Student: Nikola Marin
Student Number 89182009
Menthor: dr. ***Julie Ducasse***
UP FAMNIT
GitHub repositiry: https://github.com/unimer/Static-DataVis

Static Datavis

5th June 2019

OVERVIEW

This project is related IOTOK (https://iotok.eu/hr/#/epizoda/1) interactive series about latest inhabitants of island Briševo in Croatia. My idea was to put myself in position of YouTube content creator and answer some questions which will give me idea about audience and make base for further tactics for making and posting YouTube videos. Purpose of this project is to show the process creating of static data visualization. It includes analysis of visual primitives, visual variables, color palettes, data transformations and filtering that are used. The final goal is to show static graphs which will be easy observable and self explanatory.

DATASET

IOTOK series consists of 13 YouTube videos. The dataset used in this project is (I suppose) dataset from YouTube which contains all data relevant to all 13 videos. The part of the dataset is shown in picture bellow.

	user	timestamp	week	day	hour	logged	episode	name	watchingTime	duration
1	0f165b85-0670-e6ac-bf14-cf00655a	17/04/20 23:48	14	17/04/20	23:48:54	FALSE	12	Ep.12	210	917
2	1882942b-e429-56c3-71d6-1fe03230	17/04/18 23:27	14	17/04/18	23:27:31	FALSE	1	Ep.1	104	419
3	22389fe9-1ac5-62e4-0f04-cb35cd64	17/04/17 12:26	14	17/04/17	12:26:04	FALSE	1	Ep.1	19	419
4	269144fa-ef0e-c81e-52ff-df9a348a	17/04/17 2:26	14	17/04/17	02:26:18	FALSE	9	Ep.9a	34	73
5	29162f05-0a33-4ebb-3768-95d33323	17/04/23 12:36	14	17/04/23	12:36:44	FALSE	1	Ep.1	401	419
6	2e038001-8a20-a71f-3756-2bec0966	17/04/23 21:42	14	17/04/23	21:42:04	FALSE	10	Ep.10	32	587
7	3469f507-2f62-0e43-70e8-0c4bc6e3	17/04/23 13:18	14	17/04/23	13:18:02	FALSE	2	Ep.2	465	481
8	3469f507-2f62-0e43-70e8-0c4bc6e3	17/04/23 13:30	14	17/04/23	13:30:02	FALSE	1	Ep.1	24	419
9	3469f507-2f62-0e43-70e8-0c4bc6e3	17/04/23 14:43	14	17/04/23	14:43:51	FALSE	4	Ep.4	250	599
10	3469f507-2f62-0e43-70e8-0c4bc6e3	17/04/23 15:12	14	17/04/23	15:12:29	FALSE	7	Ep.7	134	339
11	3469f507-2f62-0e43-70e8-0c4bc6e3	17/04/23 15:30	14	17/04/23	15:30:42	FALSE	9	Ep.9a	61	73
12	3bc92395-636f-14c6-d1f0-fe0f4de7	17/04/18 16:48	14	17/04/18	16:48:20	FALSE	1	Ep.1	77	419
13	66a8d045-4389-3b4f-12a1-77c3d5ba	17/04/21 21:26	14	17/04/21	21:26:16	FALSE	13	Ep.13a	31	47
14	66a8d045-4389-3b4f-12a1-77c3d5ba	17/04/21 21:27	14	17/04/21	21:27:08	FALSE	13	Ep.13b	644	817

Dataset Fields Description

Name of Attribute	***Semantics of Attributes***	***Type of Attributes***	***Description***
user	key	nominal	If the user is logged in in this field will be *username*. If user is not logged in this field will contain ID number of the user.
timestamp	temporal	ordinal	Date and time when the user started to watch video. Format: dd/mm/yy hh/mm
week	temporal	Interval /ordinal	Number of weeks passed from the upload of the video.
day	temporal	Nominal	Just date (dd/mm/yy) when the user watched video.
hour	temporal	Ratio/Ordinal	Time (hh/mm/ss) when user started to watch video.
logged	value	Ordinal	If user was logged in the value in this field will be *true* and if user was not logged in the value will be *false*
episode	key	Ordinal	Number of the episode.
name	discrete	Nominal/ Ordinal	Full name of the episode.
watching time	temporal	Ratio	Watching time in seconds
duration	temporal	Ratio	Duration of the video in seconds

GOALS

The goals section is separated in two groups. First group contains visualisation design principles that have to be satisfacted. Second group contains list of questions I wanted to answer.

