Data Analysis Report

By: Daniel Gregory

In recent years the space launch industry has taken off. In 2024 alone, there were a record-setting 259 launch attempts, marking the fourth consecutive year of increasing activity [1]. The majority of these launches serve commercial purposes; in fact, as of 2019, approximately 95% of revenue in the space sector came from the “space-for-Earth” economy, where satellites are used to deliver goods and services such as communications, weather monitoring, and navigation[2].With this surge in demand, small-lift launch vehicles have become increasingly prevalent for placing payloads into orbit efficiently. These small lift launch vehicles can typically transport up to 4,000 pounds (about 1,800 kilograms), and while they cannot match the capacity of larger launch systems, their smaller size makes them a more cost-effective option for organizations looking to deploy satellites efficiently. This has made them particularly attractive to small satellite operators, research institutions, and emerging commercial players.

To better understand patterns in launch success over time, particularly in relation to small-lift launch vehicles, I analyzed data from a CSV file containing detailed records of launch attempts. This dataset includes variables such as the type of launch vehicle, the number of days since the first launch attempt of each vehicle type, and whether the launch was successful. To explore these relationships, I constructed two Bayesian logistic regression models: the first using only time since first attempt as a predictor, and the second adding a random effect for vehicle type. A few key trends on this dataset reveal important insights about launch success and vehicle performance. Overall, approximately 84.74% of all launch attempts were successful. Among those that were the first attempt for a given vehicle type, the success rate was 60.00%, suggesting that reliability often improves with more launches. The most frequently used launch vehicle in the dataset is the Electron, While several launch vehicles currently report a 100.00% success rate, the Long March 6 is notable for achieving this perfect record across a significant number of launches, with 24 successful missions to date giving it the greatest apparent success rate.

As previously stated, before the first model will use Bayesian logistic regression to explain launch success in terms of days since the first launch attempt for the type of vehicle. The JAGS code for this model is as follows:

```
# Write model to a file
model_string <- "
model {
  for (i in 1:N) {
    y[i] ~ dbern(p[i])
    logit(p[i]) <- beta0 + beta1 * x[i]
  }
  beta0 ~ dt(0, pow(1/10, 2), 1)
```

[1] Space Foundation Editorial Team, "The Space Report 2024 Q4 shows record annual launches, strong H2 market performance, and growing demand for tracking and removal of orbital debris," *Space Foundation*, Jan. 21, 2025. [Online]. Available: https://www.spacefoundation.org/2025/01/21/the-space-report-2024-q4/[Accessed: May 9, 2025].

[2] M. Weinzierl and M. Sarang, "The commercial space age is here," *Harvard Business Review*, Feb. 12, 2021. [Online]. Available: https://hbr.org/2021/02/the-commercial-space-age-is-here. [Accessed: May 9, 2025].

```
  beta1 ~ dt(0, pow(1/2.5, 2), 1)
}
"
writeLines(model_string, "model.bug")
```

In this analysis, three chains were run for the Bayesian model, each with 5000 iterations and a burn-in period of 2000 iterations. The samples were thinned with a thinning interval of 3, meaning every third sample was retained, resulting in a sample size of approximately 1666 for each chain. After burn-in, the effective sample sizes (ESS) for the key parameters were calculated. For the intercept parameter (beta0), the ESS was 4142.470, and for the slope parameter (beta1), the ESS was 4200.885. These ESS values indicate that the chains have mixed well and that the number of independent samples is sufficiently large, providing reliable estimates of the posterior distributions for both parameters. Based on these results, the model appears to have converged appropriately, and the sampling process is considered effective for the subsequent analysis and interpretation.

