# *Analyzing Video Characteristics and Transcoding Time*

Corinne Medeiros
July 4, 2021
https://corinnemedeiros.github.io/

## Executive Summary

This study examines characteristics of videos from YouTube and aims to provide insight on video transcoding trends. The domain these data come from is film post-production, which entails all the steps that take place after shooting a movie and before consumer viewing. Specifically, this project represents the deliverables and distribution aspect of the post-production process; i.e. how a movie gets from the filmmakers' camera to your personal device. After a video has been edited and color corrected for example, it needs to be converted into the proper format for its intended platform, whether that be broadcast, theatrical, or online streaming.

Transcoding is the process of converting videos from one format to another (Cardwell, n.d.). Videos differ in frame rates, codecs, durations, size, and color management for instance. Transcoding to different formats involves a considerable amount of time and computer memory usage, so looking at which video metadata are responsible for the time and memory used can help give insight on where to optimize the process. In order to analyze the data, I explored the relationships between video characteristics and transcode time through a variety of visualizations representing variable distributions and correlations.

## Intro and Background of the Problem

Harnessing video data into manageable sizes for everyday use and compatibility is a constant challenge in the film industry, whether you're sending a video rough cut to a producer, putting together a feature film, or uploading content for streaming online. Achieving efficient workflows while keeping viewers content and uninterrupted requires specific video formats and codecs. Like encoding and decoding a message, codecs are different ways of compressing and decompressing video and audio. Codecs apply algorithms to reduce video file sizes for storage and transmission purposes, and then later decompress them for viewing (Ruether, 2019). Examples of common codecs include H.264, AV1, MPEG, and many more (see Appendix for a more complete list).

As the demand for increasingly higher quality content and faster loading times grows, so do video resolutions and file sizes as a result. A typical editing setup most likely cannot handle higher than 4K resolution without significant performance loss. Different devices, network bandwidths, and software require specific characteristics for video files in order to maintain functionality (Deneke, 2014). Video data has become harder and harder to manage. Thus, transcoding is a very necessary part of the post-production process and should be optimized.

The Online Video Characteristics and Transcoding Time Dataset from the UCI Machine Learning Repository contains two tsv files. The first file is 168,286 randomly sampled YouTube videos from 2015 along with their video characteristics including duration, bitrate, height, width, frame rate, codec, category, and url. The second file is 68,784 different instances of transcoding tests using samples of videos from the first file as input for the generated output columns. Additional attributes within the transcode dataset include output codec, output bitrate, output size, allocated memory, and total transcode time in seconds.

In my analysis I am exploring the following research questions:

Which video attributes contribute to longer transcode times?

Which video attributes contribute to higher memory usage?

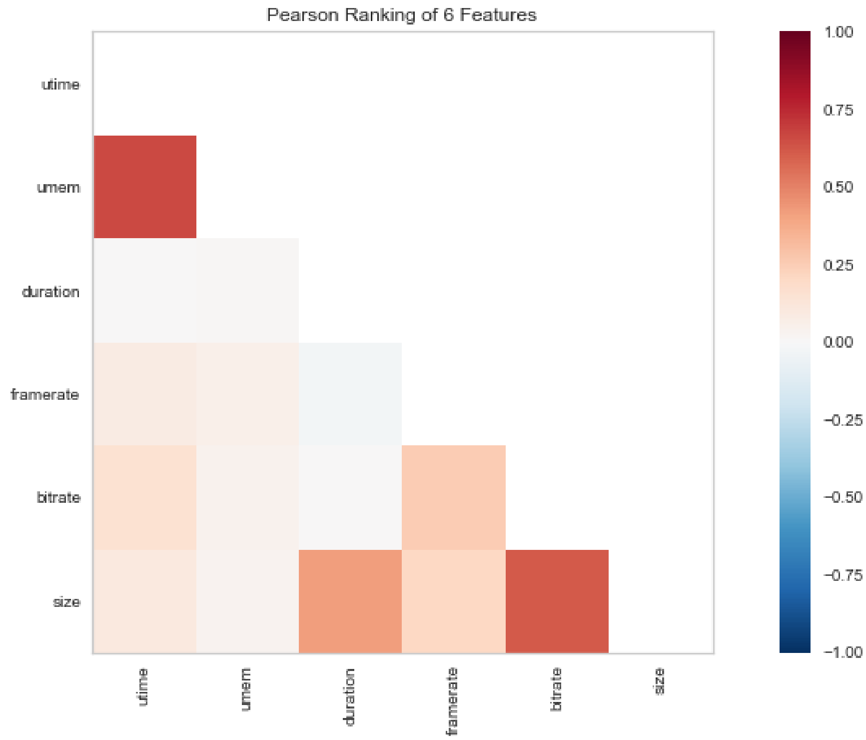
How can transcoding time be reduced?

