Theo Feng

Freedom Trip Forecasting: From Obsession to Optimization

1. Introduction

I'm the kind of person who sets two alarms, drives to San Pedro at 2 a.m., and falls asleep on a gunny sack of frozen squid—because the *Freedom* (my favourite tuna fishing boat) might find foaming bluefin. After a streak of "you-should-have-been-here-yesterday" trips, I started to wonder: was there a smarter way to plan a trip?

Could data tell me when the odds were better than pure luck?

To find out, I built a model to simulate catch rates using past data. The goal wasn't perfection, just consistency. I wanted a repeatable, data-based way to decide when to ride the *Freedom* out of 22nd Street Landing. Using open-source logs from 2023 and 2024 I collected:

- Fish counts and boat load info from public reports on 976-TUNA.com, a well-known Southern California sportfishing site.
- Moon phase data and water temperatures near Tanner Bank from the NOAA (*National Oceanic and Atmospheric Administration*), including buoy records and forecast tables.

I analyzed how fish per person per day varied with:

- Date (within the July–October window, which is the tuna saeson)
- Water Temperature
- Moon Illumination
- Boat Load (number of people)

2. The Model: Quadratic Logic

From my personal fishing experience, I predicted a shallow arch shape model—productivity first rises, then peaks, then falls. This shape made me choose a quadratic model. A straight line would miss the middle peak, and higher-order curves would overfit the small dataset.

The global model has this form:

$$\mathbf{y = \beta_0 + \beta_1 \cdot d + \beta_2 \cdot T + \beta_3 \cdot M + \beta_4 \cdot B + \beta_5 \cdot d^2 + \beta_6 \cdot T^2 + \beta_7 \cdot M^2 + \beta_8 \cdot B^2}$$

Where:

- *y* is the catch rate (fish/person/day)
- *d* is the day index (July 1 = 1)
- *T* is water temperature*M* is moon illumination percentage

- *B* is boat load (number of anglers)

This model is designed to be simple but flexible enough to capture "sweet spots" in each variable.

3. The Code

The entire analysis was built in MATLAB and divided into two phases: exploratory visualization and predictive modeling.

I started by loading trip data from Excel using `readtable`. Then I filtered for July to October trips and created a new variable, `day_index`, which counts the number of days since July 1. This provides a numeric timeline to fit into the model.

Next, I renamed the key columns to make them easier to work with: `WaterTemp` became `water_temperature`, `MoonPhase` became `lunar_light_percentage`, and so on.

For the initial analysis, I wrote a function called `quadPanel` to plot each variable against the catch rate and fit a quadratic curve. The function also displayed the R^2 value and p-value on the plot, helping me evaluate how strong the relationships were.

Then I built the global model using all four variables—both their original and squared forms. I combined them into a design matrix, then used least-squares regression (`X \ y`) to compute the model coefficients.

After fitting the model, I measured how well it performed using two key metrics:

- R^2 (explained variance): about 0.228
- RMSE (root mean squared error): around 0.455 fish/person/day

I also created a custom prediction function called `predictCatch`. It takes four inputs—day index, temperature, moon illumination, and people count—and outputs a catch prediction.

To test the model's reliability, I ran a bootstrap simulation. I randomly sampled 1000 trips from the original data and predicted catch rates for each one. The resulting histogram showed the distribution of expected outcomes across a typical season.

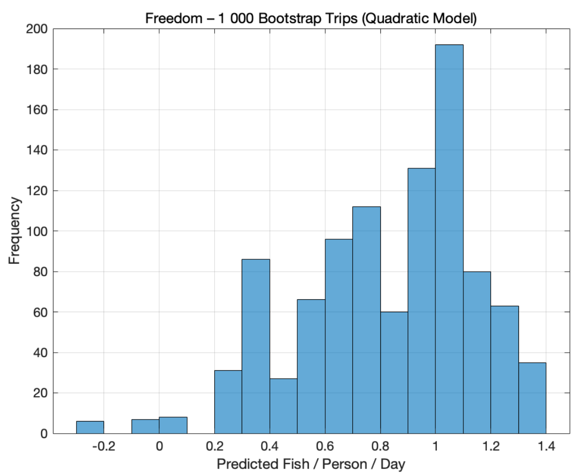
4. Result Analysis

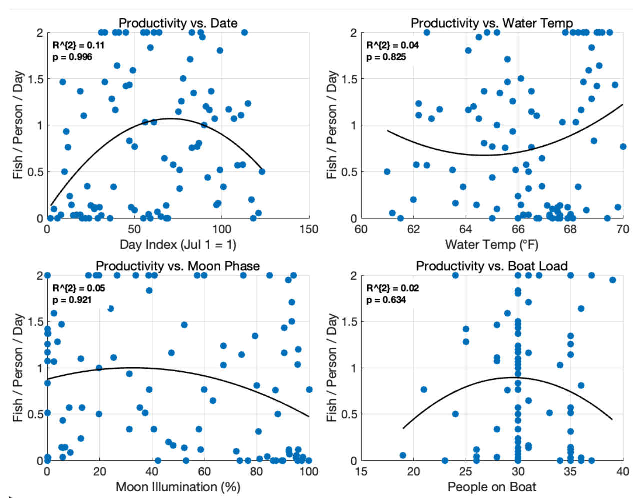
The scatterplots of individual variables showed very weak relationships with catch efficiency. The best R^2 was only 0.11, and the p-values were all above 0.6—statistically meaningless. That means no single factor is a good predictor on its own.

The global model, however, improved the picture. With an R^2 of 0.228, it explained nearly 23% of the variation in catch rates. That's still not high, but it shows the four variables work better together than apart.

The bootstrap simulation revealed a stable prediction range. Most simulated trips had outcomes between 0.6 and 1.2 fish per person per day. Very low or very high values were rare, which suggests the model is stable but conservative—it predicts average results well but not extremes.

In practical terms, this means the model is useful for deciding when a trip might be "worth it." For example, if a combination of conditions gives a prediction near 1.1, it may be a good bet. If the prediction is 0.3, maybe wait for a better day.





5. Conclusion

By combining real trip data with a basic quadratic regression model, I built a forecasting tool for offshore fishing on the *Freedom*. It doesn't promise bluefin. But it replaces guesswork with logic—and that makes every 2 a.m. drive to the dock a little less blind.

That said, luck still matters most. No model can predict the ocean perfectly, and there's no strong, universal rule hidden in the data. But if I had to tune my odds based on what I've seen, I'd look for:

- Mid August date
- Water temperature above 68°F
- Avoid full moon (low to moderate moonlight seemed better)
- Around 30 anglers on the boat (not packed, not empty)

Fishing will always involve uncertainty. But now I have a tool that helps me make smarter calls, especially when I'm debating whether to book a spot or sleep in.