The approximate posterior mean, posterior standard deviation, and 95% central posterior interval for each regression coefficient are as follows:

```
1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

       Mean     SD Naive SE Time-series SE
beta0 1.860 0.1634 0.002312       0.002540
beta1 1.374 0.4284 0.006060       0.006613

2. Quantiles for each variable:

        2.5%   25%   50%   75% 97.5%
beta0 1.5532 1.745 1.853 1.968 2.198
beta1 0.5932 1.074 1.352 1.650 2.252
```

The posterior probability that the slope coefficient (beta1) related to days after the first launch attempt is positive is 1. This indicates strong evidence that the slope is positive, suggesting that as time progresses after the first launch attempt, the launch vehicle's success rate improves. Therefore, it seems that a type of launch vehicle does indeed become more reliable over time.

Based on the first model, the 95% central posterior interval for the probability of a successful launch on the first attempt is [0.825, 0.900]. This means we are 95% confident that the true probability of a successful launch on the first attempt lies within this interval. The result suggests that the probability of success for a first attempt launch is relatively high, with a lower bound of 82.5% and an upper bound of 90.0%.

The DIC for the model was calculated as 315.4, with a mean deviance of 313.4 and a penalty of 1.993. The effective number of parameters is 2, corresponding to beta0 and beta1. This is equal to the actual number of parameters in the model, indicating that both coefficients are actively contributing to the model's predictions.

The second model builds on this by introducing a vehicle-specific random intercept to account for variation between different launch vehicle types. In addition to the fixed effects beta0 and beta1, this model includes a random effect uj for each vehicle type j, where uj$\sim$Normal($0,\sigma_{vehicle}^2$). A hyperprior is placed on $\sigma_{vehicle}$, typically specified using a distribution that is approximately flat over $(0,\infty)$. The code for this model is as follows:

```
cat("
model {
  for (i in 1:N) {
    y[i] ~ dbern(p[i])
    logit(p[i]) <- alpha + beta * x[i] + vehicle_effect[vehicle[i]]
  }
  #priors for fixed effects
  alpha ~ dt(0, 1 / (10^2), 1)
  beta ~ dt(0, 1 / (2.5^2), 1)

  #random effects for vehicle type
  for (j in 1:J) {
    vehicle_effect[j] ~ dnorm(0, tau_vehicle)
  }
  #hyperparameter for standard deviation
  tau_vehicle <- 1 / (sigma_vehicle^2)
  sigma_vehicle ~ dunif(0, 10)
}
", file = "model2.txt")
```

The second model was run using three Markov chains, each with a burn-in of 6,000 iterations and a total of 10,000 iterations per chain. A thinning interval of 3 was applied, yielding 3,333 samples per chain and a combined posterior sample size of 9,999. Over dispersed starting values were used to aid convergence. The effective sample sizes for the top-level parameters were 2,521 for alpha, 5,780 for beta, and 1,649 for $\sigma_{vehicle}$. Additionally, ESS values for the 35 vehicle-specific random effects ranged from approximately 3,947 to 9,656, indicating strong mixing and reliable estimation across parameters.

The posterior summaries for the key parameters in the second model are as follows. The intercept (alpha) has a posterior mean of approximately 1.68 with a standard deviation of 0.31 and a 95% central posterior interval of (1.07, 2.30). The slope coefficient (beta), which corresponds to the centered and rescaled number of days after the first launch attempt, has a posterior mean of about 1.06, a standard deviation of 0.47, and a 95% posterior interval of (0.16, 2.01). The standard deviation of the vehicle-specific random effects ($\sigma_{vehicle}$) has a posterior mean of 1.28, a standard deviation of 0.37, and a 95% posterior interval of (0.67, 2.12).

