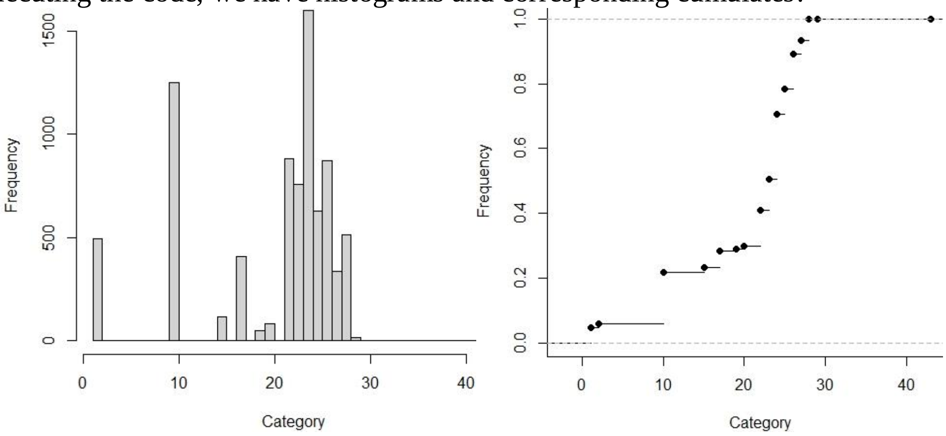


Figure 6: Histogram of data and cumulation of the number of videos of the corresponding category

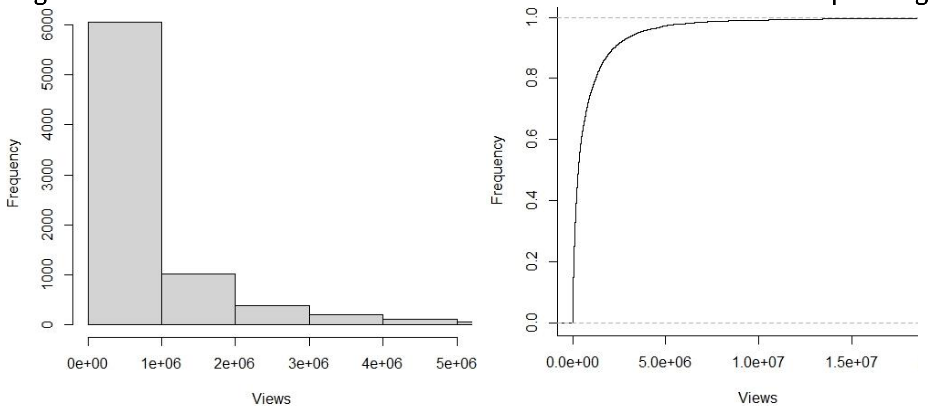


Figure 7: Histogram of data and cumulation of the number of videos of the corresponding category

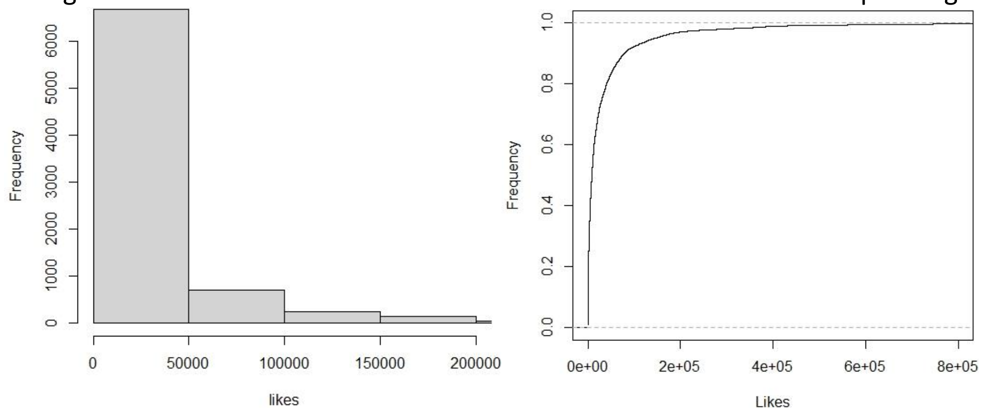


Figure 8: Histogram of data and cumulation of the number of videos of the corresponding category

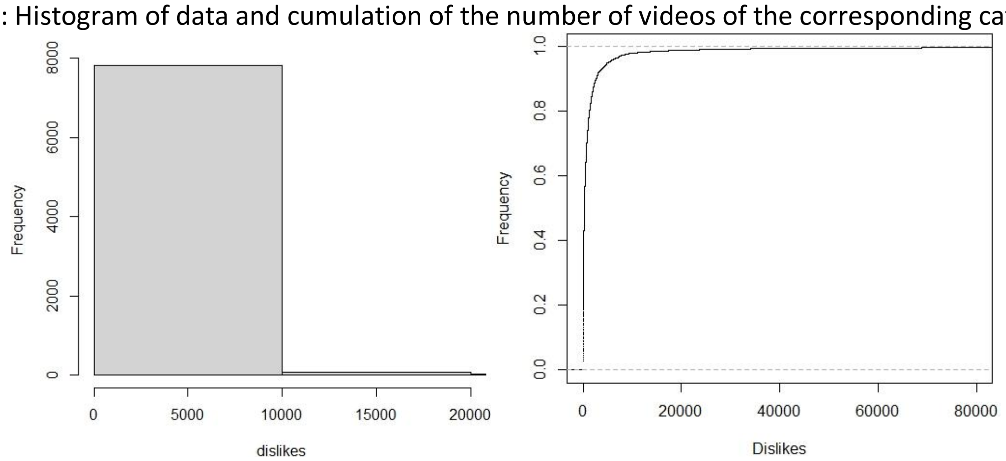


Figure 9: Histogram of data and cumulation of the number of videos of the corresponding category

Smoothing methods are used to reduce the influence of the random component (random fluctuations) in time series. They make it possible to obtain more "pure" values, which consist only of deterministic components. Some of the methods are aimed at highlighting some components, for example, a trend.

Smoothing methods can be conventionally divided into two classes based on different approaches: analytical and algorithmic. The simplest method of forecasting is considered to be an approach that determines the forecast estimate from the actually achieved level using the average level, average growth, average growth rate. Extrapolation based on the average level of the series.

The resulting confidence interval takes into account the uncertainty hidden in the estimate of the average value. However, the assumption remains that the predicted indicator is equal to the sample mean, that is, this approach does not take into account the fact that individual values of the indicator have fluctuated around the average in the past, and this will happen in the future.

Analytical smoothing methods include regression analysis together with the method of least squares and its modifications. To identify the main trend by analytical method means to give the studied process the same development throughout the entire observation period. Therefore, for four of these methods, it is important to choose the optimal function of the deterministic trend (growth curve), which smoothes a number of observations.

Forecasting methods based on regression methods are used for short- and medium-term forecasting. They do not allow for adaptation: with the receipt of new data, the forecast construction procedure must be repeated from the beginning. The optimal length of the lead-up period is determined separately for each economic process, taking into account its statistical instability.

The most widely used are the methods of smoothing time series using moving averages.

For moving average smoothing, we will use Kendel's formulas to calculate the lost levels at the beginning and end of the smoothed series. Let's prepare the data for using smoothing methods:

```
dates<-seq(as.Date("2021-09-13"), as.Date("2021-10-22"), by="days")
str1<-c("13/09/2021","14/09/2021","15/09/2021","16/09/2021","17/09/2021","18/09/2021",
        "19/09/2021","20/09/2021","21/09/2021","22/09/2021","23/09/2021","24/09/2021",
        "25/09/2021","26/09/2021","27/09/2021","28/09/2021","29/09/2021","30/09/2021",
        "01/10/2021","02/10/2021","03/10/2021","04/10/2021","05/10/2021","06/10/2021",
        "07/10/2021","08/10/2021","09/10/2021","10/10/2021","11/10/2021","12/10/2021",
        "13/10/2021","14/10/2021","15/10/2021","16/10/2021","17/10/2021","18/10/2021",
        "19/10/2021","20/10/2021","21/10/2021","22/10/2021")
sums<-1:40
k<-1
sum<-0
for(i in 1:7998)
{
  if(videos$date[i]==str1[k]){
    videos$date[i]<-str1[k]
    sum<-sum+videos$views[i]
  }else if(videos$date[i]==str1[k+1]){
    sums[k]<-sum
    sum<-0
    k<-k+1
  }
}
sums[40]<-sum
views<-sums
k<-1
sum<-0
for(i in 1:7998){
  if(videos$date[i]==str1[k]){
    videos$date[i]<-str1[k]
    sum<-sum+videos$likes[i]
  }else if(videos$date[i]==str1[k+1]){
    sums[k]<-sum
    sum<-0
    k<-k+1
  }
}
sums[40]<-sum
likes<-sums
k<-1
sum<-0
for(i in 1:7998){
  if(videos$date[i]==str1[k]){
    videos$date[i]<-str1[k]
    sum<-sum+videos$dislikes[i]
  }else if(videos$date[i]==str1[k+1]){
    sums[k]<-sum
    sum<-0
    k<-k+1
  }}
sums[40]<-sum
dislikes<-sums
viewh<-data.frame(dates,views,likes,dislikes)
```

Let's visualize:

```
viewh %>%
  gather(metric, views, likes:dislikes) %>%
  ggplot(aes(dates,views, color = metric)) +
  geom_line(size=1)
```

