Bayesian Regression: Learning the Shape of a Pandemic

Minerva University

CS146 - Computational Methods for Bayesian Statistics

Prof. Scheffler

November 1, 2025

Bayesian Regression: Learning the Shape of a Pandemic

Introduction

The goal of this project was to find a reliable statistical model that accurately describes the trend of confirmed COVID-19 cases in Buenos Aires (CABA) during 2024. A good model can help us understand the pandemic's trajectory and separate the real and underlying trend from simple day-to-day noise. Case counts are a timely and relevant metric for Buenos Aires. Understanding the trend of cases over 2024 can provide insights into how the pandemic evolved locally (e.g., whether there were waves or a steady decline).

Data Transformation

We started with the official dataset of all COVID-19 case reports from the CABA open data portal (*Buenos Aires Data | Casos COVID-19*, n.d.). This raw file (casos_covid19.csv) contained over 57,000 individual reports, which is too messy to model directly. To find a clear trend, we first had to process this data into a clean time series. This involved three steps: first, we filtered the list to keep only "confirmado" (confirmed) cases. Second, we isolated all cases where the sample was taken in the 2024 calendar year. Finally, we "aggregated" these individual reports, counting the total number of cases for each day. This process turned our messy list of reports into a clean, 366-day dataset, covid_cases_processed.csv, which was the foundation for our analysis.

No major data quality issues were found, but we did note an outlier pattern: the early part of the year had a short spike of cases (hundreds per day) followed by a long period of near-zero cases. This kind of pattern can pose challenges for simple linear models, as discussed later. We did not remove any outliers; instead, we will address them through robust modeling choices.

Data Exploration: What Are We Trying to Model?

Before writing any code, we plotted our new data. This is the most important step, as it tells us what challenges to expect.

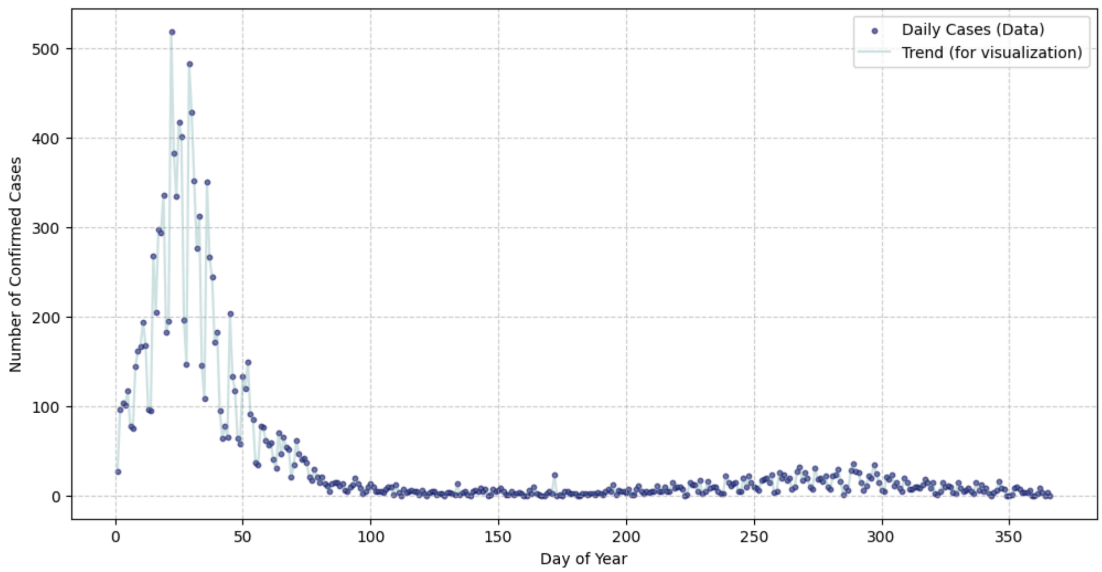


Figure 1. Daily confirmed COVID-19 cases in Buenos Aires throughout 2024. The data shows a sharp peak in cases in January, followed by a steep decline and an extended period of near-zero cases, with a small rise toward the end of the year. This non-linear trend poses challenges for simple linear models.

The plot in Figure 1 is central to the project. It shows that a straight-line regression isn't appropriate. The data follows an "S-shape," with a sharp peak in cases early in the year (January/February), a long flat period in the middle, and a small rise again toward the end. To capture this pattern, our model must be flexible enough to bend at least twice to capture this S-shape.

Model 1: Simple Linear Regression

As a first step, we always build the simplest possible model to act as a "control group" or baseline. This model, taken from Session 7, is a standard linear regression. It assumes the trend for the entire year can be explained with a single straight line. We define the independent variable x_i as the day of the year (1 through 365) and the dependent variable y_i as the case count for a given day:

$y_i \sim Normal(\mu_i, \sigma)$ and $\mu_i = a + b \cdot x_i$

Here a is the intercept (the model's estimate of cases at day 0) and b is the slope (the change in cases per day). We assume Gaussian ("normal") noise with standard deviation σ around this trend line. This is a standard linear regression assumption of constant variance residuals.

In a Bayesian framework (using PyMC), we set broad and non-informative priors for the parameters. These priors are weakly informative, just to constrain extreme values, as we lacked strong prior knowledge of the exact slope or intercept.

As expected, this model does not fit the data well. Figure 2 shows how it misses patterns in the trend. The model averages the early spike in January with the low values during mid-year, leading to a smooth downward slope. This creates a misleading picture, suggesting a constant rate of decline that isn't present. In reality, cases fall rapidly early in the year, stay low, and then rise again near the end. The linear model cannot capture this shape, so it isn't useful for understanding the actual pattern.