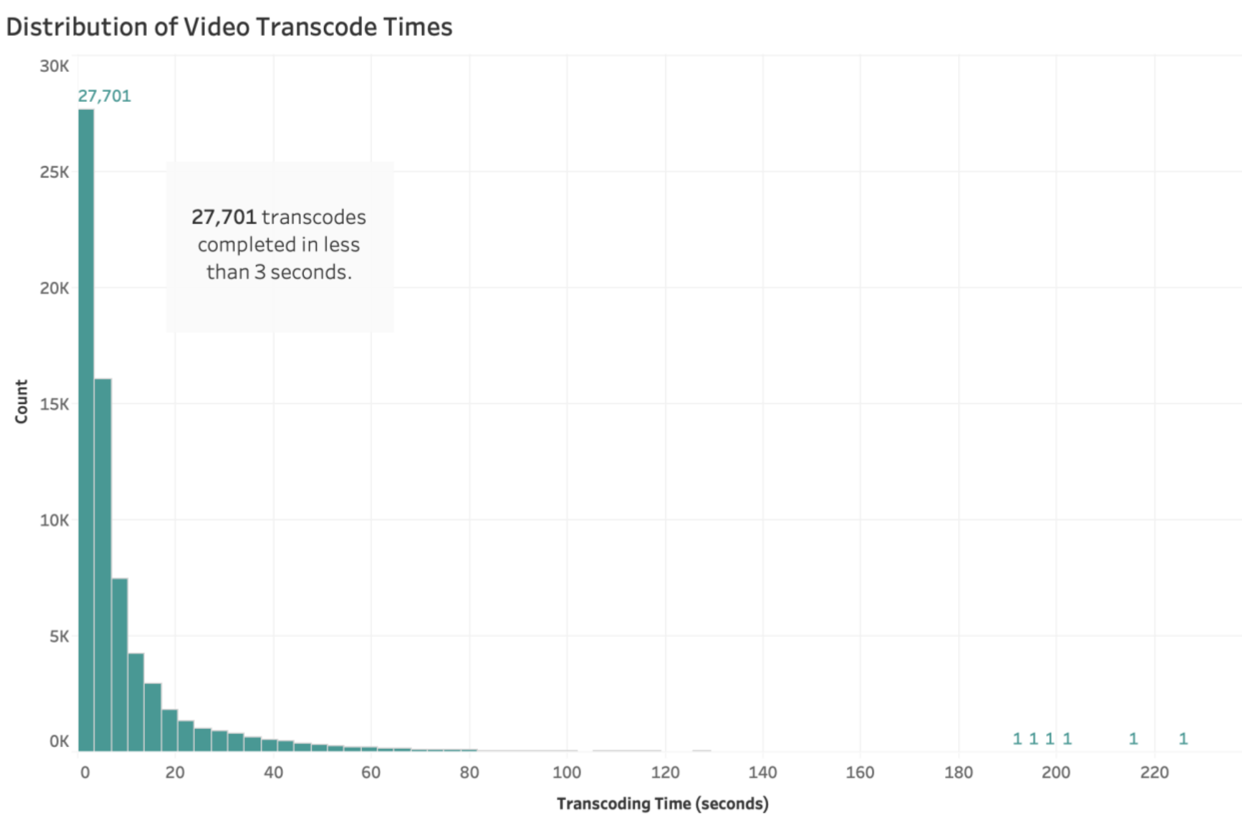
How can memory usage be optimized?

# Methods

For this analysis, I used Python in Jupyter Notebook to preview, clean, and explore the data with printed data summaries and rough visualizations. I got a sense of the patterns in the data through histograms and bar charts. I also used Tableau Prep Builder to experiment with manipulating and merging the raw data files. I cleaned the video id values by removing special characters and excess spaces, transforming the characters to the same case, and inspecting the data types. I wasn't able to merge the two datasets based on shared attributes however, so this helped me conclude that they are better suited for separate analyses.