The method of smoothing according to Kendel's formulas:

```
ma <- viewh %>%
  select(dates,sums) %>%
  mutate(ma1 = rollmean(sums, k = 3, fill = NA), ma2 = rollmean(sums, k = 5, fill = NA), ma3 = rollmean(sums, k = 7, fill = NA),
         ma4 = rollmean(sums, k = 9, fill = NA), ma5 = rollmean(sums, k = 11, fill = NA),ma6 = rollmean(sums, k = 13, fill = NA),
         ma7 = rollmean(sums, k = 15, fill = NA))
ma <- viewh %>%
  select(dates,views) %>%
  mutate(ma1 = rollmean(views, k = 3, fill = NA), ma2 = rollmean(views, k = 5, fill = NA), ma3 = rollmean(views, k = 7, fill = NA),
         ma4 = rollmean(views, k = 9, fill = NA), ma5 = rollmean(views, k = 11, fill = NA),  ma6 = rollmean(views, k = 13, fill = NA),
         ma7 = rollmean(views, k = 15, fill = NA))
sums<-views
ma$ma1[1]<-(5*sums[1]+2*sums[2]-sums[3])/6
ma$ma1[40]<-(-sums[38]+2*sums[39]+5*sums[40])/6
ma$ma2[1]<-(3*sums[1]+2*sums[2]+sums[3]-sums[5])/5
ma$ma2[2]<-(4*sums[1]+3*sums[2]+2*sums[3]+sums[4])/10
ma$ma2[39]<-(4*sums[40]+3*sums[39]+2*sums[38]+sums[37])/10
ma$ma2[40]<-(-sums[36]+sums[38]+2*sums[39]+3*sums[40])/5
ma$ma3[1]<-(13*sums[1]+10*sums[2]+7*sums[3]+4*sums[4]+1*sums[5]-2*sums[6]-5*sums[7])/28
ma$ma3[2]<-(5*sums[1]+4*sums[2]+3*sums[3]+2*sums[4]+1*sums[5]+0*sums[6]-1*sums[7])/14
ma$ma3[3]<-(7*sums[1]+6*sums[2]+5*sums[3]+4*sums[4]+3*sums[5]+2*sums[6]+sums[7])/28
ma$ma3[38]<-(7*sums[34]+6*sums[35]+5*sums[36]+4*sums[37]+3*sums[38]+2*sums[39]+sums[40])/28
ma$ma3[39]<-(5*sums[40]+4*sums[39]+3*sums[38]+2*sums[37]+1*sums[36]+0*sums[35]+1*sums[34])/14
ma$ma3[40]<-(13*sums[40]+10*sums[39]+7*sums[38]+4*sums[37]+sums[36]-2*sums[35]-5*sums[34])/28
ma$ma4[1]<-(17*sums[1]+14*sums[2]+11*sums[3]+8*sums[4]+5*sums[5]+2*sums[6]-1*sums[7]-4*sums[8]-7*sums[9])/45
ma$ma4[2]<-(56*sums[1]+47*sums[2]+38*sums[3]+29*sums[4]+20*sums[5]+11*sums[6]+2*sums[7]-7*sums[8]-16*sums[9])/180
ma$ma4[3]<-(22*sums[1]+19*sums[2]+16*sums[3]+13*sums[4]+10*sums[5]+7*sums[6]+4*sums[7]+sums[8]-2*sums[9])/90
ma$ma4[4]<-(32*sums[1]+29*sums[2]+26*sums[3]+23*sums[4]+20*sums[5]+17*sums[6]+14*sums[7]+11*sums[8]+8*sums[9])/180
ma$ma4[37]<-(32*sums[40]+29*sums[39]+26*sums[38]+23*sums[37]+20*sums[36]+17*sums[35]+14*sums[34]+11*sums[33]+8*sums[32])/180
ma$ma4[38]<-(22*sums[40]+19*sums[39]+16*sums[38]+13*sums[37]+10*sums[36]+7*sums[35]+4*sums[34]+sums[33]-2*sums[32])/90
ma$ma4[39]<-(56*sums[40]+47*sums[39]+38*sums[38]+29*sums[37]+20*sums[36]+11*sums[35]+2*sums[34]-7*sums[33]-16*sums[32])/180
ma$ma4[40]<-(17*sums[40]+14*sums[39]+11*sums[38]+8*sums[37]+5*sums[36]+2*sums[35]-1*sums[34]-4*sums[33]-7*sums[32])/45
ma$ma5[1]<-(7*sums[1]+6*sums[2]+5*sums[3]+4*sums[4]+3*sums[5]+2*sums[6]+1*sums[7]+0*sums[8]-1*sums[9]-2*sums[10]-3*sums[11])/22
ma$ma5[2]<-(15*sums[1]+13*sums[2]+11*sums[3]+9*sums[4]+7*sums[5]+5*sums[6]+3*sums[7]+1*sums[8]-1*sums[9]-3*sums[10]-5*sums[11])/55
ma$ma5[3]<-(25*sums[1]+22*sums[2]+19*sums[3]+16*sums[4]+13*sums[5]+10*sums[6]+7*sums[7]+4*sums[8]+sums[9]-2*sums[10]-5*sums[11])/110
ma$ma5[4]<-(10*sums[1]+9*sums[2]+8*sums[3]+7*sums[4]+6*sums[5]+5*sums[6]+4*sums[7]+3*sums[8]+2*sums[9]+1*sums[10]+0*sums[11])/55
ma$ma5[5]<-(15*sums[1]+14*sums[2]+13*sums[3]+12*sums[4]+11*sums[5]+10*sums[6]+9*sums[7]+8*sums[8]+7*sums[9]+6*sums[10]+5*sums[11])/110
ma$ma5[36]<-(15*sums[40]+14*sums[39]+13*sums[38]+12*sums[37]+11*sums[37]+10*sums[36]+9*sums[35]+8*sums[34]+7*sums[33]+6*sums[32]+5*sums[31])/110
ma$ma5[37]<-(10*sums[40]+9*sums[39]+8*sums[38]+7*sums[37]+6*sums[37]+5*sums[36]+4*sums[35]+3*sums[34]+2*sums[33]+1*sums[32]+0*sums[31])/55
ma$ma5[38]<-(25*sums[40]+22*sums[39]+19*sums[38]+16*sums[37]+13*sums[37]+10*sums[36]+7*sums[35]+4*sums[34]+sums[33]-2*sums[32]-5*sums[31])/110
ma$ma5[39]<-(15*sums[40]+13*sums[39]+11*sums[38]+9*sums[37]+7*sums[37]+5*sums[36]+3*sums[35]+1*sums[34]-1*sums[33]-3*sums[32]-5*sums[31])/55
ma$ma5[40]<-(7*sums[40]+6*sums[39]+5*sums[38]+4*sums[37]+3*sums[37]+2*sums[36]+1*sums[35]+0*sums[34]-1*sums[33]-2*sums[32]-3*sums[31])/22
ma$ma6[1]<-(25*sums[1]+22*sums[2]+19*sums[3]+16*sums[4]+13*sums[5]+10*sums[6]+7*sums[7]+4*sums[8]+1*sums[9]-2*sums[10]-5*sums[11]-8*sums[12]-11*sums[13])/91
ma$ma6[2]<-(44*sums[1]+39*sums[2]+34*sums[3]+29*sums[4]+24*sums[5]+19*sums[6]+14*sums[7]+9*sums[8]+4*sums[9]-1*sums[10]-6*sums[11]-11*sums[12]-15*sums[13])/182
ma$ma6[3]<-(19*sums[1]+17*sums[2]+15*sums[3]+13*sums[4]+11*sums[5]+9*sums[6]+7*sums[7]+5*sums[8]+3*sums[9]+1*sums[10]-1*sums[11]-3*sums[12]-5*sums[13])/91
ma$ma6[4]<-(32*sums[1]+29*sums[2]+26*sums[3]+23*sums[4]+20*sums[5]+17*sums[6]+14*sums[7]+11*sums[8]+8*sums[9]+5*sums[10]+2*sums[11]-1*sums[12]-4*sums[13])/182
ma$ma6[5]<-(13*sums[1]+12*sums[2]+11*sums[3]+10*sums[4]+9*sums[5]+8*sums[6]+7*sums[7]+6*sums[8]+5*sums[9]+4*sums[10]+3*sums[11]+2*sums[12]+1*sums[13])/91
ma$ma6[6]<-(20*sums[1]+19*sums[2]+18*sums[3]+17*sums[4]+16*sums[5]+15*sums[6]+14*sums[7]+13*sums[8]+12*sums[9]+11*sums[10]+10*sums[11]+9*sums[12]+8*sums[13])/182
ma$ma6[40]<-(25*sums[40]+22*sums[39]+19*sums[38]+16*sums[37]+13*sums[36]+10*sums[35]+7*sums[34]+4*sums[33]+1*sums[32]-2*sums[31]-5*sums[30]-8*sums[29]-
11*sums[28])/91
ma$ma6[39]<-(44*sums[40]+39*sums[39]+34*sums[38]+29*sums[37]+24*sums[36]+19*sums[35]+14*sums[34]+9*sums[33]+4*sums[32]-1*sums[31]-6*sums[30]-11*sums[29]-
15*sums[28])/182
ma$ma6[38]<-(19*sums[40]+17*sums[39]+15*sums[38]+13*sums[37]+11*sums[36]+9*sums[35]+7*sums[34]+5*sums[33]+3*sums[32]+1*sums[31]-1*sums[30]-3*sums[29]-5*sums[28])/91
ma$ma6[37]<-(32*sums[40]+29*sums[39]+26*sums[38]+23*sums[37]+20*sums[36]+17*sums[35]+14*sums[34]+11*sums[33]+8*sums[32]+5*sums[31]+2*sums[30]-1*sums[29]-
4*sums[28])/182
ma$ma6[36]<-(13*sums[40]+12*sums[39]+11*sums[38]+10*sums[37]+9*sums[36]+8*sums[35]+7*sums[34]+6*sums[33]+5*sums[32]+4*sums[31]+3*sums[30]+2*sums[29]+1*sums[28])/91
ma$ma6[35]<-
(20*sums[40]+19*sums[39]+18*sums[38]+17*sums[37]+16*sums[36]+15*sums[35]+14*sums[34]+13*sums[33]+12*sums[32]+11*sums[31]+10*sums[30]+9*sums[29]+8*sums[28])/182
ma$ma7[1]<-(29*sums[1]+26*sums[2]+23*sums[3]+20*sums[4]+17*sums[5]+14*sums[6]+11*sums[7]+8*sums[8]+5*sums[9]+2*sums[10]-1*sums[11]-4*sums[12]-7*sums[13]-
10*sums[14]-13*sums[15])/120
ma$ma7[2]<-(91*sums[1]+82*sums[2]+73*sums[3]+64*sums[4]+55*sums[5]+46*sums[6]+37*sums[7]+28*sums[8]+19*sums[9]+10*sums[10]+1*sums[11]-8*sums[12]-17*sums[13]-
26*sums[14]-35*sums[15])/420
ma$ma7[3]<-(161*sums[1]+146*sums[2]+131*sums[3]+116*sums[4]+101*sums[5]+86*sums[6]+71*sums[7]+56*sums[8]+41*sums[9]+26*sums[10]+11*sums[11]-4*sums[12]-19*sums[13]-
34*sums[14]-49*sums[15])/840
ma$ma7[4]<-(35*sums[1]+32*sums[2]+29*sums[3]+26*sums[4]+23*sums[5]+20*sums[6]+17*sums[7]+14*sums[8]+11*sums[9]+8*sums[10]+5*sums[11]+2*sums[12]-1*sums[13]-
4*sums[14]-7*sums[15])/210
ma$ma7[5]<-
(119*sums[1]+110*sums[2]+101*sums[3]+92*sums[4]+83*sums[5]+74*sums[6]+65*sums[7]+56*sums[8]+47*sums[9]+38*sums[10]+29*sums[11 ]+20*sums[12]+11*sums[13]+2*sums[14]-
7*sums[15])/840
ma$ma7[6]<-
(49*sums[1]+46*sums[2]+43*sums[3]+40*sums[4]+37*sums[5]+34*sums[6]+31*sums[7]+28*sums[8]+25*sums[9]+22*sums[10]+19*sums[11]+16*sums[12]+13*sums[13]+10*sums[14]+7*su
ms[15])/420
ma$ma7[7]<-
(77*sums[1]+74*sums[2]+71*sums[3]+68*sums[4]+65*sums[5]+62*sums[6]+59*sums[7]+56*sums[8]+53*sums[9]+52*sums[10]+49*sums[11]+46*sums[ 12]+43*sums[13]+40*sums[14]+37*s
ums[15])/840
ma$ma7[40]<-(29*sums[40]+26*sums[39]+23*sums[38]+20*sums[37]+17*sums[36]+14*sums[35]+11*sums[34]+8*sums[33]+5*sums[2]+2*sums[10]-1*sums[9]-4*sums[12]-7*sums[13]-
10*sums[14]-13*sums[15])/120
ma$ma7[39]<-(91*sums[40]+82*sums[39]+73*sums[38]+64*sums[37]+55*sums[36]+46*sums[35]+37*sums[34]+28*sums[33]+19*sums[32]+10*sums[31]+1*sums[30]-8*sums[29]-
17*sums[28]-26*sums[27]-35*sums[26])/420
ma$ma7[38]<-(161*sums[40]+146*sums[39]+131*sums[38]+116*sums[37]+101*sums[36]+86*sums[35]+71*sums[34]+56*sums[33]+41*sums[32]+26*sums[31]+11*sums[30]-4*sums[29]-
19*sums[28]-34*sums[27]-49*sums[26])/840
ma$ma7[37]<-(35*sums[40]+32*sums[39]+29*sums[38]+26*sums[37]+23*sums[36]+20*sums[35]+17*sums[34]+14*sums[33]+11*sums[32]+8*sums[31]+5*sums[30]+2*sums[29]-
1*sums[28]-4*sums[27]-7*sums[26])/210
ma$ma7[36]<-
(119*sums[40]+110*sums[39]+101*sums[38]+92*sums[37]+83*sums[36]+74*sums[35]+65*sums[34]+56*sums[33]+47*sums[32]+38*sums[31]+29*sums[30]+20*sums[29]+11*sums[28]+2*su
ms[27]-7*sums[26])/840
ma$ma7[35]<-
(49*sums[40]+46*sums[39]+43*sums[38]+40*sums[37]+37*sums[36]+34*sums[35]+33*sums[34]+30*sums[33]+27*sums[32]+24*sums[31]+21*sums[30]+18*sums[29]+15*sums[28]+12*sums
[27]+9*sums[26])/420
ma$ma7[34]<-
(77*sums[40]+74*sums[39]+71*sums[38]+68*sums[37]+65*sums[36]+61*sums[35]+58*sums[34]+55*sums[33]+52*sums[32]+49*sums[31]+46*sums[30]+43*sums[29]+40*sums[28]+37*sums
[27]+34*sums[26])/840
```

Data visualization:

```
ma %>%
  gather(metric, sums, ma1:ma7) %>%
  ggplot(aes(dates,sums, color = metric)) +
  geom_line(size=1)
```

Visualization of the moving average at k=5:

```
ggplot(viewh,mapping= aes(x=dates)) + geom_line(mapping= aes(y=likes, col="Real"),lwd=1.5) +
  geom_line(mapping= aes(y=ma$ma2, col="ma2"),lwd=1.5)+scale_color_manual(values=
c("Real"="blue","ma2"="red"))+  theme(legend.title = element_blank()) +  labs(x="",y="Likes")
```

Finding turning points:

```
tp1 <- turnpoints(ma$ma1)
summary(tp1)
tp2 <- turnpoints(ma$ma2)
```

```
summary(tp2)
tp3 <- turnpoints(ma$ma3)
summary(tp3)
tp4 <- turnpoints(ma$ma4)
summary(tp4)
tp5 <- turnpoints(ma$ma5)
summary(tp5)
tp6 <- turnpoints(ma$ma6)
summary(tp6)
tp7 <- turnpoints(ma$ma7)
summary(tp7)
```

Visualization of turning points:

```
plot(ma$ma2, type = "l")
lines(tp2)
```

We are looking for the correlation coefficients of the smoothed values with the original ones, taking into account the fact that with each smoothing we subtract rows:

```
cor(viewh$views,ma$ma1)
cor(viewh$views,ma$ma2)
cor(viewh$views,ma$ma3)
cor(viewh$views,ma$ma4)
cor(viewh$views,ma$ma5)
cor(viewh$views,ma$ma6)
cor(viewh$views,ma$ma7)
```

Similarly, we do research for likes and dislikes. To implement Pollard's formula, we will use the built-in method wma():

```
wma <- viewh %>%
  select(dates, views) %>%
  mutate(wma1 = WMA(views, n = 3, wts = 1:3), wma2 = WMA(views, n = 5, wts = 1:5),
         wma3 = WMA(views, n = 7, wts = 1:7), wma4 = WMA(views, n = 9, wts = 1:9),
         wma5 = WMA(views, n = 11, wts = 1:11), wma6 = WMA(views, n = 13, wts = 1:13),
         wma7 = WMA(views, n = 15, wts = 1:15))
```