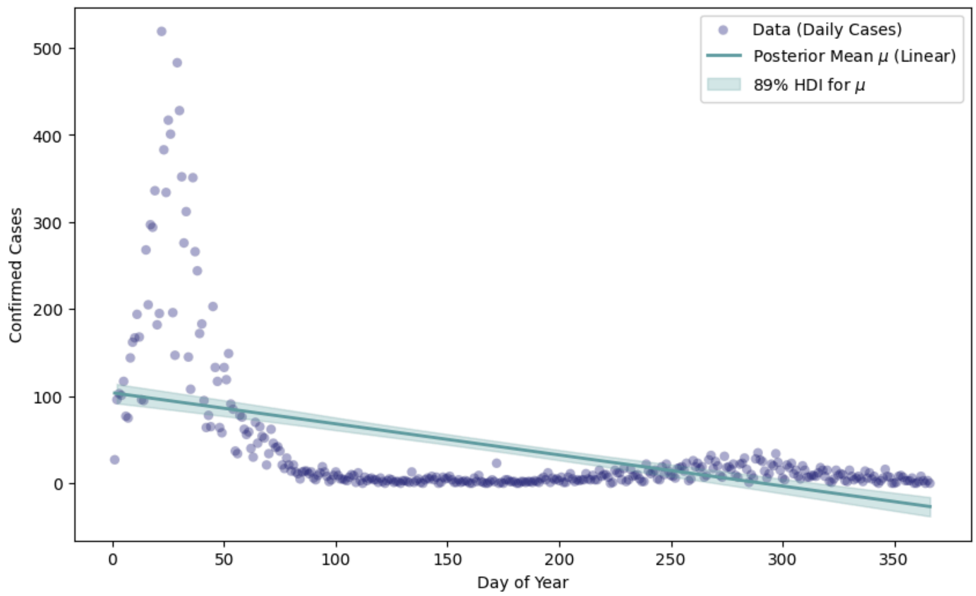


Figure 2. Linear regression model fit for daily COVID-19 cases in Buenos Aires, 2024. The line shows the posterior mean, with the shaded region indicating the 89% highest density interval (HDI). The model assumes a constant rate of change over time.

Table 1

Posterior summary for Model 1 (Simple Linear Regression)

```
--- Model 1: Summary ---
        mean    sd  hdi_5.5%  hdi_94.5%  mcse_mean  mcse_sd  ess_bulk
a      38.42  3.61     32.90      44.42       0.03     0.04  12617.18
b     -37.78  3.46    -43.32     -32.35       0.03     0.04  11893.72
sigma  69.26  2.54     65.26      73.29       0.02     0.03  11612.35

       ess_tail  r_hat
a       6308.83    1.0
b       6166.91    1.0
sigma   6414.75    1.0
```

The intercept *a* has a posterior mean of about 38.4, which suggests the model predicts around 38 daily cases at the start of the year (day 0). The slope β is −37.8, meaning the model expects the number of cases to drop by nearly 38 per day, which is far too steep and doesn't match the actual pattern of decline. The standard deviation σ of the residuals is about 69.3, indicating high variability in the data around the fitted line.

All r_hat values are 1.0, and effective sample sizes are high, meaning the model converged well and the estimates are stable. However, despite good sampling diagnostics, the model itself is not a good fit for the data.

Model 2: Polynomial Regression

To fix the problem from Model 1, we needed a model that could capture the "S-curve." As we saw in Session 9 with the cherry blossom data, a 3rd-degree (cubic) polynomial is perfect for this. Adding terms for x^2 and x^3 gives the line two "bending points," which is exactly what our S-curve needs. We kept the Normal likelihood, and model 2 is still a linear regression in the sense that it is linear in the parameters, but the predictor is non-linear in time.

$y_i \sim Normal(\mu_i, \sigma)$ and $\mu_i = a + b_1 x_i \cdot b_2 x_i^2 + b_3 x_i^3$

Here a is the intercept (baseline level), and b_1, b_2, b_3 are coefficients for linear, quadratic, and cubic terms of the day index x_i. We again assume a normal residual with standard deviation σ.

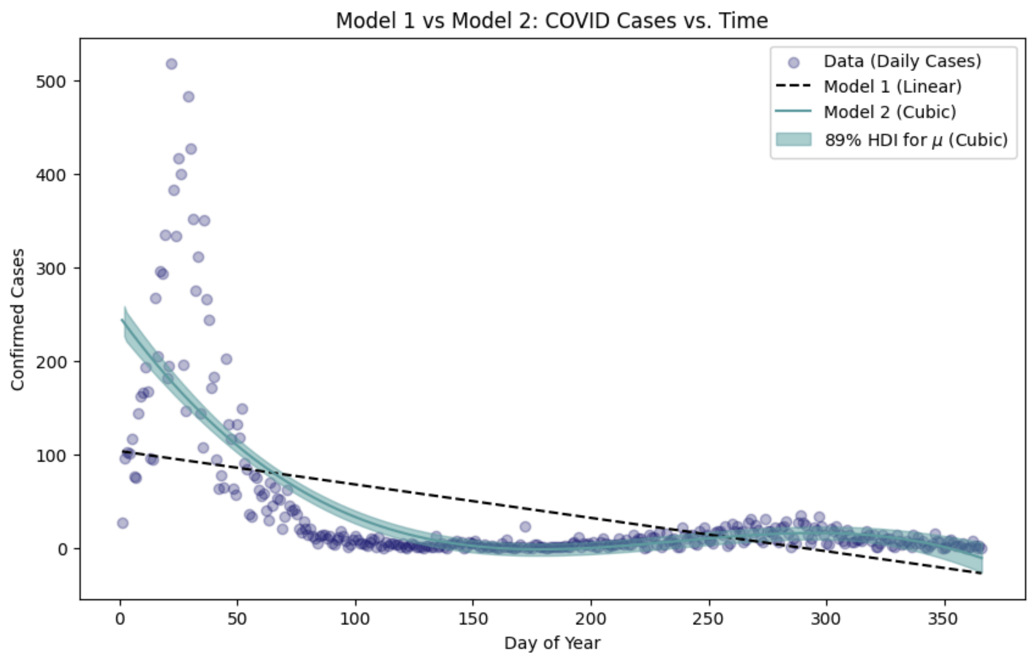


Figure 3. Comparison of Model 1 (Linear) and Model 2 (Cubic) fits to daily COVID-19 cases in Buenos Aires, 2024. The cubic model (solid line with shaded HDI) captures the early-year peak and late-year rise more than the linear model (dashed line), which oversimplifies the trend.