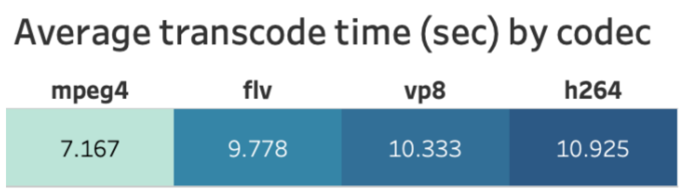
From there I went back into Jupyter Notebook to focus on just the transcodes dataset and look at the relationships between variables. I visualized the correlations between transcode time (utime), memory usage (umem), duration, framerate, bitrate, and file size using a Pearson ranking chart to determine which metadata have the most impact on memory allocation and transcode time. With the dataset summaries, exploratory visuals, and correlation insights from Python, I moved into Tableau to generate final visualizations. Below we can see the Pearson ranking chart and the histogram of transcoding time, which is right-skewed. The majority of transcodes in this dataset took less than 3 seconds.


Pearson Ranking of 6 Features
utime
umem
duration
framerate
bitrate
size
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00


Distribution of Video Transcode Times
27,701
27,701 transcodes completed in less than 3 seconds.
Count
Transcoding Time (seconds)

## Results

The main criteria I based my conclusions on are correlations. The higher the correlation, the more impact a variable has on transcoding time. Memory usage had the strongest relationship with transcode time, meaning that the more time a video took to transcode, the more memory was used. Additional experiments and details would be needed to explore other factors that might be contributing to memory usage. Bitrate had a slight positive correlation with transcode time, but the remaining variables were insignificant. Looking at the average transcode times by codecs produced the following chart, revealing that the mpeg4 codec has the lowest average transcode time while h264 files generally took longer on average.




Based on the information given, I had to make some assumptions about the data. The units of measurement for transcoding time and memory usage were not provided, so based on the number sizes and my research I assumed the time was in seconds and the memory usage was in kilobytes (KB). I also assumed the output parameters for transcoding were selected at random.

The final visualizations from Tableau are useful for interpreting the data in a more digestible way. Decisions about improving post-production processes can then be made more easily by non-technical stakeholders using the supporting visuals from this project. Benefits of analyzing the data this way would be in helping decision making for additional resources, bandwidth, and which formats are ideal to cut costs. For example, currently available machines in a company could be checked for specifications to determine if additional machines or memory could help reduce transcoding time. It would also allow for post-production teams to efficiently schedule automated transcodes based on required transcode time and memory allocation.

## Discussion and Conclusion

Some of the challenges in this analysis include access to data, relevant research, narrow scope, and changes in technology. It is difficult to obtain publicly available data within the post-production domain because a lot of the data come from protected intellectual property (IP). It is also hard to keep up with the quickly evolving technologies. Research and analysis become obsolete very quickly as compression software advances and higher quality video formats emerge. The dataset used for this project is from 2015, meaning the codecs and transcode times may be out of date. The v8 codec featured in the data for example is an older version of the A1 codec, so the newer iteration likely has better performance and would produce different results. The scope becomes specific to that time period, making it hard to compare the insights to more modern use cases and decision-making. Finding or generating newer datasets would be the most important next step to improve this project and take it to the next level.

Analyzing video metadata in relation to transcoding times and memory usage is important in understanding what resources are needed to make workflows more efficient. In the film industry, the need for optimal time estimations and video quality in a world dominated by streaming services is crucial to meet the needs of consumers expecting instantaneous and high-quality visual content. It is also necessary for production companies experiencing this massive growth in produced content to be aware of the best methods for cutting costs and time by purchasing the right storage solutions and software. Visualizing the relationships between these specifics will provide a framework to analyze more current data and compression types.

## References

Agrawal, P., Zabrovskiy, A., Ilangovan, A. et al. (2020). FastTTPS: Fast approach for video transcoding time prediction and scheduling for HTTP adaptive streaming videos. Cluster Comput. https://doi.org/10.1007/s10586-020-03207-x

Cardwell, S. (n.d.). Understanding codecs, proxies, and transcoding. Emerson College Post Production Labs. Retrieved from https://postproduction.emerson.edu/hc/en-us/articles/225933327-Understanding-Codecs-Proxies-and-Transcoding

Deneke, T. (2015, May 19). Online Video Characteristics and Transcoding Time Dataset. UCI Machine Learning Repository. [https://archive.ics.uci.edu/ml/datasets/Online+Video+Characteristics+and+Transcoding+Time+Dataset]. Irvine, CA: University of California, School of Information and Computer Science.