Visualizations should satisfy next visualization design principles:

1. Encode all the data relations intended and no other data relations - *(expressiveness)*
2. Get informations from visualization without any distractions - *(expressiveness)*
3. Use appropriate graphical language to make visualization appropriate to human visual system - *(effectiveness)*
4. Appropriate scales - *appropriate usage of object sizes and scales*
5. Avoid chart junk
6. Clear vision - *avoid data overlap*
7. Clear understanding - *make it readable on first look*
8. Appropriate annotation - *appropriate encoding of labels, define units ...*

Questions to be answered are:

1. What is the number of views per hour?
2. What is the number of views per episode?
3. What is difference in views between registered and unregistered users?
4. In which point of timeline most people stopped to watch video?

Creation of a Visualization

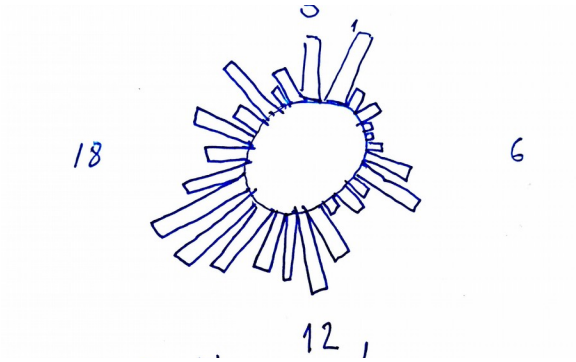
In this section I will show the process of creating visualizations for each question from above. While I was developing visualizations there were four common stages for each visualization. First, I analysed type of question and then I tried to find a way to present the answer to the user, on easy to read and emerging way. In the other words, I wanted observer to get a clue from one look at the visualization. Commonly I would search a graph gallery on ggplot website to find a graph and type of graph that fits the best with my imagination and to stay somehow conventional at the same time. On the second step I would make simple prototype on the paper in one color to get idea how it looks and how data have to be prepared. Then I would make data preparation, transformations and filtering to make it easier to present and finally the rendering of results.

1. Number of views per hour

What is the number of views per hour and in which period of the day videos are more likely to be watched? Having answer on those two questions content creator can get a clue about his audience and experiment with different uploading times or something else. For example from IOTOK dataset, as we will see later, most of the users are starting to watch videos from 6 a.m. until 2 a.m. next day. This can help to content creator to decide in which time he will upload video to YouTube or to change any other habit and way of creating.

Prototype

Since there we are talking here about time the best way to present the information is to make bar graph but rounded. The conventional name for this type of graph is actually circular bar graph. The bars will change their size and color depending on number of views. Every bar will be encoded with number of hour.



Data Preparation

Since the time of watching the video is written in dataset in format hh/mm/ss and we are interested just to get hour of watching we have to somehow parse the string hh/mm/ss and get

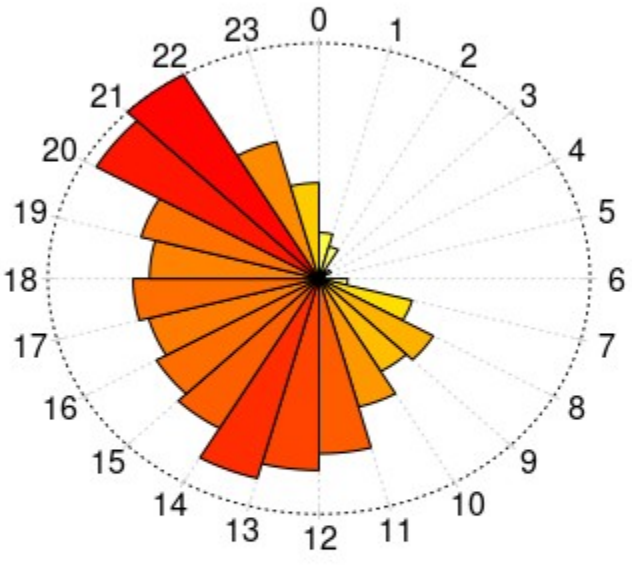
just hour number. To get only hour from timestamp I library *Lubridate* and *Lubridate* makes it easier to do the things R does with date-times and possible to do the things R does not. *Lubridate* library is relies on *tidyverse* library so I had to install *tidyverse* too. Using function *hour()* I was able to easily parse timestamp and to make new dataset which contains all columns like original dataset plus column *just_hour* which contains only hour number from timestamp. Then I filtered data grouping them by *hour* and then I summarized number of views for each hour for 14 weeks. So, for better understanding, there is an example: I summarized all views that happened between 1 a.m. and 2 a.m. for 14 weeks. Etc. Now each column contains *just_hour* value, that is actually number of hour and *nov(number of views)* which represent number of views in this hour. Now we are ready to use this data to render a graph.