The cubic model in Figure 3 indeed fits the data trend much better. The posterior mean of u_i now traces a steep decline in January and levels off near zero for the rest of the year, matching the observed pattern. In comparison to the mild decline from the simple model.

Figure 4 shows the residuals from Model 2 plotted against time. The points are scattered evenly around zero for most of the year, suggesting the model fits well. At the start of the year, we see some larger residuals. This happens because the early spike in cases is hard to capture, even with a flexible model. Overall, this plot suggests that Model 2 captures the general pattern in the data more effectively than Model 1.

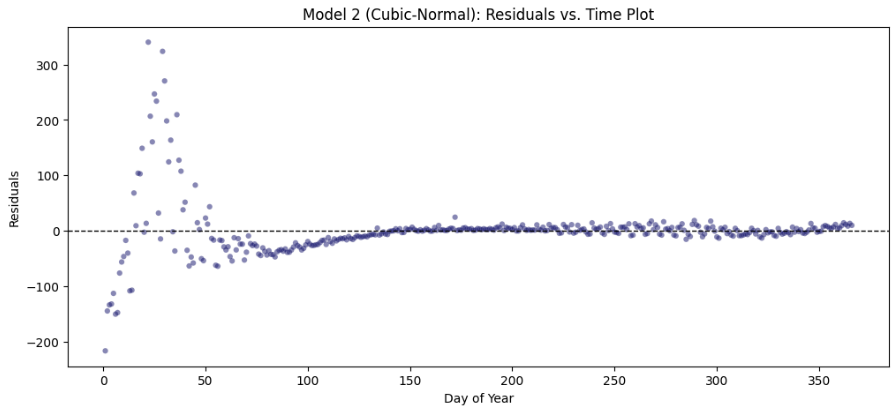


Figure 4. Residuals vs. Time for Model 2 (Cubic-Normal). The residuals are plotted across the year to assess model fit over time.

Table 2

Posterior Summary for Model 2 (Cubic Regression)

	mean	sd	hdi_5.5%	hdi_94.5%	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
a	-1.07	4.00	-7.92	4.92	0.05	0.04	5285.57	5163.37	1.0
b1	3.72	5.47	-4.85	12.57	0.08	0.07	4709.67	4287.72	1.0
b2	39.62	2.96	34.91	44.36	0.04	0.03	5231.56	5025.59	1.0
b3	-26.05	2.89	-30.66	-21.42	0.04	0.04	4748.83	4570.66	1.0
sigma	52.41	1.98	49.32	55.65	0.02	0.02	6912.25	5294.79	1.0

In Table 2, all R-hat values are 1.0, which suggests the model converged well. The effective sample sizes are all high, so the estimates are stable. The HDI ranges help us understand the uncertainty in each parameter, and we can see that the polynomial terms are all meaningfully different from zero, especially b2 and b3, which help the model bend and follow the shape of the data.

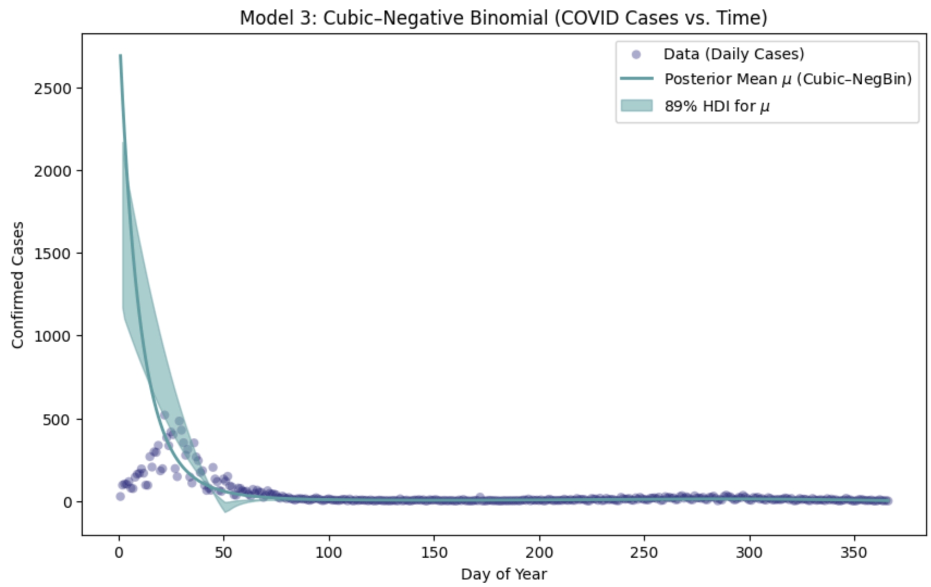
While the polynomial regression captures the trend shape well, it still uses a normal error assumption with a single σ for all days. This may not be entirely appropriate. We notice that the variance of residuals in early January is much larger than the variance of residuals later (when counts are ~0, residual variance is tiny). Model 2 still assumes homoscedasticity (constant variance). This mismatch can affect predictive accuracy and inference.

This issue became even clearer during model comparison. Even though this is already model 2, when we ran PSIS-LOO cross-validation, we received the warning "Estimated shape parameter of Pareto distribution is greater than 0.70 for one or more samples. You should consider using a more robust model…" (Appendix A). This warning suggests that some observations were influential and that the model's predictive distribution differed a lot from the actual posterior for those points, which is especially problematic when the likelihood is misspecified. In our case, this was likely due to extreme early-January values where the Normal assumption and fixed sigma couldn't accommodate the high variance. This is why there's a new model next.

Model 3: The Right Tool

For Model 3, we keep the same polynomial trend (cubic mean structure) but change the distributional assumption for the outcome to better handle the characteristics of the data. We use a Negative Binomial (NB) likelihood, which is a common choice for over-dispersed count data (*Negative Binomial Regression | R Data Analysis Examples*, n.d.). The NB can be thought of as a Poisson whose rate varies randomly (Gamma-distributed). This overdispersion gives heavier tails (more probability on extreme counts) than a pure Poisson.

