# Cat Comedy – Predicting Feline Funniness with Bayesian Hierarchical Models

# Introduction

Is it possible to generate a realistic comedy routine about cats using statistics an AI?

Many types of art have structure – music, movies, stories, dance, and yes – even comedy routines. Like many kinds of data, these art forms are often structured in some sort of hierarchical way. For example, a comedy routine by one comic may have certain intervals between each joke. Another comic may have a routine on a different topic, over a different duration, with different intervals between their jokes. Over many comedy routines, with many comics, there are many variations.

To understand the structure of comedy routines, we will attempt to recreate the essence of comedy timing by creating a model. We will use this model to create a new comedy routine. As a final proof, we will plugin a new comedy sketch into the new routine, record it, and hopefully laugh.

*Note, I am not a comedian. I invented data driven comedy to learn more about statistics and not bore myself to death with another flippin’ coin problem or Kaggle dataset. If I get a Netflix special out of it someday – bonus.*

# Summary of Findings

* Comedians with routines about cats tell between 5-6 jokes per minute
* The mean duration between cat jokes is slightly more than 10 seconds
* The minimum interval between jokes is around 3 seconds
* The maximum time between jokes rarely exceeds 40 seconds
* Bayesian hierarchical models do a good job of predicting comedy routines

# Methodology

We explore the time of laughter in comedy routines using a Bayesian hierarchical model. That’s a statistical approach that looks at the data from different angles. We analyze ten routines looking at the intervals between laughter. Our model will learn the intervals across all routines, while also considering each routine individually. This hierarchical model allows us to see general trends, and still capture the nuances of individual routines.

We then use this model to generate predictions about what the laughter timing might be in a new, hypothetical routine. This methodology allows us to understand the common intervals and timing in comedy. We can use this information to gain insights into one specific aspect of what makes comedy funny.

# Data Set

The dataset was extracted from professional comedy routines about cats, found on YouTube. Ten routines were chosen, with varying durations from a minimum of 52 seconds, to a maximum of 7.5 minutes. The specific video titles are not relevant.

The [laughter\_detection](https://github.com/jrgillick/laughter-detection) library is a machine learning library that was used to detect laughter. In the analysis, laughter indicates the occurrence of a joke at a specific time stamp in the routine. If there is no audience laughter there is no punchline occurring or nothing funny happening/being said.

Each audio routine was processed to find where laughter occurs. These values are recorded as a time series, such as:

['00:05', '00:13', '00:21', '00:35', '00:48', '01:02']

This example is for the shortest routine analyzed, which was found to have 6 jokes. The longest routine had 45 jokes.

# Model Implementation

Our comedy timing model has multiple layers, with each layer revealing more specific information:

**The Big Picture:** At the outermost layer, we have general patterns that apply to all comedians. Think of this as the overall trends in how often jokes are told and how long comedians pause between jokes. We can't see these trends directly, but they influence everyone in the comedy world.

**The Individual Comedian:** Every performer has their own unique style. Some tell long stories; some might tell one-liners. We can't directly measure a comedian's internal ‘joke clock,’ but we can guess at it based on their performances.

**The Actual Performance:** We have the real-world data - the exact timing of laughs in each routine. This is what we see and hear when we watch a comedy show.

By looking at all these layers together, our model can:

* Spot overall trends in comedy timing.
* Figure out each comedian's personal style.
* Use all this information to predict how a new routine might be timed

The model consists of several key components:

**Global parameters (hyperparameters)**

These represent the overall characteristics across all routines. We start with an informed prior, based on other comedy routines that were analyzed in other projects.[[1]](#endnote-1) Our initial belief is the average interval between jokes is around 10 seconds.

The distribution of the global intervals is a truncated normal distribution. It’s truncated because logically – the interval wouldn’t be less than 3 seconds or more than 25 seconds. The shorter interval of 3 seconds isn’t long enough to tell a joke, while the longer interval of 25 seconds leads to a boring routine (not enough laughter).

The standard deviation for the intervals is HalfCauchy, which is appropriate when the standard deviation should be positive (there are no negative intervals).

**Routine-specific parameters**

The routine specific mean is modeled as a Gamma distribution, used for non-negative values.

The hierarchical structure allows the global parameters to inform the routine-specific parameters.

# Analysis

After fitting the model to the comedy routine dataset, some interesting insights and patterns emerge.