	just_hour	nov
1	5	5
2	4	7
3	2	12
4	3	16
5	6	33
6	1	44
7	0	61
8	7	111
9	23	127
10	9	142
11	8	152
12	10	176
13	22	188

Number of views by hour

Rendering

When it comes to rendering I was searching ggplot gallery and I found pre-made plot for this purposes. It is not written in ggplot style of syntax but it uses ggplot functions. Colors are assigned to bars depending on number of views for that hour. We can say that color scale is sequential - when data values go from low to high, (the higher the value the higher the saturation). The result is in the image.



Choices and evaluation

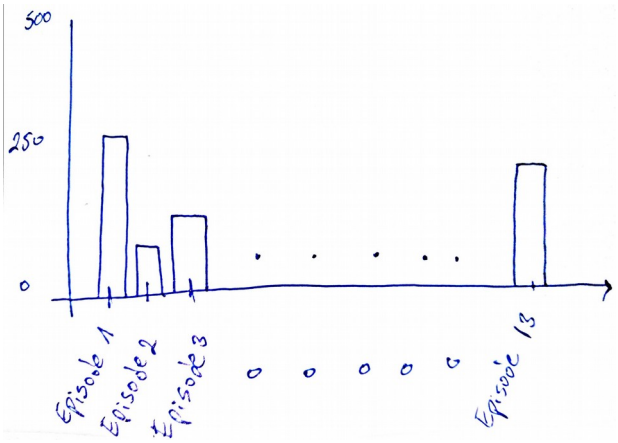
As I said earlier, the question that user might ask is related to time. It is very reasonable, and I think it is a good choice to visualize data with circular bar chart which looks like a watch and immediately tells to the user not just the answer to question, but one can conclude what was the question just looking at this answer. Also coloring choice (sequential) is appropriate for the application since the data is ordinal. There is no chart junk since all labels and visual variables are needed for easier understanding and scales are appropriate.

2. Number of views per episode

What is the number of views per episode? Which episode is the most watched one? Those two questions are not hard to show, but answer to them might mean a lot to content creator.

Prototype

Since those two questions are easy to answer and to understand, the choice I made is to make bar graph, where we will have *episodes* encoded on x-axis and the *number of views* encoded on the y-axis. Since data is nominal I was planning to assign one color to every episode bar. There is no need to have a legend because on the x-axis we will have number of episode under every bar.



Data preparation

Preparing dataset for this one is not complicated. We don't need to install libraries or to parse data. Since we have column *episodes* in the dataset where number of every episode is contained, we will group data by the episode number from *episodes* column and then we will summarize. So we get new dataset which contains number of episode and number of views(*nov)* for every episode.

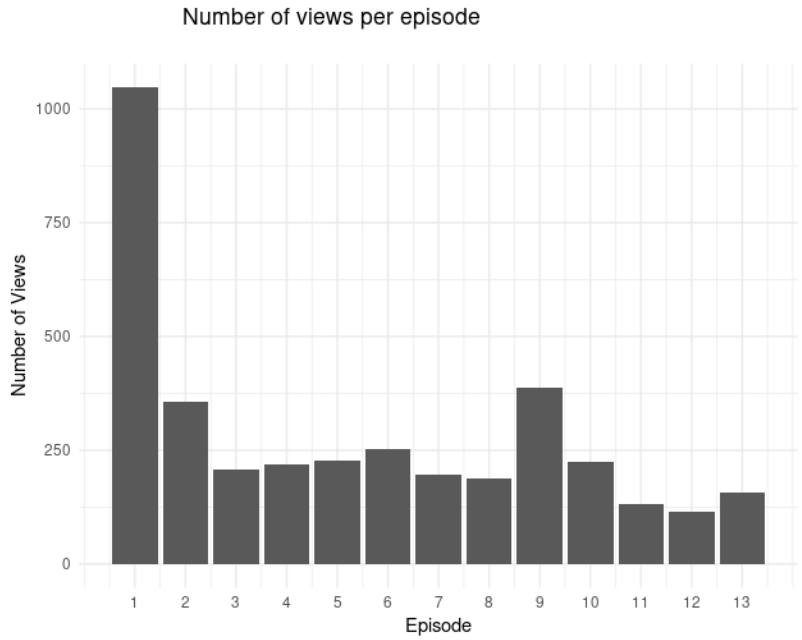
	episode	number_of_views
1	1	1047
2	2	357
3	3	208
4	4	219
5	5	228
6	6	253
7	7	196
8	8	188
9	9	386
10	10	224
11	11	131
12	12	116
13	13	158