$y_i \sim NegativeBinomial(\mu_i, \alpha)$ and $log(\mu_i) = a + b_1 x_i \cdot b_2 x_i^2 + b_3 x_i^3$, where α is a new parameter that learns the "overdispersion," or the extra spikiness/variance that the Normal distribution couldn't handle.



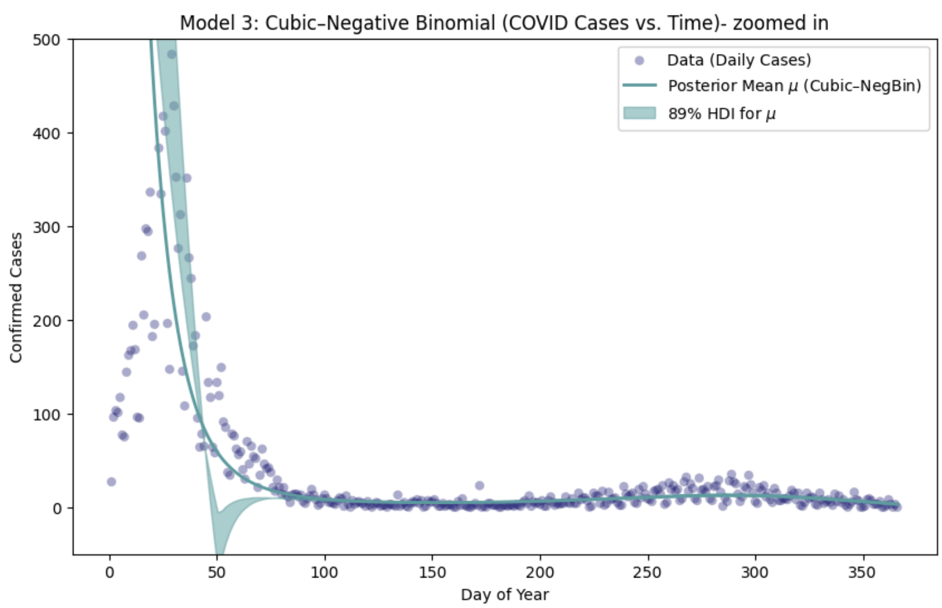


Figure 5. Posterior predictions from Model 3 (Cubic Regression with Negative Binomial Likelihood). The top plot shows the full amount of confirmed cases, while the bottom plot limits to 500 maximum. The model captures the steep early-year spike and adjusts variance through the Negative Binomial distribution. Uncertainty (HDI) is highest where counts are large and more variable, and narrows as daily cases decline.

After fitting Model 3, we saw that it offered the best overall fit. Its trend line matches that of Model 2, capturing the sharp decline at the start of the year and the long flat stretch that follows. The main improvement is in how it handles uncertainty. Unlike Model 2, Model 3 allows for wider credible intervals when case counts are high and tighter intervals when counts are low. This better reflects the changing variability in the data.

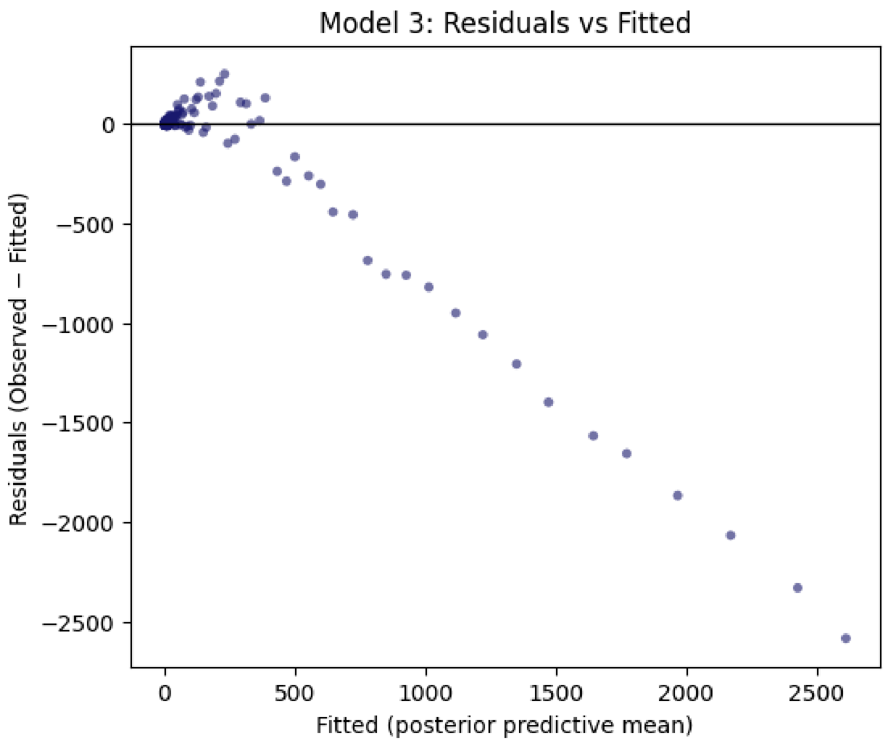


Figure 6. Model 3: Residuals vs. Fitted Values Plot. This plot shows the residuals (observed – fitted) against the fitted values for Model 3 (Cubic-Negative Binomial).

In Figure 6, we see that Model 3 handles the largest peaks in the data without producing extreme residuals, suggesting that it better accounts for the variability in high-count periods. The downward pattern in residuals at higher fitted values is expected under a Negative Binomial model, where variance increases with the mean. Importantly, there is no strong pattern of over- or under-prediction across the range, which indicates that the model adapts well across different case count levels. Additionally, Figure 7 demonstrates that at the beginning of the year, there are larger residuals, but they rapidly converge to values very close to 0, better than the previous model.