Using Kendel's formulas:

```
sums<-views
wma$wma1[1]<-(5*sums[1]+2*sums[2]-sums[3])/6
wma$wma1[2]<-(-sums[38]+2*sums[39]+5*sums[40])/6
wma$wma2[1]<-(3*sums[1]+2*sums[2]+sums[3]-sums[5])/5
wma$wma2[2]<-(4*sums[1]+3*sums[2]+2*sums[3]+sums[4])/10
wma$wma2[3]<-(4*sums[40]+3*sums[39]+2*sums[38]+sums[37])/10
wma$wma2[4]<-(-sums[36]+sums[38]+2*sums[39]+3*sums[40])/5
wma$wma3[1]<-(13*sums[1]+10*sums[2]+7*sums[3]+4*sums[4]+1*sums[5]-2*sums[6]-5*sums[7])/28
wma$wma3[2]<-(5*sums[1]+4*sums[2]+3*sums[3]+2*sums[4]+1*sums[5]+0*sums[6]-1*sums[7])/14
wma$wma3[3]<-(7*sums[1]+6*sums[2]+5*sums[3]+4*sums[4]+3*sums[5]+2*sums[6]+sums[7])/28
wma$wma3[4]<-(7*sums[34]+6*sums[35]+5*sums[36]+4*sums[37]+3*sums[38]+2*sums[39]+sums[40])/28
wma$wma3[5]<-(5*sums[40]+4*sums[39]+3*sums[38]+2*sums[37]+1*sums[36]+0*sums[35]+1*sums[34])/14
wma$wma3[6]<-(13*sums[40]+10*sums[39]+7*sums[38]+4*sums[37]+sums[36]-2*sums[35]-5*sums[34])/28
wma$wma4[1]<-(17*sums[1]+14*sums[2]+11*sums[3]+8*sums[4]+5*sums[5]+2*sums[6]-1*sums[7]-4*sums[8]-7*sums[9])/45
wma$wma4[2]<-(56*sums[1]+47*sums[2]+38*sums[3]+29*sums[4]+20*sums[5]+11*sums[6]+2*sums[7]-7*sums[8]-16*sums[9])/180
wma$wma4[3]<-(22*sums[1]+19*sums[2]+16*sums[3]+13*sums[4]+10*sums[5]+7*sums[6]+4*sums[7]+sums[8]-2*sums[9])/90
wma$wma4[4]<-(32*sums[1]+29*sums[2]+26*sums[3]+23*sums[4]+20*sums[5]+17*sums[6]+14*sums[7]+11*sums[8]+8*sums[9])/180
wma$wma4[5]<-(32*sums[40]+29*sums[39]+26*sums[38]+23*sums[37]+20*sums[36]+17*sums[35]+14*sums[34]+11*sums[33]+8*sums[32])/180
wma$wma4[6]<-(22*sums[40]+19*sums[39]+16*sums[38]+13*sums[37]+10*sums[36]+7*sums[35]+4*sums[34]+sums[33]-2*sums[32])/90
wma$wma4[7]<-(56*sums[40]+47*sums[39]+38*sums[38]+29*sums[37]+20*sums[36]+11*sums[35]+2*sums[34]-7*sums[33]-16*sums[32])/180
wma$wma4[8]<-(17*sums[40]+14*sums[39]+11*sums[38]+8*sums[37]+5*sums[36]+2*sums[35]-1*sums[34]-4*sums[33]-7*sums[32])/45
wma$wma5[1]<-(7*sums[1]+6*sums[2]+5*sums[3]+4*sums[4]+3*sums[5]+2*sums[6]+1*sums[7]+0*sums[8]-1*sums[9]-2*sums[10]-3*sums[11])/22
wma$wma5[2]<-(15*sums[1]+13*sums[2]+11*sums[3]+9*sums[4]+7*sums[5]+5*sums[6]+3*sums[7]+1*sums[8]-1*sums[9]-3*sums[10]-5*sums[11])/55
wma$wma5[3]<-(25*sums[1]+22*sums[2]+19*sums[3]+16*sums[4]+13*sums[5]+10*sums[6]+7*sums[7]+4*sums[8]+sums[9]-2*sums[10]-5*sums[11])/110
wma$wma5[4]<-(10*sums[1]+9*sums[2]+8*sums[3]+7*sums[4]+6*sums[5]+5*sums[6]+4*sums[7]+3*sums[8]+2*sums[9]+1*sums[10]+0*sums[11])/55
wma$wma5[5]<-(15*sums[1]+14*sums[2]+13*sums[3]+12*sums[4]+11*sums[5]+10*sums[6]+9*sums[7]+8*sums[8]+7*sums[9]+6*sums[10]+5*sums[11])/110
wma$wma5[6]<-(15*sums[40]+14*sums[39]+13*sums[38]+12*sums[37]+11*sums[37]+10*sums[36]+9*sums[35]+8*sums[34]+7*sums[33]+6*sums[32]+5*sums[31])/110
wma$wma5[7]<-(10*sums[40]+9*sums[39]+8*sums[38]+7*sums[37]+6*sums[37]+5*sums[36]+4*sums[35]+3*sums[34]+2*sums[33]+1*sums[32]+0*sums[31])/55
wma$wma5[8]<-(25*sums[40]+22*sums[39]+19*sums[38]+16*sums[37]+13*sums[37]+10*sums[36]+7*sums[35]+4*sums[34]+sums[33]-2*sums[32]-5*sums[31])/110
wma$wma5[9]<-(15*sums[40]+13*sums[39]+11*sums[38]+9*sums[37]+7*sums[37]+5*sums[36]+3*sums[35]+1*sums[34]-1*sums[33]-3*sums[32]-5*sums[31])/55
wma$wma5[10]<-(7*sums[40]+6*sums[39]+5*sums[38]+4*sums[37]+3*sums[37]+2*sums[36]+1*sums[35]+0*sums[34]-1*sums[33]-2*sums[32]-3*sums[31])/22
wma$wma6[1]<-(25*sums[1]+22*sums[2]+19*sums[3]+16*sums[4]+13*sums[5]+10*sums[6]+7*sums[7]+4*sums[8]+1*sums[9]-2*sums[10]-5*sums[11]-8*sums[12]-11*sums[13])/91
wma$wma6[2]<-(44*sums[1]+39*sums[2]+34*sums[3]+29*sums[4]+24*sums[5]+19*sums[6]+14*sums[7]+9*sums[8]+4*sums[9]-1*sums[10]-6*sums[11]-11*sums[12]-15*sums[13])/182
wma$wma6[3]<-(19*sums[1]+17*sums[2]+15*sums[3]+13*sums[4]+11*sums[5]+9*sums[6]+7*sums[7]+5*sums[8]+3*sums[9]+1*sums[10]-1*sums[11]-3*sums[12]-5*sums[13])/91
wma$wma6[4]<-(32*sums[1]+29*sums[2]+26*sums[3]+23*sums[4]+20*sums[5]+17*sums[6]+14*sums[7]+11*sums[8]+8*sums[9]+5*sums[10]+2*sums[11]-1*sums[12]-4*sums[13])/182
wma$wma6[5]<-(13*sums[1]+12*sums[2]+11*sums[3]+10*sums[4]+9*sums[5]+8*sums[6]+7*sums[7]+6*sums[8]+5*sums[9]+4*sums[10]+3*sums[11]+2*sums[12]+1*sums[13])/91
wma$wma6[6]<-(20*sums[1]+19*sums[2]+18*sums[3]+17*sums[4]+16*sums[5]+15*sums[6]+14*sums[7]+13*sums[8]+12*sums[9]+11*sums[10]+10*sums[11]+9*sums[12]+8*sums[13])/182
wma$wma6[7]<-(25*sums[40]+22*sums[39]+19*sums[38]+16*sums[37]+13*sums[36]+10*sums[35]+7*sums[34]+4*sums[33]+1*sums[32]-2*sums[31]-5*sums[30]-8*sums[29]-
11*sums[28])/91
wma$wma6[8]<-(44*sums[40]+39*sums[39]+34*sums[38]+29*sums[37]+24*sums[36]+19*sums[35]+14*sums[34]+9*sums[33]+4*sums[32]-1*sums[31]-6*sums[30]-11*sums[29]-
15*sums[28])/182
wma$wma6[9]<-(19*sums[40]+17*sums[39]+15*sums[38]+13*sums[37]+11*sums[36]+9*sums[35]+7*sums[34]+5*sums[33]+3*sums[32]+1*sums[31]-1*sums[30]-3*sums[29]-
5*sums[28])/91
wma$wma6[10]<-(32*sums[40]+29*sums[39]+26*sums[38]+23*sums[37]+20*sums[36]+17*sums[35]+14*sums[34]+11*sums[33]+8*sums[32]+5*sums[31]+2*sums[30]-1*sums[29]-
4*sums[28])/182
wma$wma6[11]<-
(13*sums[40]+12*sums[39]+11*sums[38]+10*sums[37]+9*sums[36]+8*sums[35]+7*sums[34]+6*sums[33]+5*sums[32]+4*sums[31]+3*sums[30]+2*sums[29]+1*sums[28])/91
wma$wma6[12]<-
(20*sums[40]+19*sums[39]+18*sums[38]+17*sums[37]+16*sums[36]+15*sums[35]+14*sums[34]+13*sums[33]+12*sums[32]+11*sums[31]+10*sums[30]+9*sums[29]+8*sums[28])/182
wma$wma7[1]<-(29*sums[1]+26*sums[2]+23*sums[3]+20*sums[4]+17*sums[5]+14*sums[6]+11*sums[7]+8*sums[8]+5*sums[9]+2*sums[10]-1*sums[11]-4*sums[12]-7*sums[13]-
10*sums[14]-13*sums[15])/120
wma$wma7[2]<-(91*sums[1]+82*sums[2]+73*sums[3]+64*sums[4]+55*sums[5]+46*sums[6]+37*sums[7]+28*sums[8]+19*sums[9]+10*sums[10]+1*sums[11]-8*sums[12]-17*sums[13]-
26*sums[14]-35*sums[15])/420
wma$wma7[3]<-(161*sums[1]+146*sums[2]+131*sums[3]+116*sums[4]+101*sums[5]+86*sums[6]+71*sums[7]+56*sums[8]+41*sums[9]+26*sums[10]+11*sums[11]-4*sums[12]-
19*sums[13]-34*sums[14]-49*sums[15])/840
wma$wma7[4]<-(35*sums[1]+32*sums[2]+29*sums[3]+26*sums[4]+23*sums[5]+20*sums[6]+17*sums[7]+14*sums[8]+11*sums[9]+8*sums[10]+5*sums[11]+2*sums[12]-1*sums[13]-
4*sums[14]-7*sums[15])/210
wma$wma7[5]<-
(119*sums[1]+110*sums[2]+101*sums[3]+92*sums[4]+83*sums[5]+74*sums[6]+65*sums[7]+56*sums[8]+47*sums[9]+38*sums[10]+29*sums[11]+20*sums[12]+11*sums[13]+2*sums[14]-
7*sums[15])/840
wma$wma7[6]<-
(49*sums[1]+46*sums[2]+43*sums[3]+40*sums[4]+37*sums[5]+34*sums[6]+31*sums[7]+28*sums[8]+25*sums[9]+22*sums[10]+19*sums[11]+16*sums[12]+13*sums[13]+10*sums[14]+7*su
ms[15])/420
```

```
wma$wma7[7]<-
(77*sums[1]+74*sums[2]+71*sums[3]+68*sums[4]+65*sums[5]+62*sums[6]+59*sums[7]+56*sums[8]+53*sums[9]+52*sums[10]+49*sums[11]+46*sums[12]+43*sums[13]+40*sums[14]+37*s
ums[15])/840
wma$wma7[8]<-(29*sums[40]+26*sums[39]+23*sums[38]+20*sums[37]+17*sums[36]+14*sums[35]+11*sums[34]+8*sums[33]+5*sums[2]+2*sums[10]-1*sums[9]-4*sums[12]-7*sums[13]-
10*sums[14]-13*sums[15])/120
wma$wma7[9]<-(91*sums[40]+82*sums[39]+73*sums[38]+64*sums[37]+55*sums[36]+46*sums[35]+37*sums[34]+28*sums[33]+19*sums[32]+10*sums[31]+1*sums[30]-8*sums[29]-
17*sums[28]-26*sums[27]-35*sums[26])/420
wma$wma7[10]<-(161*sums[40]+146*sums[39]+131*sums[38]+116*sums[37]+101*sums[36]+86*sums[35]+71*sums[34]+56*sums[33]+41*sums[32]+26*sums[31]+11*sums[30]-4*sums[29]-
19*sums[28]-34*sums[27]-49*sums[26])/840
wma$wma7[11]<-(35*sums[40]+32*sums[39]+29*sums[38]+26*sums[37]+23*sums[36]+20*sums[35]+17*sums[34]+14*sums[33]+11*sums[32]+8*sums[31]+5*sums[30]+2*sums[29]-
1*sums[28]-4*sums[27]-7*sums[26])/210
wma$wma7[12]<-
(119*sums[40]+110*sums[39]+101*sums[38]+92*sums[37]+83*sums[36]+74*sums[35]+65*sums[34]+56*sums[33]+47*sums[32]+38*sums[31]+29*sums[30]+20*sums[29]+11*sums[28]+2*su
ms[27]-7*sums[26])/840
wma$wma7[13]<-
(49*sums[40]+46*sums[39]+43*sums[38]+40*sums[37]+37*sums[36]+34*sums[35]+33*sums[34]+30*sums[33]+27*sums[32]+24*sums[31]+21*sums[30]+18*sums[29]+15*sums[28]+12*sums
[27]+9*sums[26])/420
wma$wma7[14]<-
(77*sums[40]+74*sums[39]+71*sums[38]+68*sums[37]+65*sums[36]+61*sums[35]+58*sums[34]+55*sums[33]+52*sums[32]+49*sums[31]+46*sums[30]+43*sums[29]+40*sums[28]+37*sums
[27]+34*sums[26])/840
```

Visualization of all graphs:

```
wma %>%
  gather(metric, views, views:wma7) %>%
  ggplot(aes(dates, views, color = metric)) + geom_line(size=1)
```

Graph visualization of real data and weighted data:

```
ggplot(viewh,mapping= aes(x=dates)) + geom_line(mapping= aes(y=views, col="Real"),lwd=1.5) +
  geom_line(mapping= aes(y=wma$wma2, col="ma2"),lwd=1.5)+
  scale_color_manual(values= c("Real"="blue","ma2"="red"))+ theme(legend.title = element_blank()) +
  labs(x="",y="Likes")
```

Turning points:

```
tp1 <- turnpoints(wma$wma1)
summary(tp1)
tp2 <- turnpoints(wma$wma2)
summary(tp2)
tp3 <- turnpoints(wma$wma3)
summary(tp3)
tp4 <- turnpoints(wma$wma4)
summary(tp4)
tp5 <- turnpoints(wma$wma5)
summary(tp5)
tp6 <- turnpoints(wma$wma6)
summary(tp6)
tp7 <- turnpoints(wma$wma7)
summary(tp7)
```

Visualization of turning points:

```
plot(wma$wma1, type = "l")
lines(tp1)
```

Correlation coefficients of weighted smoothed data with the original:

```
cor(wma$views,wma$wma1)
cor(wma$views,wma$wma2)
cor(wma$views,wma$wma3)
cor(wma$views,wma$wma4)
cor(wma$views,wma$wma5)
cor(wma$views,wma$wma6)
cor(wma$views,wma$wma7)
```

Exponential smoothing:

```
alpha<-0.1
sums<-views
exp_smooth<-1:40
exp_smooth[1]<-sums[1]
for(i in 2:40){
  exp_smooth[i]<-sums[i]*alpha +(1-alpha)*exp_smooth[i-1]
}
viewh<-data.frame(dates,sums,exp_smooth) #save date into structure
```

Visualization:

```
ggplot(viewh,mapping= aes(x=dates)) + geom_line(mapping= aes(y=sums, col="Real"),lwd=1.5) +
  geom_line(mapping= aes(y=exp_smooth, col="es"),lwd=1.5)+
  scale_color_manual(values= c("Real"="blue","es"="red"))+ labs(x="",y="Views",title ="alpha = 0.3")+
  theme(legend.title = element_blank(),plot.title = element_text(hjust = 0.5))
```

Turning points:

```
tp_es<-turnpoints(exp_smooth)
summary(tp_es)
```

Visualization of turning points:

```
plot(views, type = "l")
lines(tp_es)
```

Correlation coefficient of smoothed and real values: `cor(views,exp_smooth)`

Median filtering

```
med_fil<-1:40
med_fil[1]<-(5*sums[1]+2*sums[2]-sums[3])/6
med_fil[40]<-(-sums[38]+2*sums[39]+5*sums[40])/6
for(i in 2:39){med_fil[i]<-max(min(sums[i-1],sums[i]),min(sums[i],sums[i+1]),min(sums[i-1],sums[i+1]))}
viewh<-data.frame(dates,views,med_fil)
```

Visualization:

```
ggplot(viewh,mapping= aes(x=dates)) + geom_line(mapping= aes(y=views, col="Real"),lwd=1.5) +
  geom_line(mapping= aes(y=med_fil, col="Median"),lwd=1.5)+
  scale_color_manual(values= c("Real"="blue","Median"="red"))+
  labs(x="",y="Views",title ="Median filter")+
  theme(legend.title = element_blank(),plot.title = element_text(hjust = 0.5))
```

Turning points:

```
tp_mf<-turnpoints(med_fil)
summary(tp_mf)
```

Visualization of turning points:

```
plot(views, type = "l")
lines(tp_mf)
```

Correlation coefficient: `cor(views,med_fil)`

In general, correlation can be described as any statistical relationship of data. Correlation allows us to see the trends of changes in the average values of the functions depending on the parameter changes. Correlation can be positive or negative. Negative correlation is a correlation in which an increase in one variable is associated with a decrease in another, and the correlation coefficient is negative. Positive correlation is a correlation in which an increase in one variable is associated with an increase in another, and the correlation coefficient is positive.

Construction of the correlation field (plot)

```
plot(dt$likes, dt$dislikes, main="Correlation field",xlab="Likes", ylab="Dislikes")
```

Correlation coefficient: `cor(dt$views, dt$dislikes)`

Correlation relation:

```
ggscatter(dt, x = 'likes', y = 'dislikes', add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "pearson", xlab = "Likes", ylab = "Dislikes")
```

Construction of graphs of autocorrelation functions:

```
data <- cbind(dt$likes, dt$dislikes)
colnames(data) <- c("Likes", "Dislikes")
autocorelation <- acf(data, lag.max = 1,type = c("correlation"), plot = TRUE,
                      xlab="Likes", ylab="Dislikes")
```

Separation of data into 3 parts:

```
part1 <- dt$likes[1:2666]
part2 <- dt$likes[2667:5332]
part3 <- dt$likes[5333:7998]
```

A correlation matrix was constructed for the parts (rcorr):

```
mydata.rcorr = rcorr(as.matrix(cbind(part1, part2,  part3)))
```

Finding multiple correlation coefficients:

```
numericData <- cbind(dt$id, dt$views, dt$likes, dt$dislikes, dt$category_id, dt$comment_total)
chart.Correlation(numericData, histogram=TRUE, pch=19)
```

Cluster analysis itself is not a separate algorithm, but a general task that needs to be solved. Therefore, this general task consists in grouping objects in such a way that the grouped objects are more similar to each other compared to other grouped objects, and the given groups are called clusters, and to conduct an analysis of these clusters through experiments. This analysis can be carried out with the help of different algorithms, although the concept of a "cluster" and how to find it can differ greatly between these algorithms, it is the understanding of the cluster model of this or that algorithm that is the key stage to a successful research analysis. Construction of a graphical representation of clustering:

```
library(ggplot2)
library(factoextra)
ggplot(dt, aes(likes, views, col=dislikes)) + geom_point()
```

K-means clustering and clustering matrix:

```
set.seed(55)
cluster <- kmeans(cbind(dt$channel_title, dt$dislikes), 4, nstart = 20)
cluster
table(cluster$cluster, dt$category_id)
```

Construction of a dendrogram:

```
rdata <- cbind(dt$category_id)
rdata <- na.omit(rdata)
data.hclust = hclust(dist(rdata), method = "single")
plot(data.hclust, labels = FALSE, hang = -1)
```

5. Results

Display likes and dislikes on a graph over time:

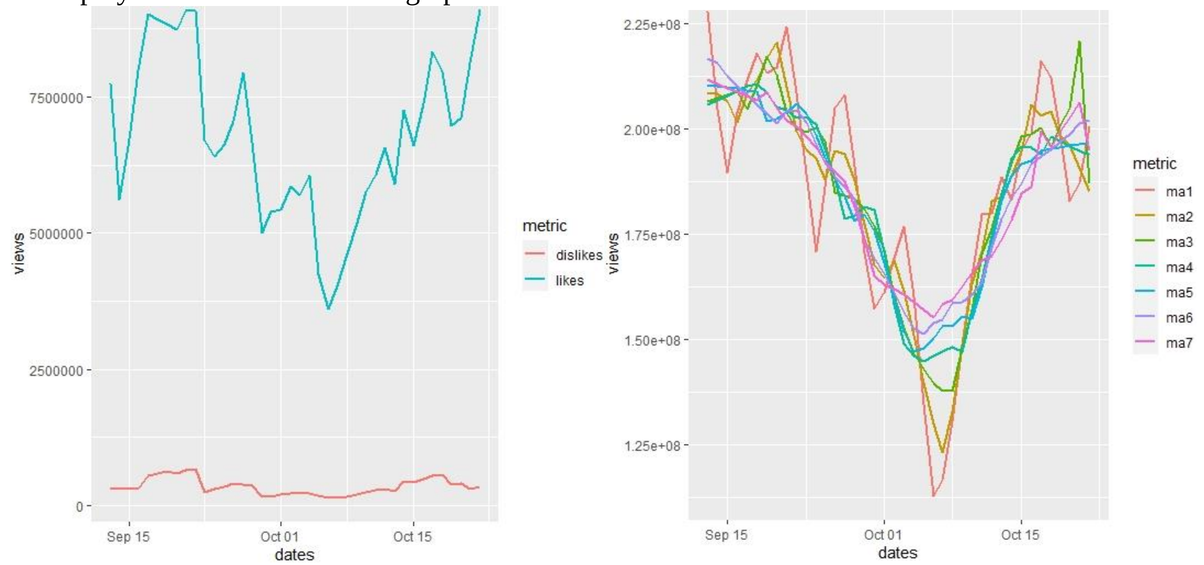
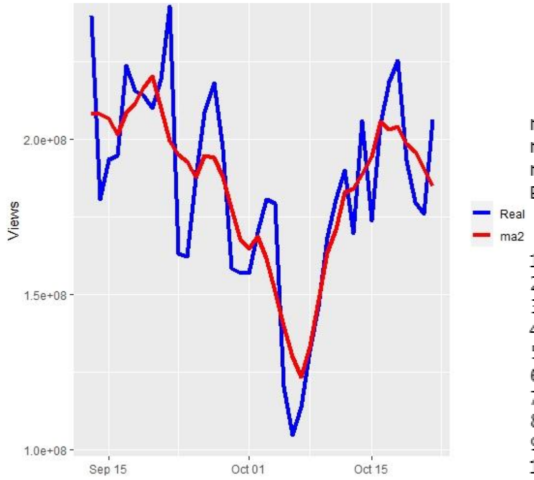


Figure 10: Graph of likes and dislikes over time and Graphs of smoothed data at k=3,5,7,9,11,13,15

From the graph, you can already see a slight dependence between the number of likes and dislikes. Using Kendel's formulas, we obtained initial and final values that were lost in the calculation of averages, depending on the average by which we calculate the data. It can be noted that ma4, ma5, ma6, ma7 are not very suitable for identifying trends, since we do not have a large date interval, only 40 days. For more accurate detection of trends, it is advisable to take ma1, ma2 or ma3.

	dates	views	ma1	ma2	ma3	ma4	ma5	ma6	ma7
1	2021-09-13	239818781	227850762	210111512	206392385	205597607	210517670	216760117	211578721
2	2021-09-14	180704846	204640884	208302773	207230618	206748773	210192594	215606497	210634805
3	2021-09-15	193399025	189647963	206494034	208068852	207899939	209867517	212390940	209690889
4	2021-09-16	194840017	203982180	201652314	208907085	209051106	209542441	210206351	208746973
5	2021-09-17	223707499	211385900	208365193	204637884	210202272	209217365	208021762	207803057
6	2021-09-18	215610185	217862308	211672263	210185260	210553690	208892288	205837174	206859141
7	2021-09-19	214269241	213271267	216611556	217268477	208587949	201820779	203652585	208731294
8	2021-09-20	209934374	214580032	220466365	212721786	205102744	202450907	201268056	204971308
9	2021-09-21	219536480	224150800	209946964	203911025	204302326	203852806	204157384	202106140
10	2021-09-22	242981546	208510401	199499550	199914750	202648150	205982451	204422171	200617011
11	2021-09-23	163013177	189342299	195039927	199136274	202943252	203540065	201616625	198186548
12	2021-09-24	162032174	170893869	192896613	200326522	201006809	198336222	196480824	195663034
13	2021-09-25	187636257	186162781	187953526	197084347	195277203	193124662	191971371	192089864
14	2021-09-26	208819913	204907427	194719141	184996686	188322270	188311292	188574503	189751709
15	2021-09-27	218266110	207975759	193986291	184129387	178767353	183817971	186313232	187435221
16	2021-09-28	196841253	191158428	187847455	183408690	179555885	178141272	183235190	181462293
17	2021-09-29	158367921	170717085	177480932	180904933	181612071	179642067	173801259	173800830
18	2021-09-30	156942080	157432432	167849702	176864639	180710479	175851910	169309056	165182869
19	2021-10-01	156987296	161346445	164589021	171329755	170879427	168304482	165592130	163094180
20	2021-10-02	170109959	169211702	168819823	160401069	158251475	159658320	161288021	162046068
21	2021-10-03	180537851	176723246	161499496	152722015	149014906	151787113	156479604	160759139
22	2021-10-04	179521927	160133408	151024945	146546307	146049897	147193408	152638544	158877972
23	2021-10-05	120340446	134825639	139745379	142931385	144868609	148099263	151389401	156993859
24	2021-10-06	104614544	112889039	129974378	139531461	146129168	150250201	153822978	155181411
25	2021-10-07	113712128	116669838	123332090	137787815	147294995	153251758	154800859	158354746
26	2021-10-08	131682843	130568486	132930467	137942168	148346837	153210356	158568601	159466665
27	2021-10-09	146310488	148775220	148128037	147894165	147250460	155522182	158838671	162653067
28	2021-10-10	168332330	165081739	163386497	157185592	156764625	154985723	160703707	165882991
29	2021-10-11	180602398	179646384	170980835	170364994	164431995	162662347	163706545	168870468
30	2021-10-12	190004425	180087119	182912325	176356141	174551017	173020918	171784203	169772323
31	2021-10-13	169654534	188542299	183970034	184709403	184203903	183169816	178586912	173726098
32	2021-10-14	205967937	183081116	188806218	191884616	192986071	188748626	183658830	178492093
33	2021-10-15	173620876	194790710	194517097	198277130	195732452	191779224	187075879	184660754
34	2021-10-16	204783318	198987672	205656191	198712178	195626304	192485782	191685998	186136454
35	2021-10-17	218558821	216230713	203072556	200139682	194081865	194816659	193430102	199353669
36	2021-10-18	225350001	212319528	204277792	195873473	198147143	195389744	195174206	195478990
37	2021-10-19	193049762	199348941	198542024	200533639	197026570	195685981	196918310	199085069
38	2021-10-20	179647060	182933766	196078667	205042049	195905997	195982218	198662414	202691148
39	2021-10-21	176104475	187331191	190562546	220771579	194785424	196278455	201331421	206297227
40	2021-10-22	206242037	200628679	185046424	187008410	193664851	196574692	202150622	194730665

Figure 11: Smoothed data according to Kendel's formulas

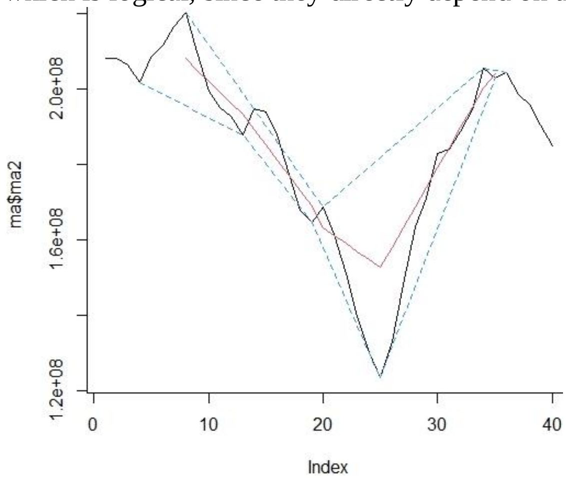



```
nbr observations  : 40
nbr ex-aequos     : 0
nbr turning points: 10 (first point is a pit)
E(p) = 25.33333 Var(p) = 6.788889 (theoretical)

   point type        proba       info
1      4  pit 1.736111e-03  9.1699250
2      8 peak 6.944444e-05 13.8137812
3     13  pit 5.952381e-03  7.3923174
4     14 peak 2.380952e-03  8.7142455
5     19  pit 5.952381e-03  7.3923174
6     20 peak 2.380952e-03  8.7142455
7     25  pit 3.061924e-09 28.2829142
8     34 peak 2.254690e-06 18.7586396
9     35  pit 6.666667e-01  0.5849625
10    36 peak 1.388889e-02  6.1699250
```

Figure 12: Graph of real data and smoothed data at k=5 and Turning points at smoothing k=5

This graph clearly shows when we have a trend change. The correlation coefficients are quite large and positive, which is logical, since they directly depend on the data.




```
> cor(viewh$views,ma$ma1)
[1] 0.930209
> cor(viewh$views,ma$ma2)
[1] 0.8394417
> cor(viewh$views,ma$ma3)
[1] 0.7129804
> cor(viewh$views,ma$ma4)
[1] 0.7108885
> cor(viewh$views,ma$ma5)
[1] 0.7087793
> cor(viewh$views,ma$ma6)
[1] 0.7160276
> cor(viewh$views,ma$ma7)
[1] 0.713408
```

Figure 13: Visualization of turning points and Correlation coefficients of smoothed data and actual

	dates	likes	ma1	ma2	ma3	ma4	ma5	ma6	ma7
1	2021-09-13	7752206	7208945	6440452	6535812	6754481	7269195	7795658	7845270
2	2021-09-14	5613566	6700089	6930221	6969177	7084139	7423373	7833710	7833797
3	2021-09-15	6734495	6777253	7419990	7402542	7413796	7577551	7798939	7822324
4	2021-09-16	7983698	7911392	7653093	7835908	7743453	7731729	7800579	7810850
5	2021-09-17	9015983	8639135	8297116	7975594	8073111	7885906	7802220	7799377
6	2021-09-18	8917724	8922463	8696220	8470318	8220258	8040084	7803860	7787904
7	2021-09-19	8833681	8827139	8914807	8804894	8341684	7917480	7805501	7880755
8	2021-09-20	8730012	8880109	8926916	8622423	8304913	8009613	7752218	7764957
9	2021-09-21	9076633	8961058	8484650	8249220	8154172	8039162	7931430	7693250
10	2021-09-22	9076528	8286520	7998626	7921977	7936789	8035491	7926976	7653140
11	2021-09-23	6706398	7395496	7578029	7668528	7828521	7822819	7698377	7564099
12	2021-09-24	6403561	6578995	7174610	7556142	7588845	7467745	7420139	7393846
13	2021-09-25	6627025	6696707	6947967	7213280	7175723	7155491	7151845	7183089
14	2021-09-26	7059536	7209959	6942007	6632620	6767084	6855481	6922687	6967521
15	2021-09-27	7943315	7226484	6663677	6445832	6361903	6562572	6688394	6782997
16	2021-09-28	6676600	6543941	6418048	6306738	6267263	6254178	6456793	6483114
17	2021-09-29	5011907	5695797	6092121	6196397	6187334	6195944	6084119	6118373
18	2021-09-30	5398883	5280231	5674385	5999920	6124978	5998508	5845590	5781603
19	2021-10-01	5429902	5561140	5475906	5731707	5810781	5723826	5662622	5642230
20	2021-10-02	5854635	5656246	5686689	5382445	5328804	5447957	5507913	5561804
21	2021-10-03	5684201	5868220	5453265	5181532	5034179	5145456	5364654	5503857
22	2021-10-04	6065824	5327262	5088388	4985261	4990168	5010962	5196538	5438411
23	2021-10-05	4231762	4634368	4722458	4868961	4967755	5078772	5150481	5347208
24	2021-10-06	3605519	3954088	4508778	4775037	5004190	5140497	5270739	5295779
25	2021-10-07	4024982	4082102	4335047	4785553	5028992	5244621	5309684	5445766
26	2021-10-08	4615805	4612651	4640257	4787272	5127999	5249215	5450592	5526186
27	2021-10-09	5197165	5190261	5134724	5122058	5110148	5392624	5508327	5656196
28	2021-10-10	5757814	5677612	5644781	5450579	5446809	5441657	5638777	5821273
29	2021-10-11	6077856	6136978	5902654	5912968	5780105	5727865	5813006	5973947
30	2021-10-12	6575264	6186096	6315562	6197166	6152891	6157436	6100893	6034934
31	2021-10-13	5905169	6580713	6485036	6509006	6565667	6516465	6360517	6226894
32	2021-10-14	7261706	6590687	6745475	6876575	6874238	6731449	6597916	6528871
33	2021-10-15	6605186	7082314	7096581	7147496	7010107	6905449	6868636	6867406
34	2021-10-16	7380050	7438677	7510408	7205406	7124918	7121572	7169087	7036505
35	2021-10-17	8330794	7895049	7454193	7377690	7298241	7396587	7382705	7620981
36	2021-10-18	7974303	7761909	7555388	7502471	7653558	7604821	7596324	7602225
37	2021-10-19	6980631	7355365	7706412	7859305	7821346	7776203	7809943	7847165
38	2021-10-20	7111161	7408988	7860858	7719500	7989134	7947585	8023561	8092105
39	2021-10-21	8135172	8116451	8202055	9193207	8156922	8118968	8268816	8337044
40	2021-10-22	9103021	9112381	8543253	8278718	8324710	8290350	8450799	7968084

Figure 14: Smoothed data according to Kendel's formulas

Using Kendel's formulas, we obtained the initial and final values that were lost in the calculation of the averages, depending on the average from which we calculate the data.