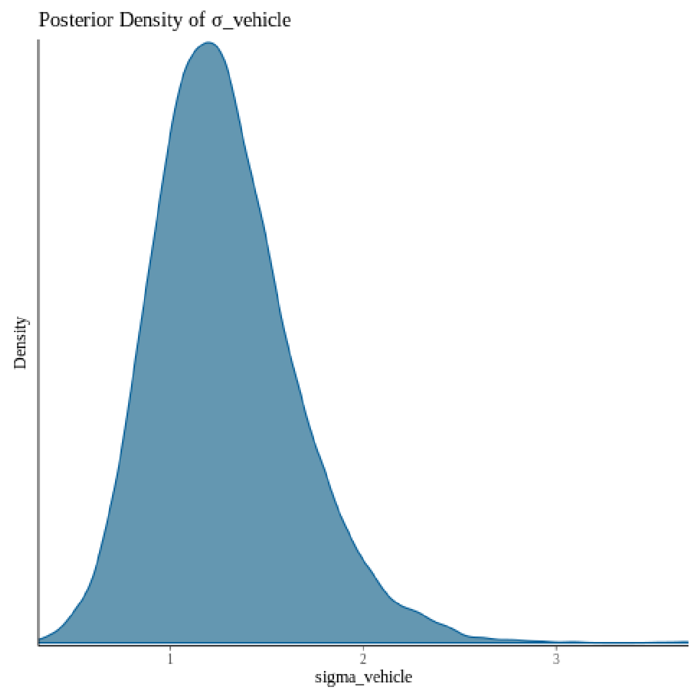


*Figure 1: Approximate density plot for σvehicle. This plot shows the posterior distribution of the vehicle reliability parameter, σvehicle, highlighting its estimated density across the sampled values.*

As seen in figure 1, The approximate density plot for the standard deviation of the vehicle-specific random effects ($\sigma_{vehicle}$) peaks around 1.3 and gradually declines, suggesting notable variability across vehicle types. Since $\sigma_{vehicle}$ is clearly greater than zero and the distribution is concentrated away from zero, it would not be reasonable to assume that all launch vehicles are equally reliable after adjusting for the number of days since the first launch attempt. The variation captured by $\sigma_{vehicle}$ indicates that vehicle-specific effects play a meaningful role in predicting launch success.

The Deviance Information Criterion (DIC) for the second model was approximately 291.1, with a mean deviance of 270.5 and an effective number of parameters (penalty) of 20.54. Compared to the first model (DIC ≈ 315.4, penalty ≈ 2), the second model shows a substantial improvement in fit, despite a higher complexity. The reduction in both the mean and penalized deviance indicates that allowing for random effects by vehicle type results in a better model overall.

Overall, this model determined that the most reliable type of launch vehicle, defined as the one with the highest posterior probability of having the largest random effect, is Long March 6. It has an approximate posterior probability of 0.21 of being the most reliable. Given that there were 35 vehicles, this is a relatively high probability indicating that Long March 6 stands out notably in terms of reliability.

The analysis done here suggests that the likelihood of a successful rocket launch tends to improve the longer a vehicle type has been in use. While the initial model showed this general trend, the second model, which accounted for differences between vehicle types, provided a better fit to the data. This model revealed that not all launch vehicles are equally reliable — some perform significantly better than others even after adjusting for experience over time. In particular, the Long March 6 vehicle stood out as the most reliable, with the highest likelihood of success among all types considered.