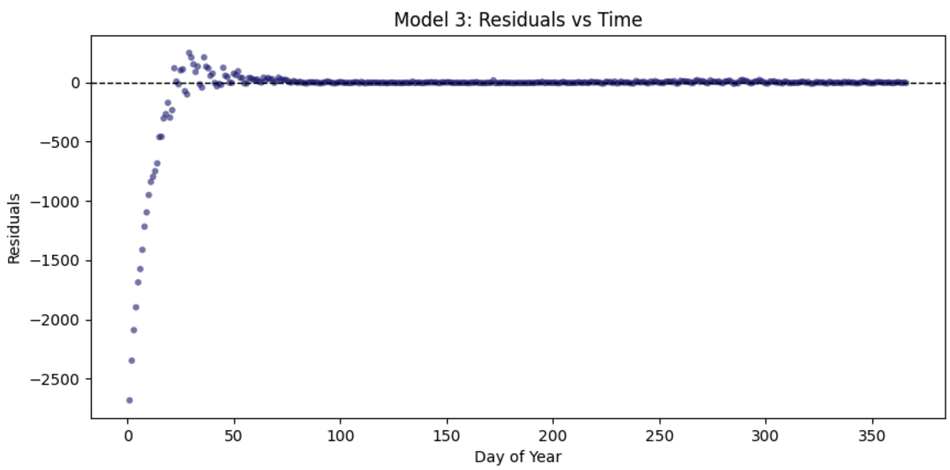


Figure 7. Residuals vs. time plot for Model 3 (Cubic-Negative Binomial). The spread of residuals is much wider at the beginning of the year, where counts are high, and becomes tighter and centered around zero as counts approach zero.

Table 3

Posterior Summary for Model 3 (Cubic-Negative Binomial)

--- Model 3 Sampling Complete ---

	mean	sd	hdi_5.5%	hdi_94.5%	mcse_mean	mcse_sd	ess_bulk
a	1.71	0.07	1.60	1.83	0.0	0.0	4954.49
b1	0.78	0.11	0.60	0.95	0.0	0.0	4799.04
b2	0.96	0.06	0.86	1.04	0.0	0.0	4973.93
b3	−0.91	0.06	−1.01	−0.80	0.0	0.0	4794.70
alpha	1.75	0.15	1.51	1.98	0.0	0.0	6014.55

	ess_tail	r_hat
a	4495.69	1.0
b1	4412.15	1.0
b2	4554.04	1.0
b3	4125.00	1.0
alpha	5124.26	1.0

Checking Table 3 confirms that all parameters are estimated with precision. The cubic terms (b1, b2, b3) define the shape of the trend line and are clearly non-zero, which supports the need for a flexible curve. The alpha parameter governs the dispersion in the Negative Binomial model. Its estimate (around 1.75) shows the extra variability present in the early part of the year, especially around the outbreak peak. All convergence diagnostics (r_hat close to 1, high effective sample sizes) show that the model was well estimated and the sampler worked.

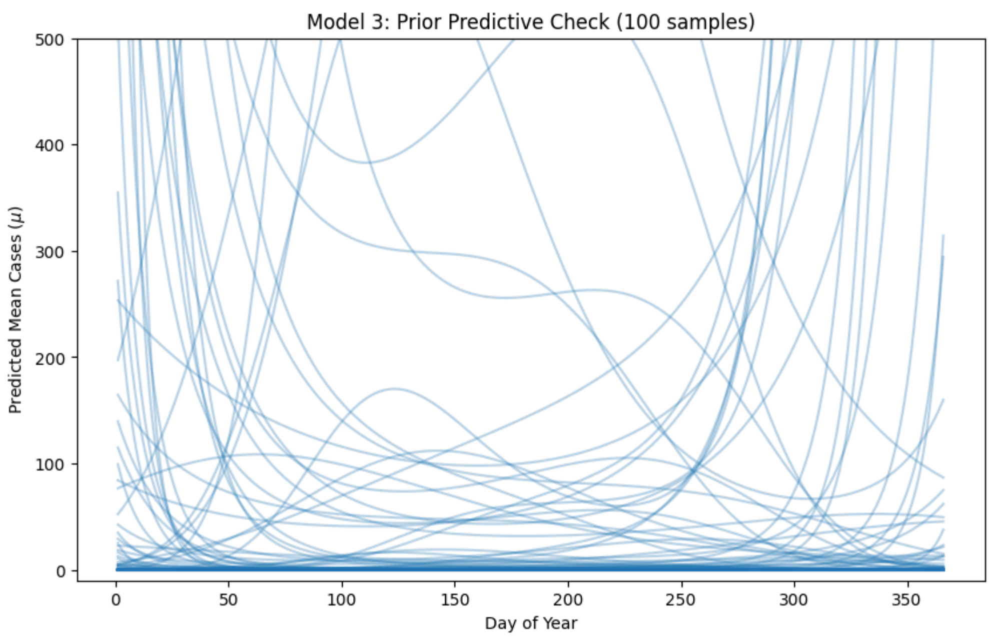


Figure 8. Prior predictive check for Model 3 (Cubic-Negative Binomial). Each line shows a sampled prior trajectory for the expected daily case counts. The spread shows the flexibility of the prior in accommodating a range of plausible pandemic trends.

Finally, the prior predictive check in Figure 8 confirms that our priors in Model 3 are well-calibrated. The model is flexible enough to simulate both sharp spikes and flat trends in daily case counts, even before seeing the actual data. This is important because it shows that the model's structure and assumptions were capable of capturing the real-world behavior of the COVID-19 curve, without being restrictive or too vague.

.

Model Comparison

We now have three models. We use PSIS Leave-One-Out Cross-Validation (LOO). This is a "scorecard" that measures each model's out-of-sample predictive accuracy.