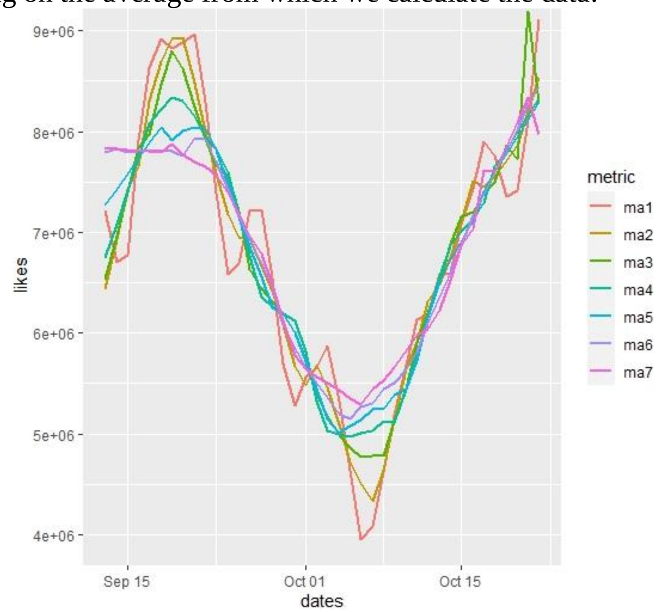
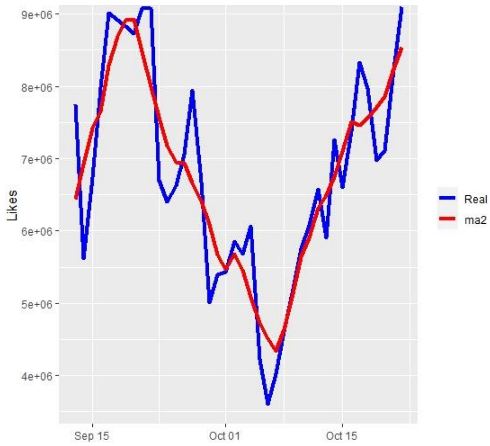


Figure 15: Graphs of structured data at k=3,5,7,9,11,13,15

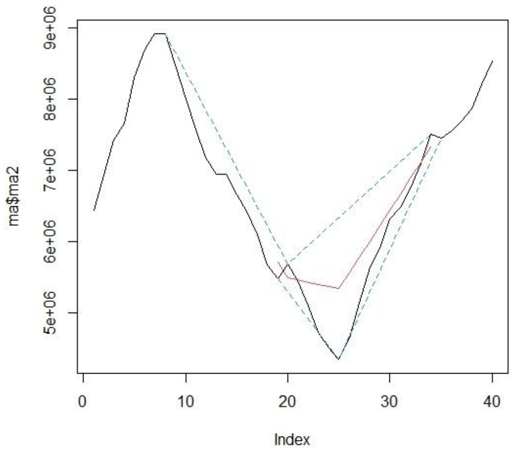
It can be noted that ma4, ma5, ma6, ma7 are not very suitable for identifying trends, since we do not have a large date interval, only 40 days. For more accurate detection of trends, it is advisable to take ma1, ma2 or ma3.




```
nbr observations  : 40
nbr ex-aequos     : 0
nbr turning points: 6 (first point is a peak)
E(p) = 25.33333 Var(p) = 6.788889 (theoretical)

  point type        proba      info
1     8 peak 5.232270e-13 40.797628
2    19  pit 2.119794e-08 25.491501
3    20 peak 2.380952e-03  8.714246
4    25  pit 3.061924e-09 28.282914
5    34 peak 2.254690e-06 18.758640
6    35  pit 2.380952e-03  8.714246
```

Figure 16: Graph of real data and smoothed data at k=5 and Turning points at smoothing k=5




```
> cor(viewh$likes,ma$ma1)
[1] 0.9634537
> cor(viewh$likes,ma$ma2)
[1] 0.905811
> cor(viewh$likes,ma$ma3)
[1] 0.8585738
> cor(viewh$likes,ma$ma4)
[1] 0.8479456
> cor(viewh$likes,ma$ma5)
[1] 0.8404666
> cor(viewh$likes,ma$ma6)
[1] 0.823287
> cor(viewh$likes,ma$ma7)
[1] 0.8143748
```

Figure 17: Visualization of turning points and correlation coefficients of smoothed and actual data

	dates	dislikes	ma1	ma2	ma3	ma4	ma5	ma6	ma7
1	2021-09-13	322647	317815.2	269396.6	244568.1	267702.4	345538.3	408902.8	428059.6
2	2021-09-14	303191	312854.7	313362.3	306140.5	318360.3	369907.1	416776.9	429401.4
3	2021-09-15	312726	307975.7	357328.0	367712.9	369018.1	394275.9	420818.1	430743.3
4	2021-09-16	308010	386934.0	409610.8	429285.3	419675.9	418644.7	426775.7	432085.1
5	2021-09-17	540066	477379.0	475831.8	467705.4	470333.8	443013.5	432733.4	433427.0
6	2021-09-18	584061	586141.0	531604.2	515309.4	508584.8	467382.4	438691.0	434768.8
7	2021-09-19	634296	603315.0	597286.0	565906.6	501707.6	464453.9	444648.7	441655.4
8	2021-09-20	591588	620767.7	622654.0	556376.0	499230.7	468599.5	450416.3	437452.5
9	2021-09-21	636419	631637.7	554101.0	520714.3	503762.1	476317.7	456611.5	439838.7
10	2021-09-22	666906	514873.7	485328.6	487104.6	487935.4	483201.3	460128.0	430125.3
11	2021-09-23	241296	399545.3	436769.6	453294.6	465676.3	466689.8	448549.5	420155.1
12	2021-09-24	290434	293507.7	389011.0	423600.4	435025.7	427910.5	419557.7	412492.3
13	2021-09-25	348793	345617.7	332375.6	383889.1	386792.4	385081.2	389481.3	390704.9
14	2021-09-26	397626	376716.0	355804.4	311115.3	334209.6	348852.1	357093.5	367082.9
15	2021-09-27	383729	379931.7	329215.4	299954.9	281560.9	310382.6	329258.5	339404.4
16	2021-09-28	358440	299886.0	292091.4	286045.4	278445.2	270639.5	297158.4	311425.5
17	2021-09-29	157489	226367.3	251179.8	266682.9	271700.6	268623.4	259081.3	278540.4
18	2021-09-30	163173	171243.3	217085.0	242698.0	257292.2	257847.9	251531.1	243445.9
19	2021-10-01	193068	189832.0	191343.4	219182.1	232212.0	239152.5	239996.8	238008.0
20	2021-10-02	213255	212018.3	203669.2	192534.1	205480.2	215776.4	225453.3	231285.1
21	2021-10-03	229732	220701.7	205415.4	190484.7	181263.4	195412.5	209450.5	223937.6
22	2021-10-04	219118	206918.0	195430.4	187244.1	181512.1	180062.5	198285.3	216595.9
23	2021-10-05	171904	178055.0	180877.2	182481.1	184447.3	187434.5	192828.5	211348.7
24	2021-10-06	143143	151845.3	166876.2	179100.4	189504.3	198737.1	204177.1	204704.4
25	2021-10-07	140489	147786.3	160970.6	180364.6	197753.9	208914.6	211531.1	224247.1
26	2021-10-08	159727	163268.7	174306.0	190133.6	206119.3	213052.8	231343.5	241396.0
27	2021-10-09	189590	195966.0	203177.6	209150.3	210525.7	233134.4	247278.2	260753.7
28	2021-10-10	238581	238557.3	236084.0	225669.1	241495.1	251433.4	266793.8	282598.9
29	2021-10-11	287501	277034.3	255893.6	269974.9	272302.2	279754.3	291548.8	305115.5
30	2021-10-12	305021	283765.7	308101.4	307214.9	310407.2	315917.0	321977.8	315861.9
31	2021-10-13	258775	338141.7	344466.6	349192.6	352763.4	354734.5	340221.7	331406.3
32	2021-10-14	450629	376603.7	383653.2	392385.7	394751.3	374787.8	360574.0	341912.9
33	2021-10-15	420407	451490.0	430835.6	432382.9	410499.4	394376.8	371421.4	355067.1
34	2021-10-16	483434	481591.3	492576.8	443139.0	423562.6	400027.9	382822.3	367174.6
35	2021-10-17	540933	530616.0	478513.8	464038.1	423087.2	404600.7	392605.7	400524.8
36	2021-10-18	567481	496242.7	475446.2	442625.9	431868.0	405833.9	402389.1	403298.9
37	2021-10-19	380314	450954.7	438908.0	430825.1	413073.8	400222.2	412172.4	419376.2
38	2021-10-20	405069	362042.0	398281.8	469385.4	394279.7	394610.6	421955.8	435453.5
39	2021-10-21	300743	347871.3	344388.9	422766.6	375485.5	388998.9	433050.1	451530.8
40	2021-10-22	337802	314237.8	290496.0	315144.3	356691.4	383387.3	441522.6	417375.3

Figure 18: Smoothed data according to Kendel's formulas

Using Kendel's formulas, we obtained the initial and final values that were lost in the calculation of the averages, depending on the average from which we calculate the data.

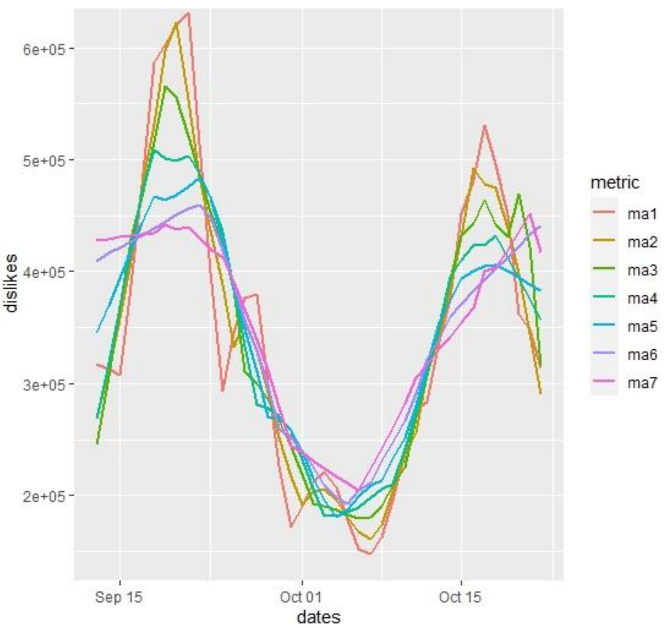


Figure 19: Graphs of structured data at k=3,5,7,9,11,13,15

It can be noted that ma4, ma5, ma6, ma7 are not very suitable for identifying trends, since we do not have a large date interval, only 40 days. For more accurate detection of trends, it is advisable to take ma1, ma2 or ma3.

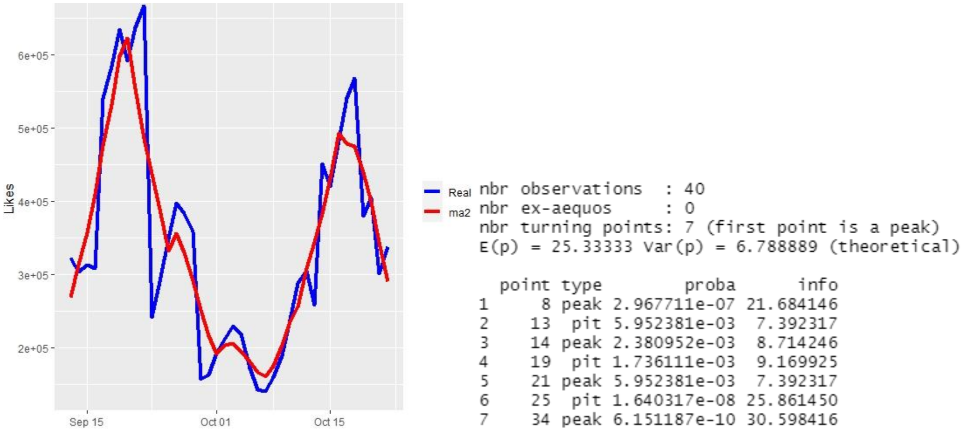


Figure 20: Plot of real data and smoothed data at k=5 and turning points at k=5 smoothing

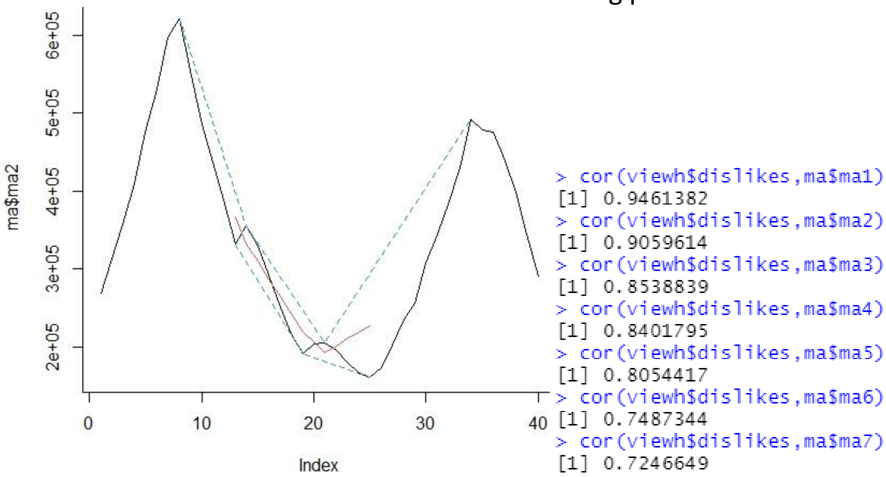


Figure 21: Visualization of turning points and correlation coefficients between smoothed data and actual data

	dates	views	wma1	wma2	wma3	wma4	wma5	wma6	wma7
1	2021-09-13	239818781	227850762	210111512	206392385	205597607	210517670	216760117	211578721
2	2021-09-14	180704846	200628679	208302773	207230618	206748773	210192594	215606497	210634805
3	2021-09-15	193399025	196904258	190562546	208068852	207899939	209867517	212390940	209690889
4	2021-09-16	194840017	192003825	185046424	205042049	209051106	209542441	210206351	208746973
5	2021-09-17	223707499	209033593	205288207	220771579	197026570	209217365	208021762	207803057
6	2021-09-18	215610185	214847595	208326925	187008410	195905997	195389744	205837174	206859141
7	2021-09-19	214269241	216289265	212532567	209745318	194785424	195685981	202150622	208731294
8	2021-09-20	209934374	212325298	213055627	210002140	193664851	195982218	201331421	194730665
9	2021-09-21	219536480	215457905	215677033	213726789	211737160	196278455	198662414	206297227
10	2021-09-22	242981546	229658662	224467029	221925861	218293015	196574692	196918310	202691148
11	2021-09-23	163013177	199089851	205315967	208362036	208784913	208350494	195174206	199085069
12	2021-09-24	162032174	175850737	189344370	195689633	199473758	200540475	193430102	195478990
13	2021-09-25	187636257	174997716	185389939	191620941	195980460	198176388	199283407	199353669
14	2021-09-26	208819913	193960738	189983268	193847232	196883978	199237889	200021597	186136454
15	2021-09-27	218266110	210012402	198439767	198629690	200007570	201640107	202449890	202768837
16	2021-09-28	196841253	205979315	201402342	197758373	198787170	200116574	201404729	201752580
17	2021-09-29	158367921	181175397	189285269	188079267	190259392	192587883	194825550	196285303
18	2021-09-30	156942080	164067223	176937198	181065615	182592367	185688860	188443472	190825937
19	2021-10-01	156987296	157202328	166650479	174280092	176325373	179665965	182801540	185676030
20	2021-10-02	170109959	163541092	164193488	170955409	174593894	176632410	179678481	182481896
21	2021-10-03	180537851	173136795	168422871	170863639	174790287	176085723	178530388	181037894
22	2021-10-04	179521927	178291907	173400506	171527961	174372258	176315832	177560201	179759171
23	2021-10-05	120340446	150100507	157240714	158780634	162298252	166432229	168575238	171372325
24	2021-10-06	104614544	122341075	138279063	144834003	149045275	154559334	158691421	161766356
25	2021-10-07	113712128	111784320	125841457	135081531	140137406	145460609	150749003	154255268
26	2021-10-08	131682843	121181222	123153945	131365665	136670993	140798029	145904819	150067765
27	2021-10-09	146310488	136001546	128599315	132210440	136723111	139885258	143765172	147969803
28	2021-10-10	168332330	154883468	143599395	139410658	141415855	143408412	145458418	148755586
29	2021-10-11	180602398	170797057	159490039	150114303	148310501	148825601	149453255	151235993
30	2021-10-12	190004425	183258400	173448835	163129868	156852387	155451305	154969687	155126800
31	2021-10-13	169654534	178262475	175538181	168569960	161113927	158185101	157231338	156709385
32	2021-10-14	205967937	191202884	187200548	180765546	172857422	166978031	164540920	163057700
33	2021-10-15	173620876	183742173	184103399	181579517	176228672	169994480	166691245	164965966
34	2021-10-16	204783318	194593274	191041160	188686311	184298937	178294080	173254766	170630548
35	2021-10-17	218558821	206477329	200958694	197148665	193100498	187610159	181519782	177618767
36	2021-10-18	225350001	219658494	211236329	205515012	201329717	196331672	190325990	185052144
37	2021-10-19	193049762	208068018	207034186	204208170	201342456	197978330	193363927	188074555
38	2021-10-20	179647060	191731784	199225687	199441890	198125377	196461402	193515377	189308897
39	2021-10-21	176104475	180109551	189834582	193433088	194221011	193848944	192436183	189606195
40	2021-10-22	206242037	191763687	192401253	196025229	196653046	196141653	195174206	193074938

Figure 22: Smoothed data according to formulas from Pollard

The number of turning points allows better analysis of trends. As we can see, the trends of the views attribute are best viewed. This is due to the peculiarity of the data. Correlation coefficients are close to 1 and decrease as the step increases, as less and less data will affect the average. As in the case of a simple moving average, for our data it is better not to use averages with a step greater than 7 to get more accurate information. We can notice a noticeable difference between the graphs of the simple moving average and the weighted moving average. This is due to the fact that the weighted moving average reacts to changes more quickly than the simple moving average. This may allow us to predict, although the results will be quite imprecise, as smoothing methods are not designed to predict.