Appendix

This section provides all of the R code used in this Data Analysis Report.

```
# Install JAGS
!apt-get install -y jags

# Load required libraries
import rpy2
from rpy2.robjects import r

r('install.packages("rjags", repos="http://cran.rstudio.com")')
r('library(rjags)')
r('library(coda)')
r('library(dplyr)')

# Load the dataset
data = r.read.csv("launches.csv")

# Center and rescale SinceFirst
data['SinceFirst_scaled'] = (data['SinceFirst'] - data['SinceFirst'].mean()) / (2 *
data['SinceFirst'].std())

# Prepare data for JAGS
data_jags = {
    'y': data['Success'],
    'x': data['SinceFirst_scaled'],
    'N': len(data)
}

# Write model to a file
model_string = """
model {
  for (i in 1:N) {
    y[i] ~ dbern(p[i])
```

```
    logit(p[i]) <- beta0 + beta1 * x[i]
  }
  beta0 ~ dt(0, pow(1/10, 2), 1)
  beta1 ~ dt(0, pow(1/2.5, 2), 1)
}
"""
with open("model.bug", "w") as f:
    f.write(model_string)

# Initialize and run the model
model = jags.model("model.bug", data=data_jags, n.chains=3, n.adapt=1000)

# Burn-in
model.update(2000)

# Sample from posterior
samples = coda.samples(model, variable.names=["beta0", "beta1"], n.iter=5000, thin=3)

# Combine chains
posterior = as.mcmc(do.call(rbind, samples))

# Summary
summary_stats = summary(samples)
ess_value = effectiveSize(samples)
print(summary_stats)
print(ess_value)

# Posterior P(beta1 > 0)
prob_positive_beta1 = mean(posterior[, "beta1"] > 0)
print("Posterior P(beta1 > 0):", prob_positive_beta1)

# Probability of success on first attempt (x = 0)
p_first = 1 / (1 + exp(-posterior[, "beta0"]))
cred_interval = quantile(p_first, probs=c(0.025, 0.975))
print("95% CI for success on first attempt:", cred_interval)

# DIC
dic_result = dic.samples(model, n.iter=5000)
print(dic_result)
```

```
# Preprocess the data
data2 = data.copy()
data2['SinceFirst_scaled'] = (data2['SinceFirst'] - data2['SinceFirst'].mean()) / (2 *
data2['SinceFirst'].std())
data2['VehicleID'] = pd.factorize(data2['Vehicle'])[0]

# Prepare data for JAGS
data_jags_2 = {
    'y': data2['Success'],
    'x': data2['SinceFirst_scaled'],
    'vehicle': data2['VehicleID'],
    'N': len(data2),
    'J': len(data2['Vehicle'].unique())
}

# Write JAGS model to file
model_string_2 = """
model {
  for (i in 1:N) {
    y[i] ~ dbern(p[i])
    logit(p[i]) <- alpha + beta * x[i] + vehicle_effect[vehicle[i]]
  }
  alpha ~ dt(0, 1 / (10^2), 1)
  beta ~ dt(0, 1 / (2.5^2), 1)

  for (j in 1:J) {
    vehicle_effect[j] ~ dnorm(0, tau_vehicle)
  }
  tau_vehicle <- 1 / (sigma_vehicle^2)
  sigma_vehicle ~ dunif(0, 10)
}
"""
with open("model2.txt", "w") as f:
    f.write(model_string_2)

# Run the model
model = jags.model("model2.txt", data=data_jags_2, n.chains=3, n.adapt=1000)

# Burn-in
model.update(5000)
```

```
samples2 = coda.samples(model, variable.names=["alpha", "beta", "sigma_vehicle",
"vehicle_effect"], n.iter=10000, thin=3)

# Posterior summary
summary_stats = summary(samples2)
ess_value = effectiveSize(samples2)

posterior_mean_sd = summary_stats['statistics']['posteriorCI']
print(summary_stats)
print(posterior_mean_sd[["alpha", "beta", "sigma_vehicle"]])
print(ess_value)

# Load devtools
library(devtools)

# Install the package using devtools
devtools::install_github("stan-dev/bayesplot")

# Load bayesplot
library(bayesplot)

# Convert MCMC samples to matrix
samples_matrix = as.matrix(samples2)

# Plot density for sigma_vehicle
mcmc_dens(samples_matrix, pars="sigma_vehicle") + ylab("Density") + ggtitle("Posterior
Density of σ_vehicle")

# DIC
dic_val = dic.samples(model, n.iter=5000, type="pD")
print(dic_val)

# Most reliable vehicle analysis
vehicle_effect_samples = as.matrix(samples2)[, grep("vehicle_effect\\[",
colnames(as.matrix(samples2)))]
most_reliable_vehicle_index = apply(vehicle_effect_samples, 1, which.max)
vehicle_counts = table(most_reliable_vehicle_index)
most_likely_vehicle = as.integer(names(which.max(vehicle_counts)))
posterior_prob_most_reliable = max(vehicle_counts) / sum(vehicle_counts)
```

```
vehicle_name = levels(factor(data2['Vehicle']))[most_likely_vehicle]
print("\nMost reliable vehicle type:", vehicle_name)
print("Posterior probability it is most reliable:", round(posterior_prob_most_reliable, 3))
```