Table 4

Model Comparison Using PSIS-LOO Cross-Validation

```
--- Model Comparison Results ---
                     rank     elpd_loo      p_loo   elpd_diff   weight
Model 3 (Cubic-NegBin)   0 -1339.899216   5.122618    0.000000  0.98791
Model 2 (Cubic-Normal)   1 -1975.867622  15.088885  635.968406  0.00000
Model 1 (Linear)         2 -2074.747001   8.808981  734.847786  0.01209

                            se        dse  warning scale
Model 3 (Cubic-NegBin)  31.061470   0.000000    False   log
Model 2 (Cubic-Normal)  42.042996  37.601945     True   log
Model 1 (Linear)        36.473492  37.513417    False   log
```

Table 4 also shows that Model 3 (Cubic-Negative Binomial) is clearly the best model. It has the highest elpd_loo score (−1339.9), meaning it fits the data better than the others. Model 2 and Model 1 have much lower scores, with differences of over 600 and 730 points, respectively. This is a large and meaningful gap. The weight column shows that Model 3 holds almost all the model weight (0.98791), confirming its strength compared to the others.

Furthermore, Model 2 has a warning: True. This is because its Pareto-k diagnostics were "bad" and "very bad," meaning the model is unstable and its score can't even be fully trusted. The Model 3 has no warnings and all "good" k-values, which indicates that it is both accurate and reliable.

We can also check the rank plots to confirm the model’s reliability.

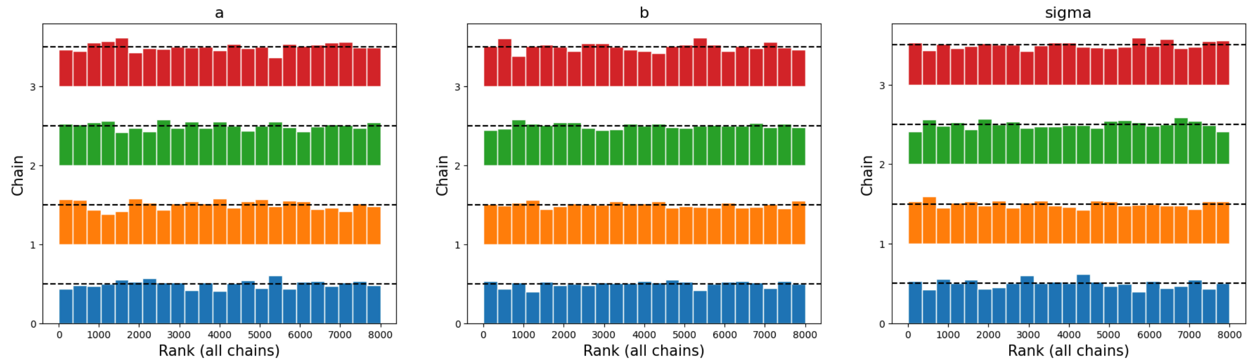


Figure 9. Rank histograms for Model 1 (Linear). The parameter draws are evenly spread across chains, showing no sampling issues, but the model itself remains a poor fit to the data.

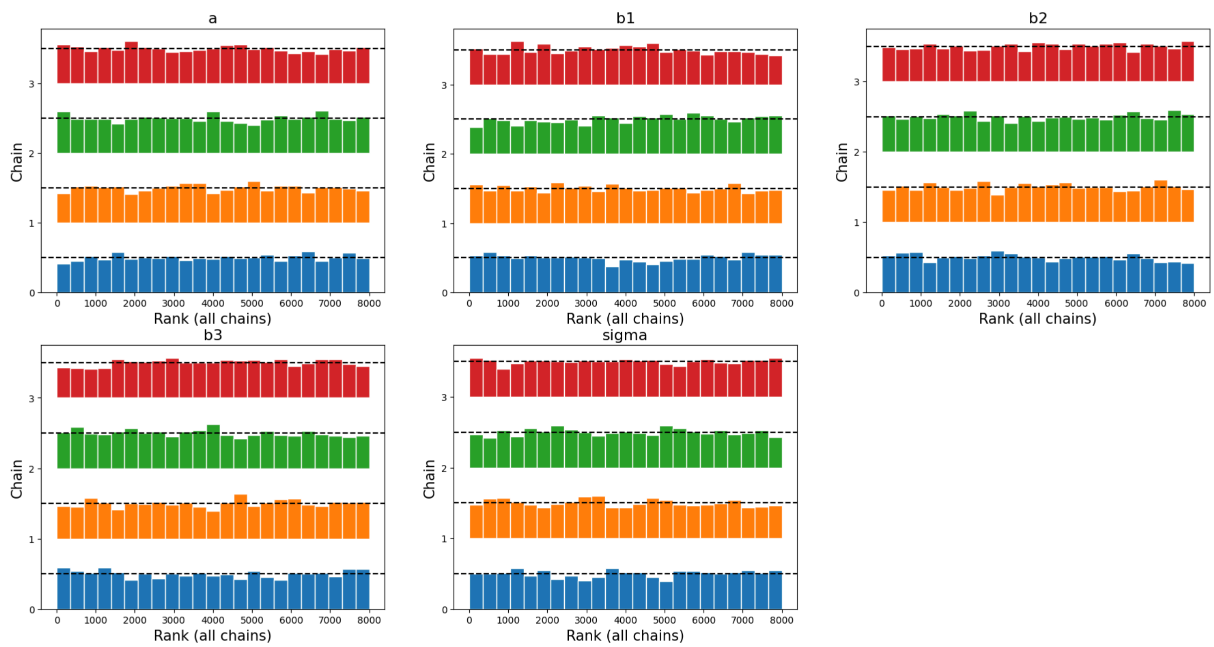


Figure 10. Rank histograms for Model 2 (Cubic-Normal). The chains are well-mixed and the ranks are flat, but this model still struggles with changing variance in the early-year spike.