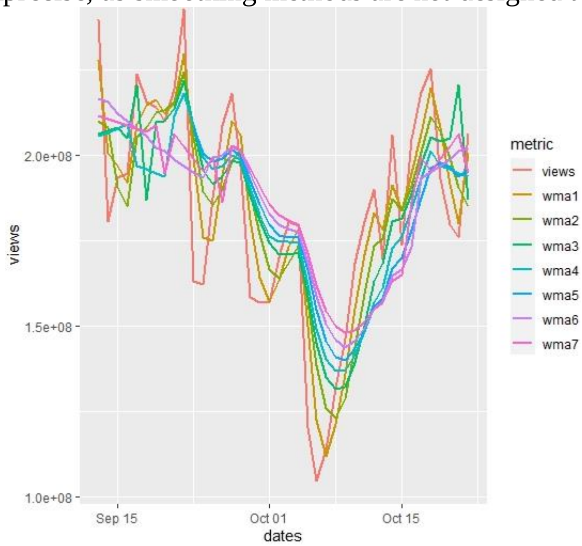
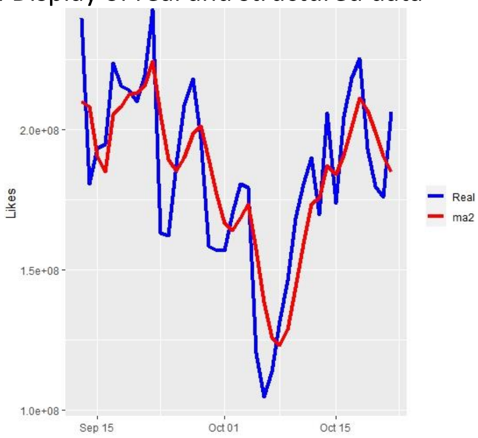


Figure 23: Display of real and structured data

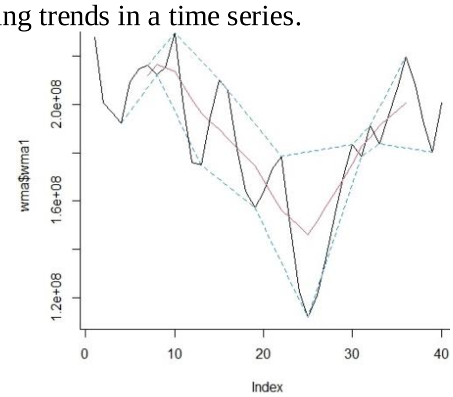



```
nbr observations  : 40
nbr ex-aequos     : 0
nbr turning points: 11 (first point is a pit)
E(p) = 25.33333 Var(p) = 6.788889 (theoretical)

   point type        proba      info
1      4  pit 4.629630e-05 14.398744
2     10 peak 6.944444e-05 13.813781
3     13  pit 7.936508e-03  6.977280
4     16 peak 1.736111e-03  9.169925
5     20  pit 7.936508e-03  6.977280
6     22 peak 5.952381e-03  7.392317
7     26  pit 1.052189e-05 16.536247
8     32 peak 1.041667e-03  9.906891
9     33  pit 6.666667e-02  3.906891
10    36 peak 7.936508e-03  6.977280
11    39  pit 1.000000e-01  3.321928
```

Figure 24: Display of real and smoothed data at w=5 and Pivot points at w = 5

Compared to a simple moving average chart, we can say that the weighted moving average is more suitable for spotting trends in a time series.




```
> cor(wma$views,wma$wma1)
[1] 0.9177515
> cor(wma$views,wma$wma2)
[1] 0.8169842
> cor(wma$views,wma$wma3)
[1] 0.7533848
> cor(wma$views,wma$wma4)
[1] 0.6916134
> cor(wma$views,wma$wma5)
[1] 0.6161428
> cor(wma$views,wma$wma6)
[1] 0.6050743
> cor(wma$views,wma$wma7)
[1] 0.5634467
```

Figure 25: Visualization of turning points and correlation coefficients of real and smoothed data

As the step increases, the correlation coefficient decreases, since the values have less influence on the average. Exponential smoothing directly depends on the latest data, i.e. how the weighted average will react quickly to changes.

	dates	views	exp_smooth
1	2021-09-13	239818781	239818781
2	2021-09-14	180704846	222084601
3	2021-09-15	193399025	213478928
4	2021-09-16	194840017	207887255
5	2021-09-17	223707499	212633328
6	2021-09-18	215610185	213526385
7	2021-09-19	214269241	213749242
8	2021-09-20	209934374	212604781
9	2021-09-21	219536480	214684291
10	2021-09-22	242981546	223173468
11	2021-09-23	163013177	205125380
12	2021-09-24	162032174	192197418
13	2021-09-25	187636257	190829070
14	2021-09-26	208819913	196226323
15	2021-09-27	218266110	202838259
16	2021-09-28	196841253	201039157
17	2021-09-29	158367921	188237786
18	2021-09-30	156942080	178849074
19	2021-10-01	156987296	172290541
20	2021-10-02	170109959	171636366
21	2021-10-03	180537851	174306812
22	2021-10-04	179521927	175871346
23	2021-10-05	120340446	159212076
24	2021-10-06	104614544	142832817
25	2021-10-07	113712128	134096610
26	2021-10-08	131682843	133372480
27	2021-10-09	146310488	137253882
28	2021-10-10	168332330	146577417
29	2021-10-11	180602398	156784911
30	2021-10-12	190004425	166750765
31	2021-10-13	169654534	167621896
32	2021-10-14	205967937	179125708
33	2021-10-15	173620876	177474259
34	2021-10-16	204783318	185666976
35	2021-10-17	218558821	195534530
36	2021-10-18	225350001	204479171
37	2021-10-19	193049762	201050348
38	2021-10-20	179647060	194629362
39	2021-10-21	176104475	189071896
40	2021-10-22	206242037	194222938

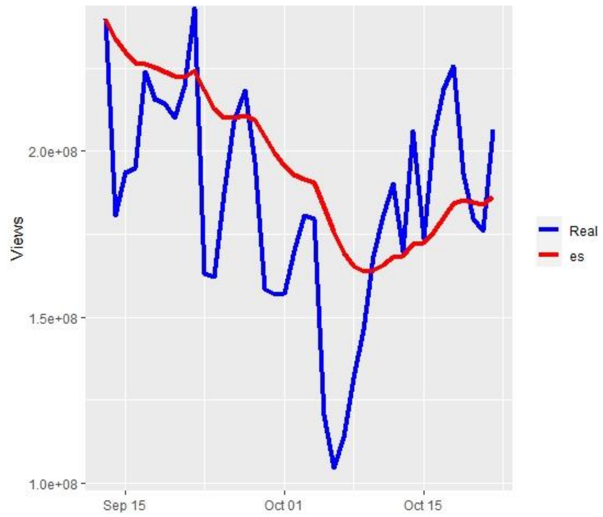


Figure 26: Exponentially smoothed data and visualization of smoothed data at alpha = 0.1\

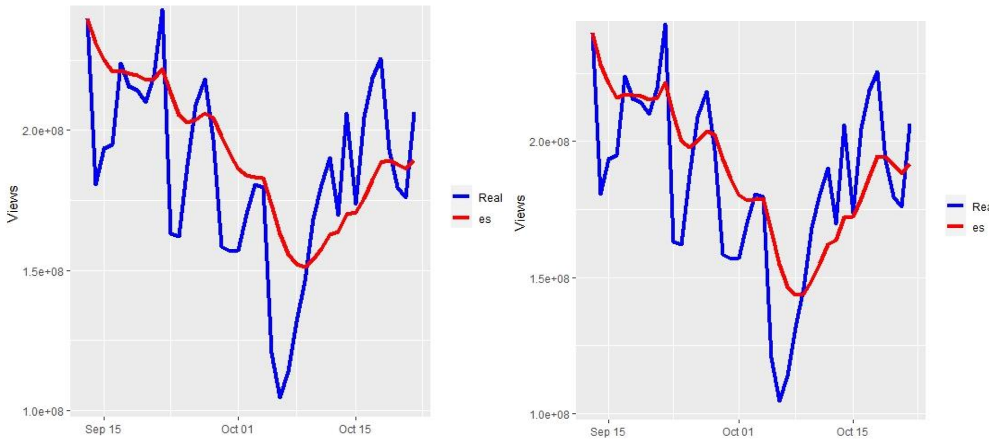


Figure 27: Visualization of smoothed data at alpha = 0.15 and alpha = 0.2

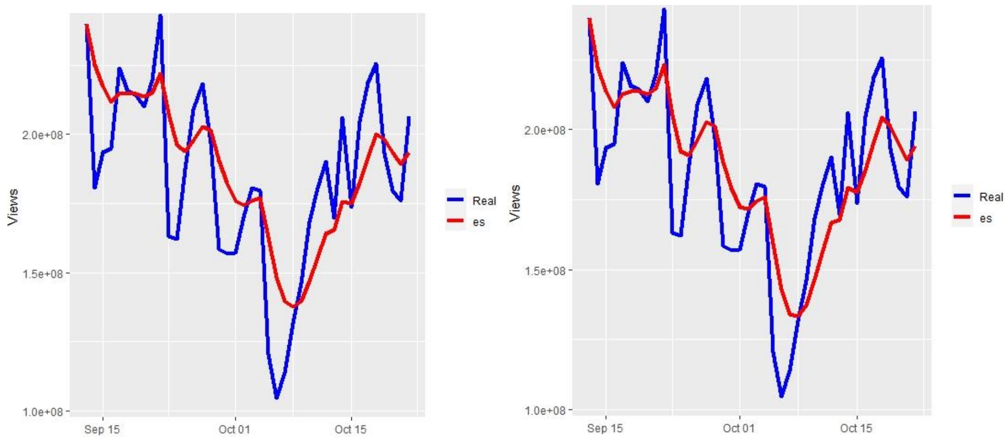


Figure 28: Visualization of smoothed data at alpha = 0.25 and alpha = 0.3

```
nbr observations  : 40
nbr ex-aequos     : 0
nbr turning points: 13 (first point is a pit)
E(p) = 25.33333 Var(p) = 6.788889 (theoretical)

   point type        proba      info
1      4  pit 7.936508e-03  6.977280
2      7 peak 1.000000e-01  3.321928
3      8  pit 2.500000e-01  2.000000
4     10 peak 2.777778e-02  5.169925
5     13  pit 2.777778e-02  5.169925
6     15 peak 1.041667e-03  9.906891
7     20  pit 1.736111e-03  9.169925
8     22 peak 5.952381e-03  7.392317
9     26  pit 1.052189e-05 16.536247
10    32 peak 1.041667e-03  9.906891
11    33  pit 6.666667e-02  3.906891
12    36 peak 7.936508e-03  6.977280
13    39  pit 1.000000e-01  3.321928
```

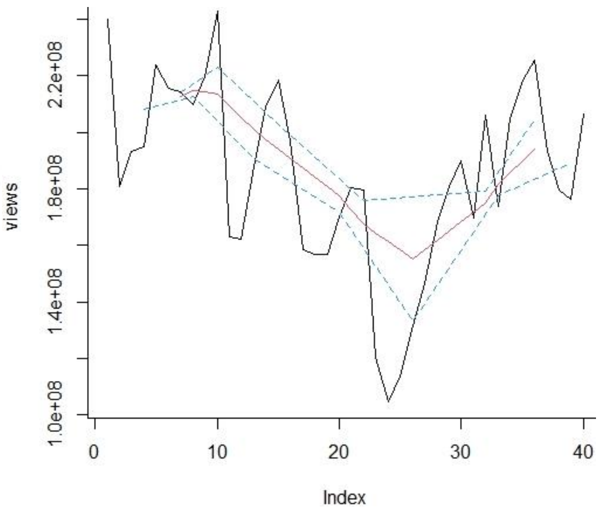


Figure 29: Turning points and visualization of turning points with exponential smoothing, alpha =0.1

```
> cor(views,exp_smooth)
[1] 0.8010819
```

Figure 30: Correlation coefficients of real data and exponentially smoothed data at alpha =0.1

Median smoothing completely removes single extreme or anomalous values of levels that are separated from each other by at least half of the smoothing interval; preserves sharp changes in the trend (moving average and exponential smoothing smooth them); effectively removes single levels with very large or very small values that are random in nature and stand out sharply from other levels.

	dates	views	med_fil
1	2021-09-13	239818781	227850762
2	2021-09-14	180704846	193399025
3	2021-09-15	193399025	193399025
4	2021-09-16	194840017	194840017
5	2021-09-17	223707499	215610185
6	2021-09-18	215610185	215610185
7	2021-09-19	214269241	214269241
8	2021-09-20	209934374	214269241
9	2021-09-21	219536480	219536480
10	2021-09-22	242981546	219536480
11	2021-09-23	163013177	163013177
12	2021-09-24	162032174	163013177
13	2021-09-25	187636257	187636257
14	2021-09-26	208819913	208819913
15	2021-09-27	218266110	208819913
16	2021-09-28	196841253	196841253
17	2021-09-29	158367921	158367921
18	2021-09-30	156942080	156987296
19	2021-10-01	156987296	156987296
20	2021-10-02	170109959	170109959
21	2021-10-03	180537851	179521927
22	2021-10-04	179521927	179521927
23	2021-10-05	120340446	120340446
24	2021-10-06	104614544	113712128
25	2021-10-07	113712128	113712128
26	2021-10-08	131682843	131682843
27	2021-10-09	146310488	146310488
28	2021-10-10	168332330	168332330
29	2021-10-11	180602398	180602398
30	2021-10-12	190004425	180602398
31	2021-10-13	169654534	190004425
32	2021-10-14	205967937	173620876
33	2021-10-15	173620876	204783318
34	2021-10-16	204783318	204783318
35	2021-10-17	218558821	218558821
36	2021-10-18	225350001	218558821
37	2021-10-19	193049762	193049762
38	2021-10-20	179647060	179647060
39	2021-10-21	176104475	179647060
40	2021-10-22	206242037	200628679