Number of views per episode

Rendering

Now when it comes to rendering it is easy to make bar plot using ggplot library. We define which data to put on x-axis and which on y-axis. Then we define other parameters, like label of axis and color encoding for the graph. While implementing this particular graph I was experimenting with different color palettes and different labels on axis. First I made annotation on x-axis which were under 65 degrees angle under every bar, and the string of annotation was "Episode 1", "Episode 2", "Episode 3" etc. Then I found out that it is better to put just x-axis label with "Episode" and then annotate bars on x-axis with numbers of episode to avoid chart junk and unnecessary redundancy. I was also experimenting with color palettes because I wanted to enable easier distinguishing between episodes since color has selective characteristics but I conclude that the best option is to put the same color for all bars. The reasons for this are next. If I pick a color for every bar that represents different episode then whole graph will be too colorful because there are 13 different episodes and it becomes hard for the user to read it. The other reason was that there is absolutely no need to put different color of

bars since we have annotated bars on x-axis with numbers of episode, so the color is just one redundant visual variable. Also if we put different colors for every bar then we have to make legend for 13 colors which is enormous and it is hard to read and it is hard to make a good choice and to pick 13 different colors that will be easy for observer to distinguish between them. And in the end, data rendered on this visualization is so trivial and there is no need to try to help to observer by including more visual variables and primitives. So after this argumentation I decided to make minimalistic bar graph that has uniquely colored bars and it is easy observable. It is shown on the picture.



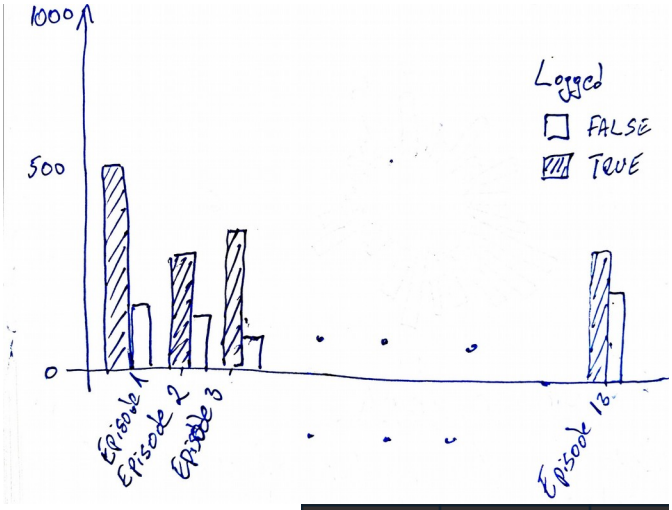
3. Registered vs Unregistered

The content creator might be wondering: What is the difference in views between registered and unregistered users? Is it important to send notifications about new video to viewers? Does YouTube algorithm influences users to stay in touch with content and keep watching?

Prototyping

Since we have column called *logged* in our IOTOK dataset, and fields of this column are containing *true* value if the viewer were logged in and *false* value if the viewer were unregistered. Using this data we can filter data and distinguish between two group of viewers. This means that we will need visualisation which shows two values for every episode. There is a lot of possible solutions but the most common one is bar graph with two data series. Using this

graph we will be able to put number of views from registered and number of views from unregistered viewers on one graph. This type of graph will allow us to read data from it easily since we can compare data from different group of viewers immediately without need to switch between two graph or views.