*Mean interval*: Figure X shows the posterior predictive mean interval between jokes for all routines. The mean interval is predicted to be 12 (seconds) (94% CI of 8.5-15.4). The accompanying notebook has the posterior predictive means for each individual routine, which all show normal distributions.

A graph of a normal distribution

Description automatically generated

**Figure X:** Mean interval across all routines

Figure X is a summary table of the posterior distribution for each parameter in the model. The *r\_hat* value of all the parameters is 1.0, indicating the sampling converged and we are confident in the estimates.

A table with numbers and a number of objects

Description automatically generated with medium confidence

**Figure X:** Global joke rate for all routines based on the dataset

To verify our model can do a good job of predicting, we run *posterior predictive checks*. These checks make simulated runs and compare the mean of a predictive output to the input data. Figure X shows four of ten posterior predictive checks illustrating how the model fit the data (the remainder were not included for brevity). The orange line (which is what was predicted) roughly follows the shape of the observed data from the dataset. The blue lines represent the data samples that were predicted from the model.

The observed data (our input data) appears to be very squiggly. This is expected due to the small number of data points / wide variation of data from each input routine.

A group of graphs with lines

Description automatically generated with medium confidence

**Figure X:** Posterior predictive plots to check how the model fits the data

# Predictions

After fitting the hierarchical Bayesian model to the data, we generate posterior predictive *samples*. This helps us assess the models’ performance and get insights into the predictions. We can compare the model’s predictions with the observed data to understand any uncertainty in our estimates.

Figure X (while a bit busy looking) shows the observed intervals (dotted lines) compared to the predicted intervals (solid lines) for each routine. Overall, the model captured the mean intervals across different routines, as shown by the common peaks and narrowness of the curves. There was some variability in the tails, and in a couple of the peaks for the observed data that is not present in the predicted data. This reflects some uncertainty in the estimates.

For most routines, the observed rate was within the central range of the predicted rates. This indicates a good model fit.

**A graph of a number of data

Description automatically generated with medium confidence**

**Figure X:** Combined plot of all predicted intervals compared to observed data in every routine.

Figure X highlights how well the model predicts the mean of each routine compared to the observed data. The red line would be a perfect 1:1 prediction. Most points are close to the line, suggesting the model does a good job of capturing most of the variation. The sketches above the line indicate the model underestimated their mean intervals. Sketch #9 seems to be an outlier; with some properties the model couldn’t capture well.

A graph with numbers and letters

Description automatically generated with medium confidence

**Figure X:** Observed mean compared to the predicted means for each routine.

Figure X is an isolated predictive posterior plot for Routine 1. It provides a comparison between the observed interval distribution (blue) and the predicted interval distribution (red). The remaining routine plots are available in the notebook for reference.

The model captured the overall shape of the distribution. The highest densities were for the 10-second interval, which is consistent with our previous analysis.

The predicted distribution a slightly higher density for the predicted mean. This can be seen by the taller peak. This indicates the model has less variability than was observed in this specific routine. This can indicate uncertainty, or that information from other routines was incorporated into the prediction.

Overall, the model captures the essence of this routine. It reveals some limitations of precision that may be harder to capture completely.

A graph of a normal distribution

Description automatically generated

**Figure X:** Interval distribution of the predicted vs. observed dataset for Routine 1.

# New Hypothetical Routine

Let's imagine we're planning a short comedy set for an up-and-coming comedian (me!) at a local open mic night. Based on our model, we'll generate a hypothetical routine and make predictions about its structure.

The only input to the prediction is the desired total length of the time series. Since we want the sample text to fit the predicted intervals of the time series, we choose a duration of 173 seconds. The duration is derived from a word-per-minute speaking rate and the word count of the text.

**Interval timing:** The model predicts a mean interval between jokes of ~10 seconds. That’s ~6 jokes per minute.

Let's verify these predictions by analyzing the actual generated routine:

*The notebook may have a different predictive routine, because it changes every time it’s run. Which is a plus, because the code can be used to generate infinite routines of any duration*.

From our posterior distribution the mean interval is about 10 seconds. Keep in mind the analysis was done on routines that had pauses, breaks, build-ups, etc. Our starting sketch is straight text with no pacing or pausing.

With a final estimated duration of about 02:45 (173 seconds rounded), we can create a new hypothetical routine (Figure X).