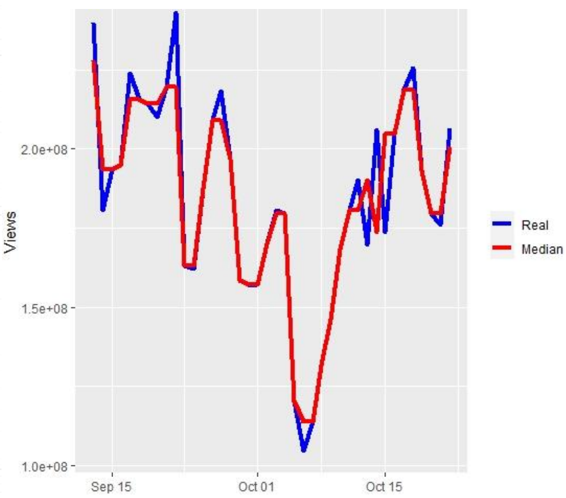


Figure 31: Median filtering and visualization of median filtering

As can be seen from fig. 31-33, median filtering eliminated random levels that are random in nature. As a result, we have a more stable schedule.

```
nbr observations  : 40
nbr ex-aequos     : 13
nbr turning points: 13 (first point is a pit)
E(p) = 25.33333 Var(p) = 6.788889 (theoretical)

   point type       proba      info
1      3  pit 0.250000000 2.0000000
2      6 peak 0.250000000 2.0000000
3      8  pit 0.666666667 0.5849625
4     10 peak 0.666666667 0.5849625
5     12  pit 0.250000000 2.0000000
6     15 peak 0.027777778 5.1699250
7     19  pit 0.027777778 5.1699250
8     22 peak 0.100000000 3.3219281
9     25  pit 0.001041667 9.9068906
10    31 peak 0.005952381 7.3923174
11    32  pit 0.250000000 2.0000000
12    36 peak 0.100000000 3.3219281
13    39  pit 0.250000000 2.0000000
```

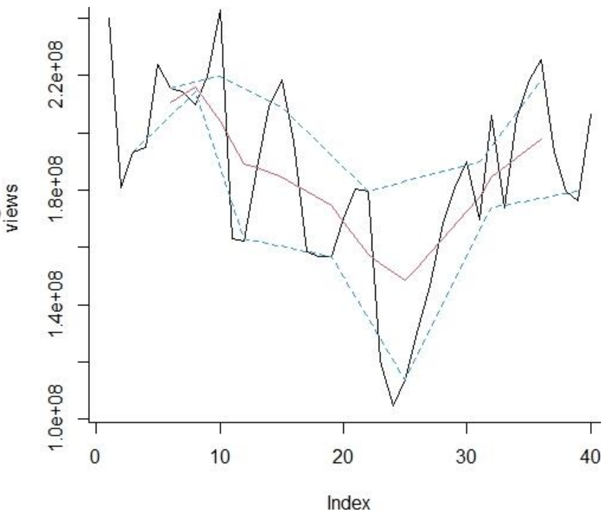


Figure 32: Turning points and visualization of turning points in median filtering

```
> cor(views,med_fil)
[1] 0.9556164
```

Figure 33: Correlation coefficient

Note that the correlation is high, because the median filtering does not calculate, does not generalize, but shows the median on a certain interval. That is why median filtering is very effective when studying time series.

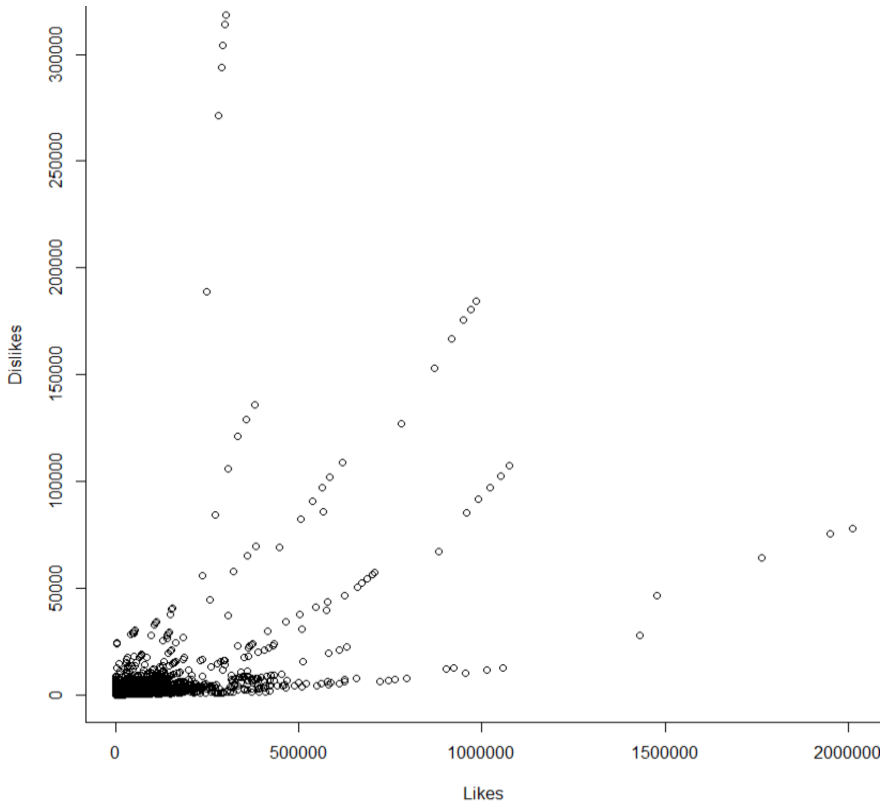


Figure 34: Correlation field of Likes and Dislikes indicators

```
> cor(dt$views, dt$dislikes)
[1] 0.5419003
```

Figure 35: Correlation coefficient

This value shows us a rather obvious influence of the number of views on the number of dislikes of videos on the YouTube platform.

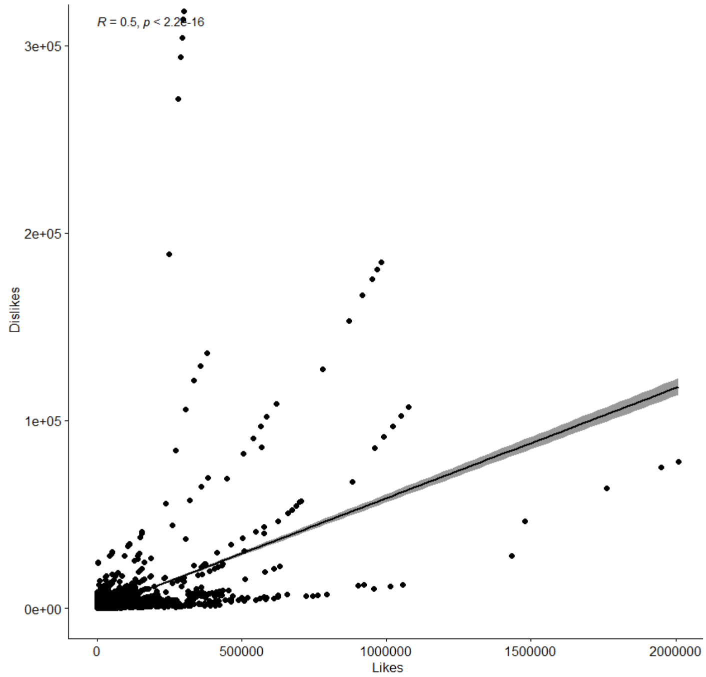
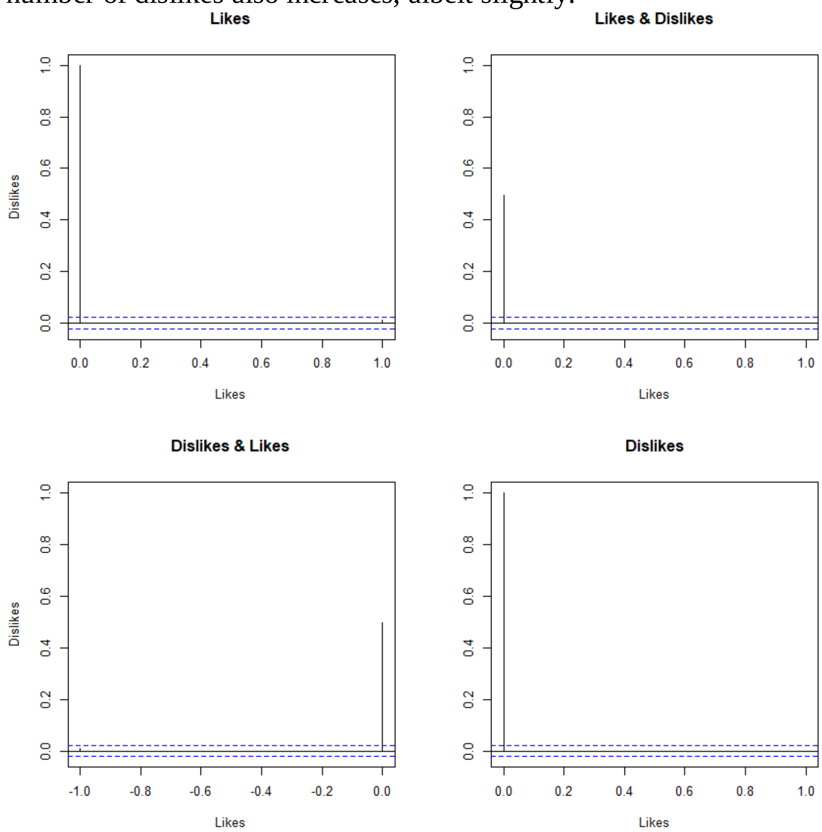


Figure 36: Correlation relation

Thanks to this correlation graph, we can observe that with a rapid increase in the number of likes, the number of dislikes also increases, albeit slightly.




```
      part1 part2 part3
part1  1.00  0.06  0.01
part2  0.06  1.00  0.02
part3  0.01  0.02  1.00

n= 2666


P
      part1  part2  part3
part1        0.0016 0.4423
part2 0.0016        0.4081
part3 0.4423 0.4081
```

Figure 37: Graphs of autocorrelation functions and Correlation matrix

From this visualization, it is possible to conclude that our board is stationary, and since the data on any interval are not equal to zero, their regularity follows. We can conclude that the attribute by which the matrix is built is quite homogeneous, which is logical, because this attribute is likes.

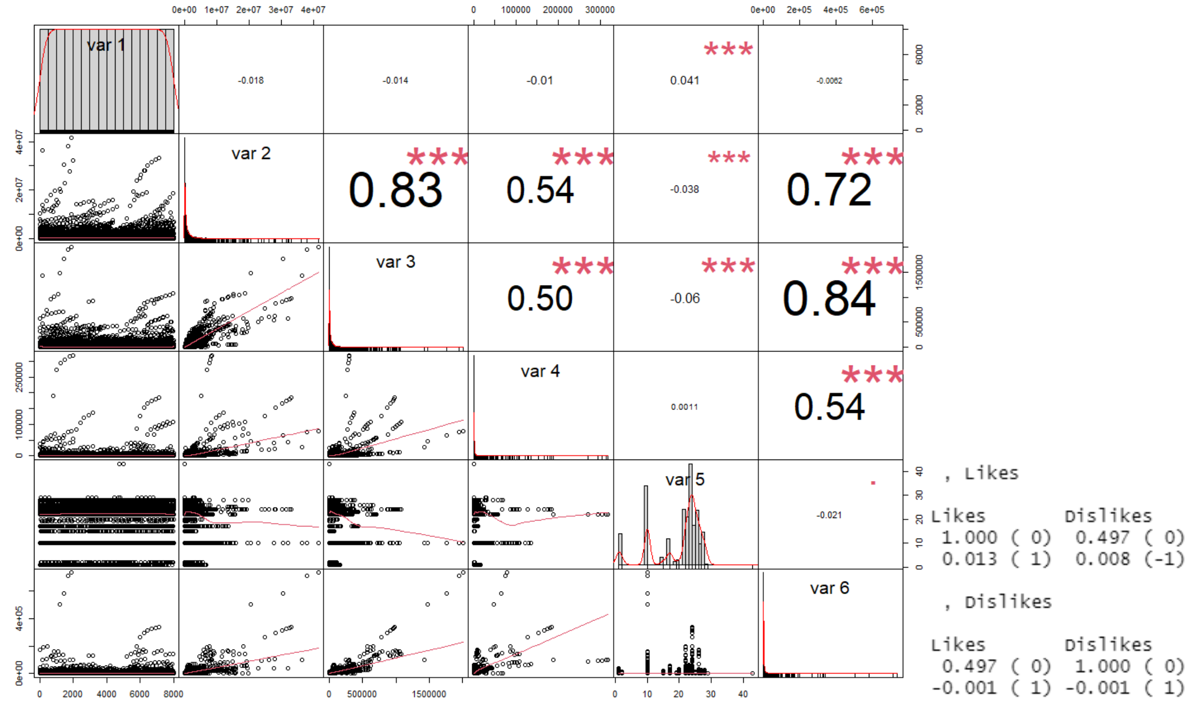


Figure 38: Multiple correlation coefficients and autocorrelation result

From this visualization, we can see that there are quite strong relationships between attributes, but there are also negative correlation coefficient results.

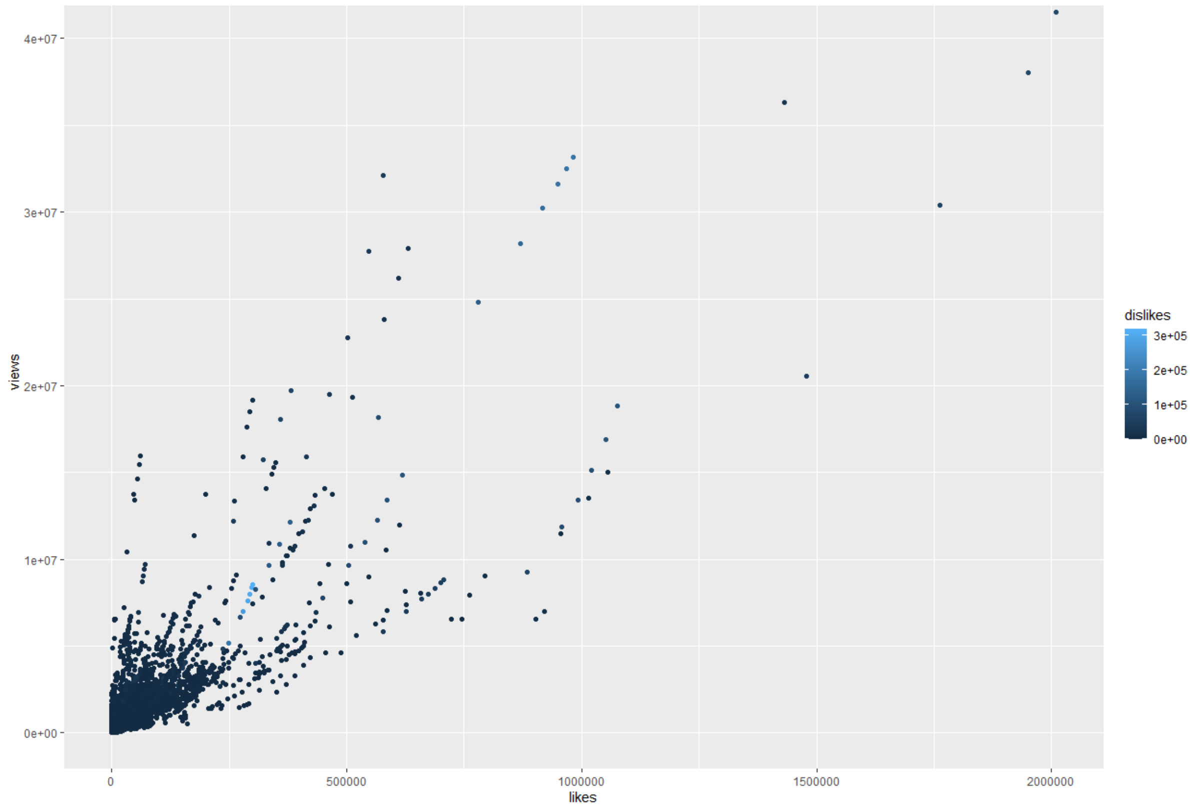


Figure 39: Graphic representation of cluster analysis

Again, as it was proven before, as the number of likes increases, the number of views increases. Although the clusters on the visualization are quite difficult to distinguish, it is possible to generally identify two clusters, namely, light blue spots - the relationship of a high number of views and likes, dark blue spots - a small number of these attributes.

```
K-means clustering with 3 clusters of sizes 53, 7471, 474

Cluster means:
       [,1]     [,2]
1 836660.49 16.71698
2  17288.13 20.32017
3 216245.50 18.99367

Clustering vector:
   [1] 3 3 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2
  [42] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
  [83] 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [124] 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3
 [165] 2 2 2 2 2 2 2 2 2 2 3 2 2 1 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 3 2 2 2 2 2 2 2 3 3
 [206] 2 2 3 2 2 1 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [247] 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [288] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [329] 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [370] 2 2 2 2 2 2 2 2 2 3 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 2 2 2 2 2 2 1 2
 [411] 2 2 2 2 2 2 3 2 2 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 3 2 2 2 2 2 2 2 2
 [452] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2
 [493] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [534] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [575] 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 3 2 2 2 2
 [616] 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [657] 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [698] 2 3 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2
 [739] 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [780] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2
 [821] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 3 2 2 2 2 2 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [862] 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2
 [903] 2 2 2 3 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 3 2 2 2 2 2 2 2 2 2 2 2
 [944] 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2
 [985] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [ reached getOption("max.print") -- omitted 6998 entries ]

Within cluster sum of squares by cluster:
[1] 6.100590e+12 4.067499e+12 4.598632e+12
 (between_SS / total_SS =  77.9 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
>
> table(cluster$cluster, dt$category_id)

       1    2   10   15   17   19   20   22   23   24   25   26   27   28   29   43
  1    0    0   27    0    0    0    0    7    0   16    0    3    0    0    0    0
  2  363  116 1102  116  395   48   81  784  635 1517  626  840  329  503   14    2
  3   15    0  123    0   15    0    1   91  120   68    0   27    5    9    0    0
```

Figure 40: Clustering matrix using the k-means method

According to this matrix, it is difficult to determine which cluster is built and according to which parameter, that is, there are no clear boundaries.