Data preparation

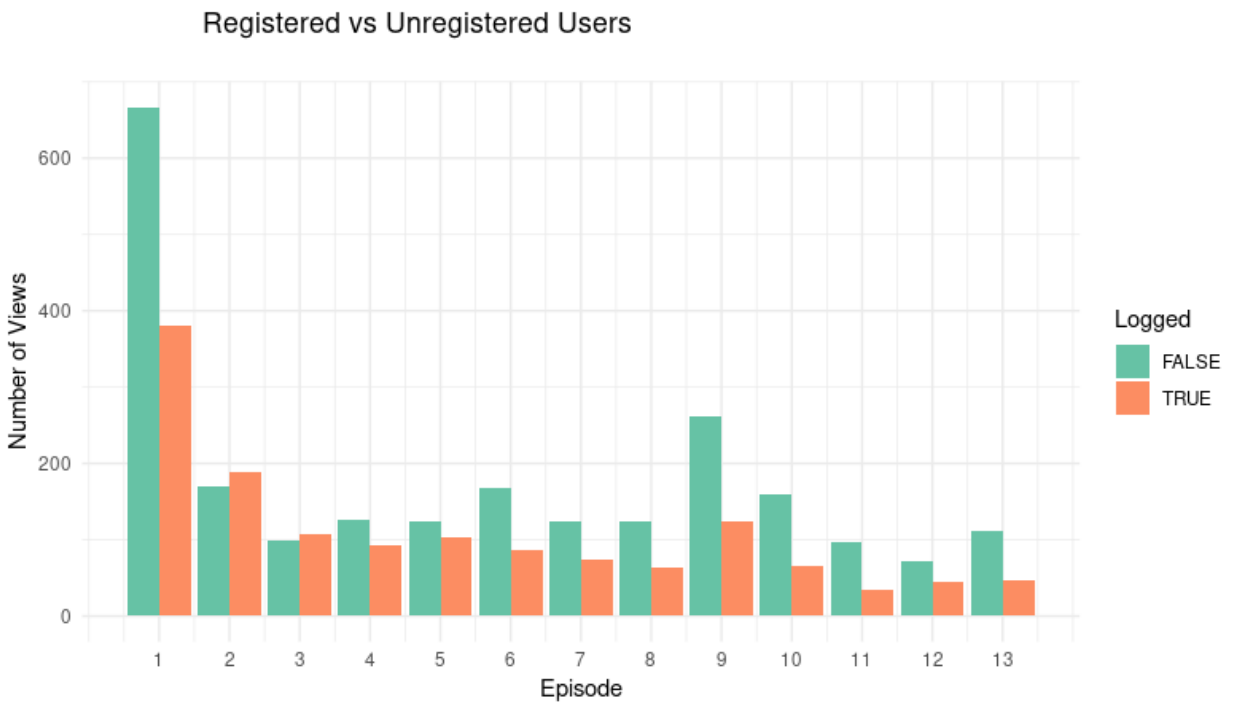
For this visualization we have to take data from the original dataset and group data by *logged* column. Then we will summarize registered users for every episode separately and unregistered users for every episode separately. So we will have new dataset that is contained of three columns (*logged(True/False), episode, number_of_views)*. We can separate this new dataset in two. The first half contains group of episodes which are viewed by unregistered users and the second half contains the group of episodes which are viewed by registered users. Every episode in each group has calculated number of views.

FALSE	7	123
FALSE	8	124
FALSE	9	261
FALSE	10	159
FALSE	11	97
FALSE	12	71
FALSE	13	112
TRUE	1	380
TRUE	2	188

Rendering

Using ggplot allows us to make this type of graph very easily. There is a function in which we put, which data we want to show on x-axis, which data we want to show on y-axis and by which attribute we want to distinguish between two bars that are showing value. The mapping is almost the same to the same to the mapping on graph before. This graph just has two bars for one episode, and since those bars represent difference between two group of viewers we have to include one more visual variable to make graph easily understandable. For this purpose we can use *hue*(color) or *texture* since both have selective and associative characteristics, and we need those characteristics because we have two groups of viewers. I choose to include hue visual variable. So the bars will have different color encoding, so the blue one represents registered viewers and the orange one represents unregistered viewers. The same color encoding of the bars is the same for every episode. There is also a legend which describes color encoding. This way observer can easily distinguish between two group of viewers and since it is repeated for every episode, observer can easily remember difference between two different bars. Labels under which was planned on the paper sketch were just making chart junk so I removed them, and put just episode number,

like in visualization above. To make visualization to look more modern I included minimalistic theme. The resulting graph is shown in the image below.