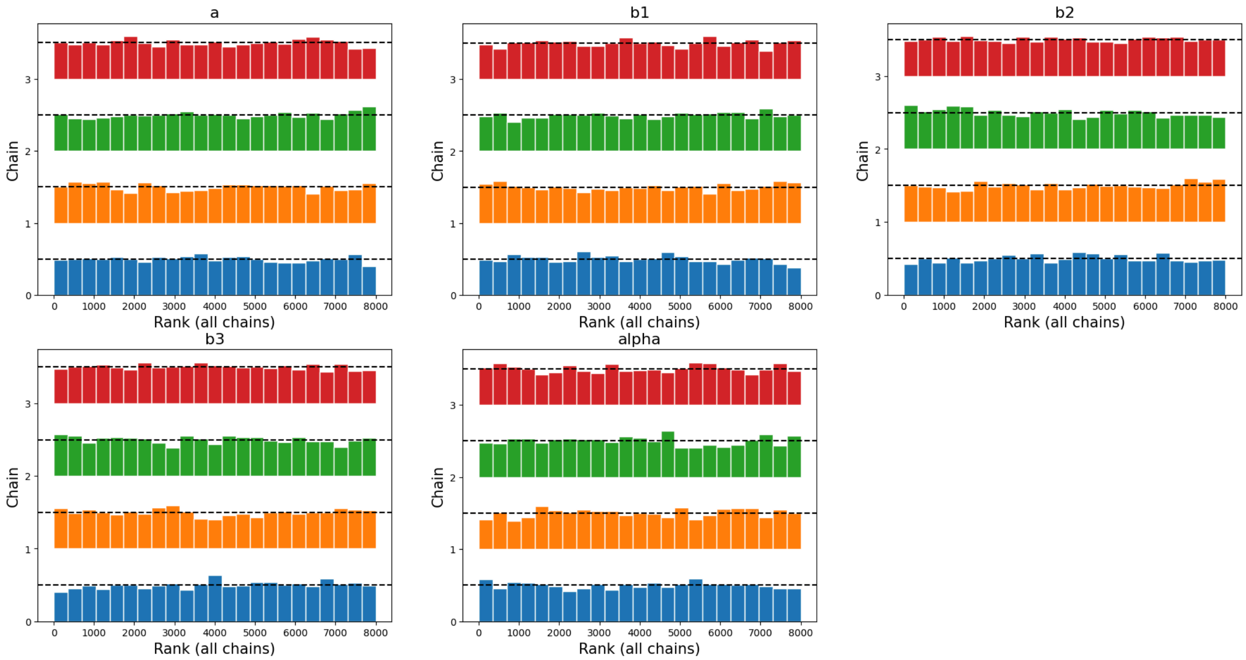


Figure 11. Rank plots for Model 3 (Cubic-Negative Binomial). All parameters, including alpha, show balanced rank histograms across chains, which confirms proper convergence.

The rank plots for Model 1 (Figure 9) and Model 2 (Figure 10) show that both models sampled in a good way. The histograms are mostly flat across all chains, which shows good mixing and no convergence issues. This means that while the models may differ in how well they fit the data, the underlying estimation process for both was stable and trustworthy. However, even with well-behaved chains, Model 2 still suffers from a mismatch between its constant-variance assumption and the real-world data variability, especially during the early-year spike.

The Model 3's rank plots (Figure 11) confirm that it is well-behaved. All parameters, including the new dispersion term (alpha), show uniform and stable ranks across chains. There are no patterns or imbalances, which shows the sampler did a good job exploring the posterior.

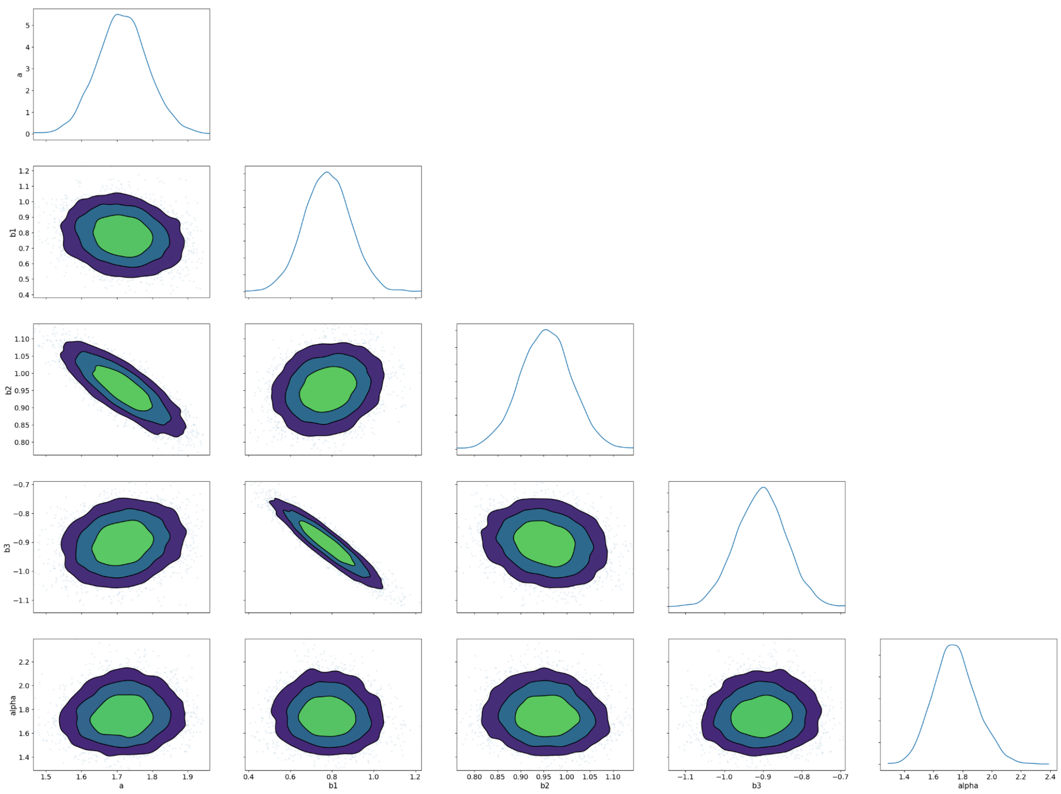


Figure 12. Pairwise posterior plots for Model 3. Each plot shows the joint distribution between parameter pairs. Clear and concentrated shapes show stability and low correlation, with no signs of multimodality.