*We keep generating routines until one meets our personal preference of starting with a short joke interval (3 seconds vs. 20 seconds). 20-seconds until the first joke is certainly within the model’s predictions. However, that’s a long build to begin with. The shorter limit could be programmed into the code, but it wouldn’t be representative of the dataset.*

A screenshot of a cell phone

Description automatically generated

**Figure X:** Predicted routine showing the timestamps where punchlines should occur, and the intervals between jokes.

Figure X shows line plots representing the kernel density estimation of the interval distributions for all routines, including the hypothetical routine. We observe the peaks of all the routines hover around a similar mean. The spread varies, indicating some routines have longer vs. shorter intervals, but on average they are the same.

A graph of different colored lines

Description automatically generated

**Figure X:** Sampled posterior distribution of the mean intervals for the hypothetical model mapped against the original dataset.

Figure X shows the cumulative distribution of intervals for all routines. This plot is valuable for validating the hypothetical routine against the original data. The hypothetical routine has a similar overall pattern compared to the originals.

A graph of a number of bars

Description automatically generated with medium confidence

**Figure X:** Cumulative distribution of the intervals of the hypothetical routine mapped against observed routines from the original dataset.

# Conclusions

The hierarchical Bayesian analysis of comedy timing provided valuable insights into the structure and joke delivery of comedy routines. By looking at individual routines and performances as a whole, we come up with a new view into if data driven comedy exists. An art form often seen as subjective, is shown to have mathematical foundations.

The model provides a good general framework for comedy timing, and the general distribution of intervals between jokes. We get a solid basis for pacing from this. The hierarchical model captures both global trends and routine-specific variations. From this we uncover general principles and unique individual styles.

There was a good predictive ability of the model. This is what we were after – can we discover things that work well in comedy, in a repeatable fashion. Yes – there are things that can be both quantified and predicted.

**Areas for improvement**

Given there were edge cases (Routine 9) additional refinement could be done to handle edge cases. Sometimes the model’s predictions were smoother than the observed data. This indicates the model may be missing out on some of the more nuanced timing intervals and joke organization by individual comedians.

The laughter detection the dataset is based on doesn’t always capture all laughter. Also, audiences don’t always laugh at every joke, or don’t laugh enough to trigger the detector. From a simplicity standpoint, it beats listening to hours of audio and manually writing down time series.

Future models might benefit from:

* Adding the routine length as a factor
* Developing separate models for routines of significantly different lengths
* Grouping jokes into short/medium/long intervals to explore the distributions within routines
* Dog jokes

The model has limits, but it provides a decent framework for data driven comedy.

And let’s not forget the most important piece 🐱🐱🐱 – [an actual cat comedy routine applied to this model](https://youtu.be/qfJLRpvajRA?si=2J-zUQpeussQ2OfQ).

# Appendix A: Applying the Hypothetical Routine

It’s outside the scope of the analysis, but someone might wonder – what can we do with this time series output? GPTs can be used to ‘fill in the blanks’. The blanks being – the time series. We can take an existing piece of humorous content and turn it into comedy gold.

This prompt was used to populate a new routine:

You are a comedian working on your timing of a new routine. Use the following timeline that was successful in the past. Update the content of the routine to fit this timeline. Consider adding pauses where appropriate: {hypothetical time series} {rough comedy sketch}

The LLM model will produce time markers and break up the input into appropriate sections. This output is included as a table at the end of the notebook. Does the result line-up perfectly with the predicted timeline? No, but it’s surprisingly close.

To get some additional accuracy, a second prompt can be applied to refine the output:

If I speak at a rate of 1,000 words every seven minutes at a normal pace, and the provided text was 414 words, are there any adjustments to make to improve the jokes.

The LLM gives several suggestions to refine the routine. They happen to be terrible additions (sometimes the LLM output is very funny, but not this time).

The combination of the LLM and hierarchical model output can significantly reduce the time to refine a routine. The LLM can also supplement and enhance existing content.

Note: The recorded routine came out at 133 seconds, not 173. That was due to two oversights. First, the spoken words per minute to generate the duration was too low. I talk way faster with this type of content compared to doing a voice over for a course or teaching, which is where the incorrect spoken word count came from. Second, there was no audience, so there is no need to pause. I’d revise my speaking rate next time which should fit the content to the timeline better.

# Appendix B: References

1. [Exploding PowerPoints: Data Driven Comedy](https://medium.com/@scottalanturner/data-driven-comedy-1ab5018e34da) (Medium Article, Subscription Required) [↑](#endnote-ref-1)