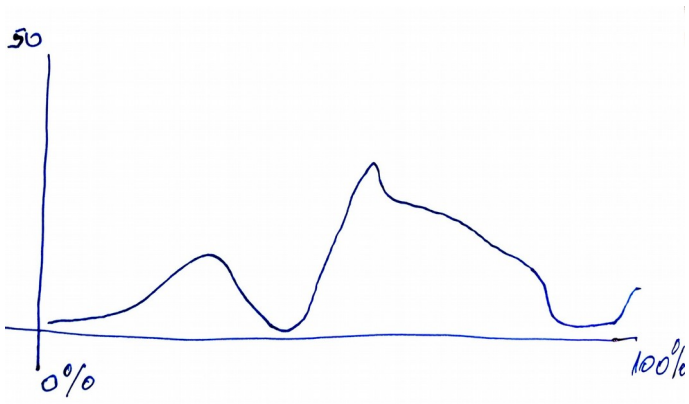
4. Where the most of the people stopped to watch video?

If I were content creator I would like to have information which shows me where the most people stopped to watch video. Then, for example I could check what was happening in the video at the point where most people decided to stop. On this way I would be more able to make content that people like to watch.

Prototype

Since I want to show those graphs in one view for all videos I will make data transformation to convert video duration in percentage. So on the x-axis we will have percentage which represents

duration of the video. On the y-axis we will have scale which will represent



number of users. Because we are presenting data related to time the most appropriate graph type to choose would be line graph.

Data preparation

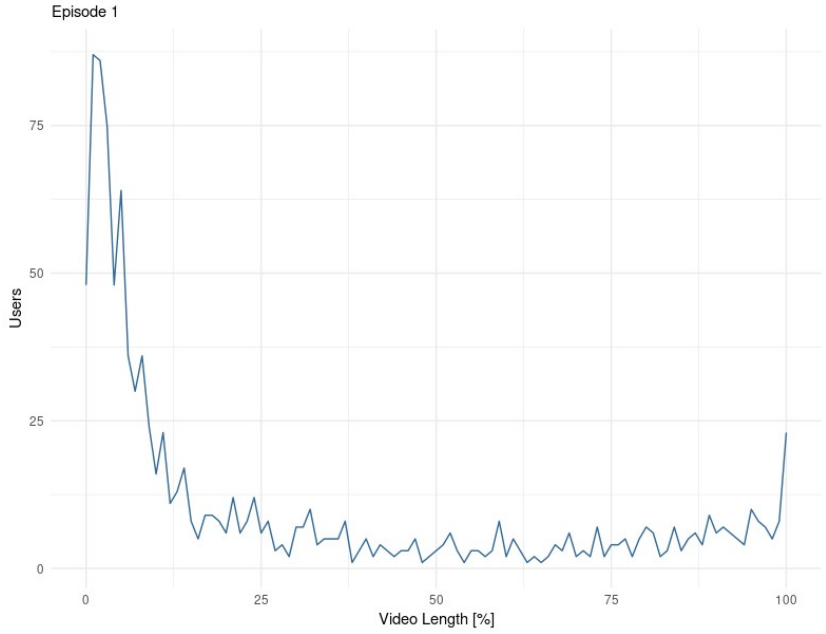
In dataset we have episode number, time in seconds that represents how much of video viewer watched and time in seconds which represents total length of the video. We have to make new data set that will contain watching time for every episode and for every viewer. Then we will mutate dataset and add one more column that will contain percentage of video where every viewer stopped to watch. ***NOTE:*** During testing I found out that in data set are some false values in episode duration section. But since there are 1% of wrong values, I made a quick fix by averaging the time duration of each episode.