Lastly, Figure 12 gives another view of how the parameters interact in Model 3 through the pair plot. We see that each parameter's posterior is concentrated and unimodal. Some mild correlations between b1, b2, and b3 are expected because of the polynomial structure. These relationships are smooth and interpretable. There are no odd shapes or long tails, which supports our earlier conclusion that the model is both stable and flexible.

Conclusion

Our analysis shows that the daily COVID-19 cases in Buenos Aires 2024 have a rapid exponential decline in early 2024, followed by an extended period of near-zero cases, modeled via a cubic polynomial trend on a Negative Binomial regression. The best model (Model 3) captures both the sharp drop in the first month and the low flat trend thereafter, while accounting for the high variability during the peak. Simpler models either failed to capture the shape or the uncertainty structure.

References

Buenos Aires Data | Casos COVID-19. (n.d.). Buenos Aires Data. https://data.buenosaires.gob.ar/dataset/casos-covid-19

Negative Binomial Regression | R Data Analysis Examples. (n.d.). https://stats.oarc.ucla.edu/r/dae/negative-binomial-regression/)

Appendix

Appendix A - Code Output

```
--- Starting Model Comparison (PSIS-LOO) ---
/usr/local/lib/python3.12/dist-packages/arviz/stats/stats.py:797: UserWarning: Estimated shape parameter of Pareto distribution is greater than 0.70 for one or more samples. You should consider using a more robust model, this is because importance sampling is less likely to work well if the marginal posterior and LOO posterior are very different. This is more likely to happen with a non-robust model and highly influential observations.
  warnings.warn(
--- Model Comparison Results ---
                  rank     elpd_loo      p_loo  elpd_diff   weight        se  \
Model 2 (Cubic)      0 -1975.867622  15.088885   0.000000  0.93975  42.042996
Model 1 (Linear)     1 -2074.747001   8.808981  98.879379  0.06025  36.473492

                        dse  warning scale
Model 2 (Cubic)    0.000000     True   log
Model 1 (Linear)  15.168312    False   log
```

Appendix B - Code From Class Citation

Scheffler, K. (2025, September 29). *CS146 Session 7 – [4.1] Linear regression – pre-class work*. CS146: Computational Methods for Bayesian Statistics, University course material.

Scheffler, K. (2025, October 1). *CS146 Session 8 – [4.2] Robust linear regression – pre-class work*. CS146: Computational Methods for Bayesian Statistics, University course material.

Scheffler, K. (2025, October 6). *CS146 Session 9 – [5.1] Linear regression for non-linear data – pre-class work*. CS146: Computational Methods for Bayesian Statistics, University course material.

Appendix C - AI Statement

I used Gemini to help me organize what I needed to include in the paper, so I wouldn’t get lost and would ensure that I included everything. Also, I sent a dataset to ask if it would work to use for the assignment (even though it didn't help me because it said it would work, and then when I did the simple regression, I saw that it had no correlation whatsoever, so I changed to the COVID one).

Besides, I used Grammarly to help with my writing, and that’s it. The biggest part was from the code we had for Pre-Class Work, which helped a lot.

Prompt: (instructions) This is an assignment that I have to do, I am confused with the requirements for it, can you help me giving a checklist so I dont forget anything?

Answer:

📝 Project Deliverables Checklist

Category	Requirement	Status (Check Box)	Notes
Primary Upload	Single PDF document (Formal Report)	☐	Max 10,000 words (excluding appendices). Must be professional and client-focused.
Secondary Upload	Zip file containing all supporting materials	☐	
Supporting Material	Your Data Set (e.g., CSV file)	☐	Must be loadable by the code.
Supporting Material	Well-organized Jupyter Notebook	☐	Headings must match the PDF report.
Supporting Material	Any other optional supporting material	☐	

🔎 **Data Collection & Preparation Checklist**

Requirement	Status (Check Box)	Notes
Find a data set **related to your current city, region, or country** (rotation city).	☐	e.g., demographic, economic, meteorological, traffic.
Data set is **suitable for linear regression** modeling.	☐	May require data transformation.
Cite all data sources clearly (including URL(s)).	☐	In the report and notebook.
Transform data if necessary (e.g., for non-linearity) **before** modeling.	☐	Document the transformation in the report.

Exportar para as Planilhas

Prompt: does any dataset work? do you have any tips to choose one?

Answer: Yes, any real dataset works as long as it is relevant to your rotation city and you can model it using linear regression.

The core challenge isn't just finding data, but finding data that lets you build and compare two distinct, sensible linear regression models and allows you to clearly justify your modeling choices.

Here are key tips for choosing a dataset that will make the rest of your project easier and stronger:

💡 Tips for Choosing the Best Dataset

1. Focus on the Variables (The Relationship)

A successful project hinges on having an interesting relationship to model.

Continuous Dependent Variable (Y): Your target variable must be continuous (a number that can take any value in a range, like price, temperature, or speed).

A Clear Hypothesis: Choose a relationship that you can explain and motivate. For example:

"How does the total square footage of a property (X) relate to its price (Y) in Buenos Aires?" (A classic linear relationship).

"How does the yearly number of tourists (X) relate to the average monthly restaurant sales (Y) in the city?"

"How does the daily maximum temperature (X) relate to daily electricity consumption (Y)?" (Likely non-linear, requiring a transformation or polynomial model).

2. Ensure Data Quality and Quantity

Sufficient Data: You need enough data points (rows) to fit two models, especially if you plan to use multiple predictors or higher-order polynomials/splines.

Check for Noise/Outliers: If a variable has obvious outliers or a heavily skewed distribution (like property prices), that's actually good for the project! It immediately provides a strong motivation for your second model, such as using a log-transformation or a robust error distribution in PyMC.

Prompt: (dataset from ecobici in Buenos Aires, an app to rent bicycles) would this one work? I just want to know if I can start with it

Answer: Yes, this dataset will work for your project, but you will need to perform an extra step of data aggregation and manipulation to create a suitable continuous dependent variable (Y) for linear regression.