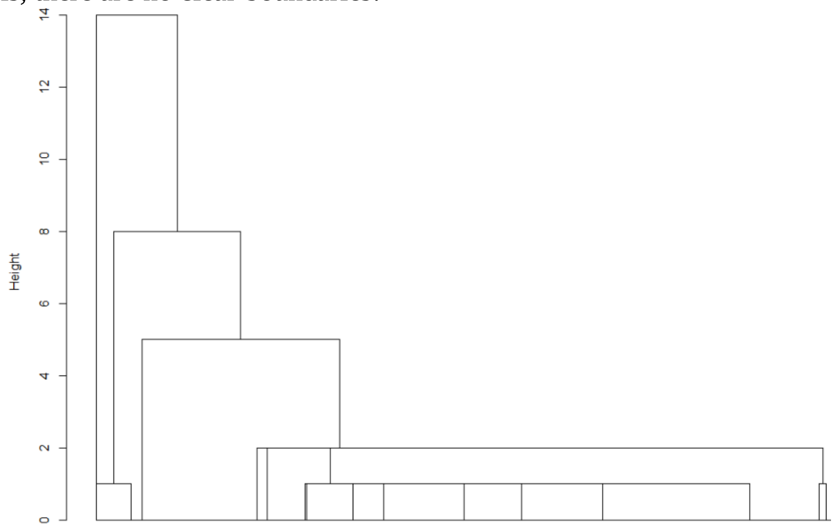


Figure 41: Dendrogram

The dendrogram was constructed without labels, in order to facilitate the understanding of the number of levels of clustering between categories of video content on the YouTube platform, sacrificing, unfortunately, the hierarchical relationship between objects.

6. Discussion

For a better analysis of the categories, let's find out the names of the categories that correspond to the identifiers: 1 - movies and cartoons, 10 - music, 15 - pets and animals, 17 - sports, 19 - travel, 20 - games, 22 - people and blogs, 23 - humor, 24 - entertainment, 25 - news and politics, 26 - style, 27 - education, 28 - science and technology, 29 - non-profit and activism.

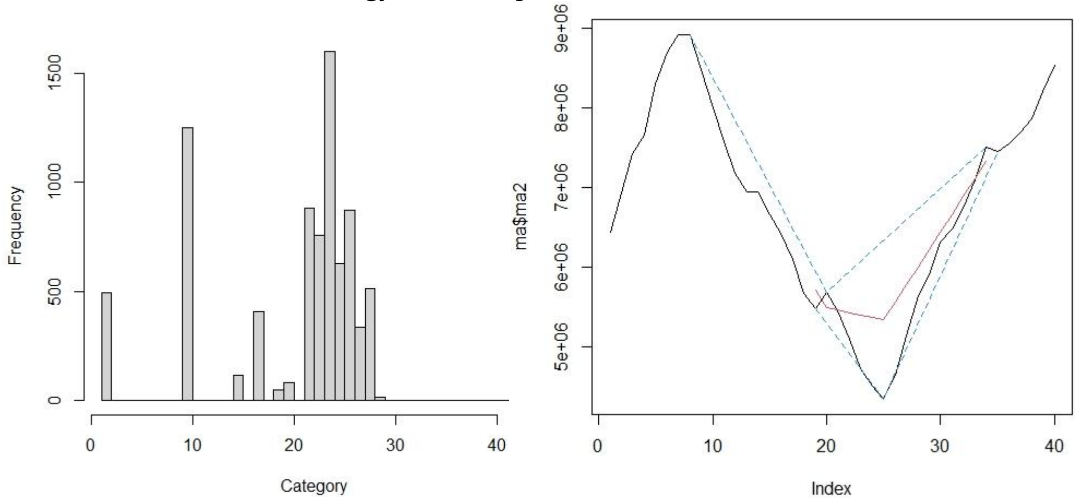


Figure 42: Histogram of category_id attribute data and visualization of turning points with simple smoothing

It can be seen from the histogram that the most popular categories are categories with identifiers 24, 10, 22, 28, that is, people are most interested in watching thoughtful videos or other people.

This graph clearly shows when we have a change in the viewing trend. A simple moving average got rid of the noise, but you have to take into account that the platform must be in trend.

A simple moving average is suitable for identifying trends in the past, which will help us predict the future with less error. However, for such a large and popular platform as YouTube, it is necessary to analyze the latest events. For this, they need methods that quickly respond to the latest data. When working on such methods, we used the weighted moving average smoothing method and exponential smoothing.

Another method is median smoothing with the size of the smoothing interval w = 3.

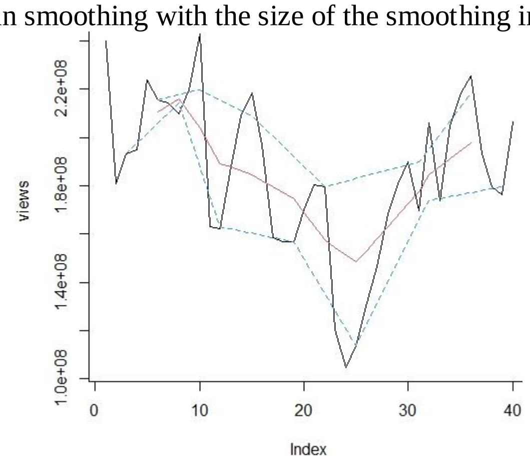


Figure 43: Visualization of turning points during median filtering

We can note that the general trends in the above graphs (Figs. 42-44) are practically identical, which makes both methods suitable for working with the selected dataset.

Thanks to the correlation relationship, we can observe that with a rapid increase in the number of likes, the number of dislikes also increases, albeit slightly. Analyzing other graphs, we can conclude that when the number of views increases, the number of likes increases, and therefore the number of dislikes increases.

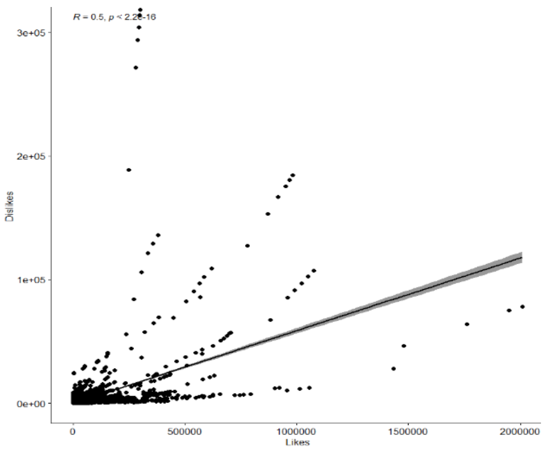


Figure 44: Correlation relation

Thanks to this correlation graph, we can observe that with a rapid increase in the number of likes, the number of dislikes also increases, albeit slightly.

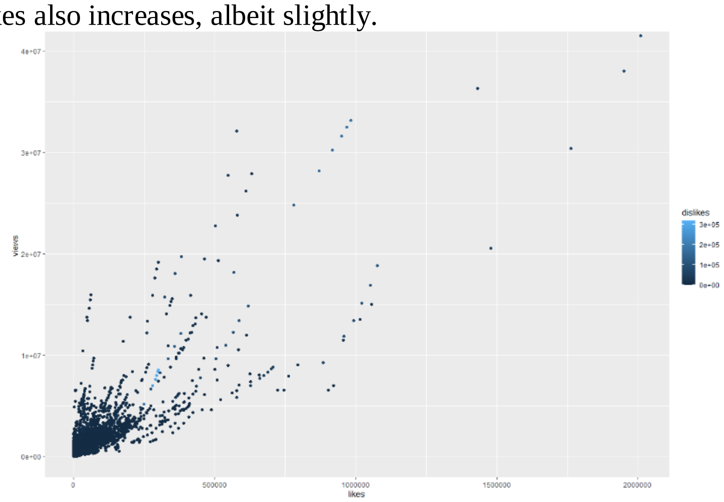


Figure 45: Graphic representation of cluster analysis

Again, as it was proven before, as the number of likes increases, the number of views increases. Although the clusters on the visualization are quite difficult to distinguish, it is possible to generally identify two clusters, namely, light blue spots - the relationship of a high number of views and likes, dark blue spots - a small number of these attributes.

7. Conclusions

A simple moving average is suitable for identifying trends in the past, which will help us predict the future with less error. However, for such a large and popular platform as YouTube, it is necessary to analyse the latest events. To do this, they need methods that quickly respond to the latest data. During the calculation and graphic work on such methods, we studied the weighted moving average smoothing method and exponential smoothing. Analysing this dataset, we learned that people like to watch videos from the categories "Music", "Entertainment" or "People and blogs" the most. These categories account for the largest number of likes and views. At the same time, the most dislikes fell on the "People and blogs" category. This can be explained by the fact that people often differ in their opinions and they simply do not agree with what was said in the video. It is also worth noting that the relationship between the number of likes, dislikes and the number of views was investigated. There is a direct relationship between them, so when one of these attributes increases, the others will also increase. It turned out to be an interesting fact that topics related to science and technology have recently become more and more popular. However, the difference between likes and dislikes is significantly in favour of likes, which means that people are mostly happy when they consume YouTube content.

8. References

[1] Most popular social networks worldwide as of 2023, ranked by number of monthly active users: https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/

[2] M. Mohsin. 10 YouTube Stats Every Marketer Should Know In 2022 [Infographic]. URL: https://www.oberlo.com/blog/youtube-statistics

[3] B. Dean. How Many People Use YouTube in 2023? [New Data]. URL: https://backlinko.com/youtube-users#youtube-statistics

[4] R. F. Alhujaili, W. M.Yafooz, (). Sentiment analysis for youtube videos with user comments. In 2021 International Conference on Artificial Intelligence and Smart Systems, 2021, pp. 814-820.

[5] How YouTube changed the world. URL: https://s.telegraph.co.uk/graphics/projects/youtube/

[6] S. Kumar, M. Yadava, P. P. Roy, Fusion of EEG response and sentiment analysis of products review to predict customer satisfaction. Information Fusion, 52 (2019) 41-52.

[7] V.V. Hnatushenko, P. I. Kogut, M. V. Uvarov, On Optimal 2-D Domain Segmentation Problem via Piecewise Smooth Approximation of Selective Target Mappings. Journal of Optimization, Differential Equations and Their Applications 27(2) (2019). 60–95. DOI: 10.15421/141908.

[8] Hnatushenko V., Kogut P., Uvarov M. On Satellite Image Segmentation via Piecewise Constant Approximation of Selective Smoothed Target Mapping, Applied Mathematic

[9] L. P. Morency, R. Mihalcea, P. Doshi, Towards multimodal sentiment analysis: Harvesting opinions from the web. In Proceedings of the 13th international conference on multimodal interfaces, 2011, pp. 169-176.

[10] V. Vysotska, O. Markiv, S. Teslia, Y. Romanova, I. Pihulechko, Correlation Analysis of Text Author Identification Results Based on N-Grams Frequency Distribution in Ukrainian Scientific and Technical Articles, CEUR Workshop Proceedings, Vol-3171 (2022) 277-314.

[11] N. Romanyshyn, Algorithm for Disclosing Artistic Concepts in the Correlation of Explicitness and Implicitness of Their Textual Manifestation, CEUR Workshop Proceedings 2870 (2021) 719-730.

[12] Y. Yusyn, T. Zabolotnia, Methods of Acceleration of Term Correlation Matrix Calculation in the Island Text Clustering Method, CEUR workshop proceedings, Vol-2604 (2020) 140-150.

[13] B. Rusyn, V. Ostap, O. Ostap, A correlation method for fingerprint image recognition using spectral features, in: Proceedings of the International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science, TCSET, 2002, pp. 219–220.

[14] S. Voloshyn, O. Markiv, V. Vysotska, I. Dyyak, L. Chyrun, V. Panasyuk, Emotion Recognition System Project of English Newspapers to Regional E-Business Adaptation, in: IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT), 2022, pp. 392-397, doi: 10.1109/CSIT56902.2022.10000527.

[15] N. Kholodna, V. Vysotska, S. Albota, A Machine Learning Model for Automatic Emotion Detection from Speech, CEUR Workshop Proceedings, Vol-2917 (2021) 699-713.

[16] M. Hryntus, M. Dilai, Translating emotion metaphors from English into Ukrainian: based on the parallel corpus of fiction, CEUR Workshop Proceedings, Vol-3171 (2022) 737-750.

[17] O. Bisikalo, V. Kovenko, I. Bogach, O. Chorna, Explaining Emotional Attitude Through the Task of Image-captioning, CEUR Workshop Proceedings, Vol-3171 (2022) 1056-1065.

[18] K. Smelyakov, O. Bohomolov, M. Kizitskyi, A. Chupryna, Identification of Modern Facial Emotion Recognition Models, CEUR Workshop Proceedings, Vol-3171 (2022) 1267-1281.

[19] D. Nazarenko, I. Afanasieva, N. Golian, V. Golian, Investigation of the Deep Learning Approaches to Classify Emotions in Texts, CEUR Workshop Proceedings, Vol-2870 (2021) 206-224.

[20] I. Bekhta, N. Hrytsiv, Computational Linguistics Tools in Mapping Emotional Dislocation of Translated Fiction, CEUR Workshop Proceedings, Vol-2870 (2021) 685-699.

[21] I. Spivak, S. Krepych, O. Fedorov, S. Spivak, Approach to Recognizing of Visualized Human Emotions for Marketing Decision Making Systems, CEUR Workshop Proceedings, Vol-2870 (2021) 1292-1301.

[22] P.C. Thoumelin, N. Grabar, Subjectivity in the medical discourse: On uncertainty and emotional markers, Revue des Nouvelles Technologies de l'Information, E.26 (2014) 455–466.

[23] N. Grabar, L.O. Dumonet, Automatic computing of global emotional polarity in French health forum messages, Lecture Notes in Computer Science 9105 (2015) 243–248.