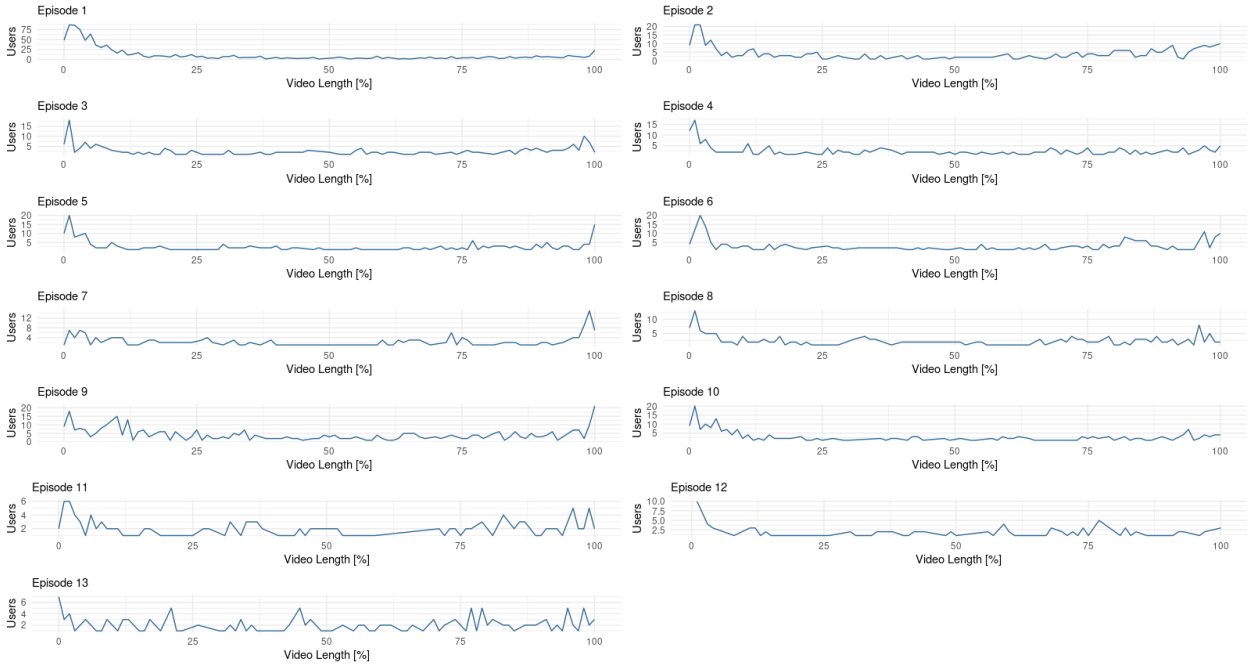
episode	percentage	viewers
1	0	48
1	1	87
1	2	86
1	3	75
1	4	48
1	5	64
1	6	36
1	7	30
1	8	36
1	9	24
1	10	16
1	11	23
1	12	11
1	13	13
1	14	17

Now using this new dataset we have to make one more dataset that will contain summarized number of users that stopped on every percentage for every episode. In other words, we have to count number of users that stopped at 10%, number of users that stopped at 50%, number of users that stopped at 51% etc. Final dataset is shown in the image.

Rendering

There is not much to say about rendering and visual mapping of this graph. This is classical line graph that has a line as a graphical primitive. Like the graphs above it uses minimalistic theme. What is interesting maybe is that I wanted to display graphs for every episode in one view. Then I used some techniques to add label above every graph which represents episode number. Also annotation on axis is minimal to avoid chart junk. For easier understanding I put units under the x-label which tells to observer that timeline is represented in percentage. The final presentation is shown on image.





Conclusion

My idea was to put myself in position of YouTube content creator and answer some questions which will give me idea about audience and make base for further tactics for making and posting YouTube videos. Purpose of this project was to show the process creating of static data visualization. It includes analysis of visual primitives, visual variables, color palettes, data

transformations and filtering that are used. The final goal was to show static graphs which will be easy observable and self explanatory.

In the beginning we posted two group of goals. First group contains visualisation design principles that have to be satisfacted. Second group contains list of questions I wanted to answer. We succeed to answer all of four questions counted at the beginning. This visualization uses basic graphs and all graphs are common and well known in data visualization. Those graphs are easy understandable. All of them have visual variables which allow user to make visual selection, association, ordering or quantization easy. I think that almost every question in this visualization is answered with respect to visualization design principles. Graphs are showing only data relations intended, distractions like chart junk, vision is clear it is readable on first look, visual variables are used in proper way, appropriate annotations and scaling is applied.

In the future this visualization might need interactive interface because it would enable us to manipulate data in the different way which would allow more emerging presentations. It would be nice if we hover over the line-graph which shows where users stop to watch and to see which exact minute and second of the video it is.