Deneke, T., Haile, H., Lafond, S. and Lilius, J. (2014). Video transcoding time prediction for proactive load balancing. 2014 IEEE International Conference on Multimedia and Expo (ICME), pp. 1-6, doi: 10.1109/ICME.2014.6890256. Retrieved from https://ieeexplore.ieee.org/abstract/document/6890256

FFmpeg developers. (2021, June 13). ffmpeg Documentation. FFmpeg. Retrieved from https://ffmpeg.org/ffmpeg.html

Kesavaraja, D., & Shenbagavalli, A. (2018). Framework for fast and efficient cloud video transcoding system using intelligent splitter and Hadoop MapReduce. Wireless Personal Communications, 102(3), 2117–2132. Retrieved from https://doi-org.ezproxy.bellevue.edu/10.1007/s11277-018-5501-3

Li, X., Salehi, M., Joshi, Y., Darwich, M., Landreneau, B., & Bayoumi, M. (2019). Performance Analysis and Modeling of Video Transcoding Using Heterogeneous Cloud Services. IEEE Transactions on Parallel and Distributed Systems, 30, 910-922. Retrieved from http://arxiv.org/pdf/1809.06529

Maayan, G. D. (2020, December 22). Video optimization: Traditional vs machine learning methods. Towards Data Science. Retrieved from https://towardsdatascience.com/video-optimization-traditional-vs-machine-learning-methods-d910c244a804

Mozilla. (2021). Web video codec guide. MDN Web Docs. Retrieved from https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Video_codecs

Ruether, T. (2019, October 16). Video codecs and encoding: Everything you should know (update). Wowza Media Systems. Retrieved from https://www.wowza.com/blog/video-codecs-encoding

Sagar, R. (2021, April 29). Chips that power your vlogs: the secret behind YouTube's uninterrupted service. Analytics India Magazine. Retrieved from https://analyticsindiamag.com/chips-that-power-your-vlogs-the-secret-behind-youtubes-uninterrupted-service/

Strickland, JR. (2020, July 13). How and why transcoding helps the editing process. Videomaker. Retrieved from https://www.videomaker.com/how-to/editing/how-and-why-transcoding-helps-the-editing-process/

Walton, S. (2012, June 14). Intel Core i7-3720QM: Mobile Ivy Bridge review. TechSpot. Retrieved from https://www.techspot.com/review/535-intel-core-i7-3720qm/

Wilbert, M. (2021, March 31). Streaming codecs for video and audio: What broadcasters need to know in 2021. Dacast. Retrieved from https://www.dacast.com/blog/codec-basics-for-online-video-audio-and-live-streaming/

**Appendix**

Mozilla (2021) provides this list of common codecs and their supported file formats:

| Codec name (short) | Full codec name | Container support |
|---|---|---|
| **AV1** | AOMedia Video 1 | MP4, WebM |
| **AVC (H.264)** | Advanced Video Coding | 3GP, MP4 |
| **H.263** | H.263 Video | 3GP |
| **HEVC (H.265)** | High Efficiency Video Coding | MP4 |
| **MP4V-ES** | MPEG-4 Video Elemental Stream | 3GP, MP4 |
| **MPEG-1** | MPEG-1 Part 2 Visual | MPEG, QuickTime |
| **MPEG-2** | MPEG-2 Part 2 Visual | MP4, MPEG, QuickTime |
| **Theora** | Theora | Ogg |
| **VP8** | Video Processor 8 | 3GP, Ogg, WebM |
| **VP9** | Video Processor 9 | MP4, Ogg, WebM |

## Questions

1. What is the best way to reduce costs using this information?
2. How can we be sure we are using the most efficient codecs?
3. Will these insights help improve load balancing between machines?
4. What potential do these data provide in terms of future predictions?
5. How can the data be improved?
6. How can this analysis be tailored to specific post-production workflows?
7. Can we implement this analysis on newer data?
8. What further visualizations and details can you provide?
9. How can we ensure optimal efficiency with transcoding times?
10. What other machine learning potential exists in this dataset?