**Bat vs Rat: The Forage Files – Investigation A**

**Introduction**

This project investigates whether Egyptian fruit bats (Rousettus aegyptiacus) perceive black rats (Rattus rattus) as competitors or potential predators. Zoologists collected data over seven months by monitoring a semi-natural bat colony using video surveillance. The aim of this investigation is to determine whether rat presence increases bats’ avoidance behaviour or alters their foraging vigilance.

**Data Description**

Two datasets were provided:

1. dataset1.csv – Each row represents a unique bat landing on the food platform, with variables including:

- `risk`: whether bats took risks to obtain food (0 = avoidance, 1 = risk-taking).

- `reward`: whether the behaviour was rewarding.

- `bat\_landing\_to\_food`: seconds taken to approach food.

- `seconds\_after\_rat\_arrival`: time since rats appeared.

- `hours\_after\_sunset`: time of night.

2. dataset2.csv – Each row represents a 30-minute observation period, with variables including:

- `rat\_arrival\_number`: number of rat arrivals.

- `bat\_landing\_number`: number of bat landings.

- `food\_availability`: estimated food remaining.

**Methodology**

The analysis was conducted in Python. Steps included:

1. Data cleaning – removal of missing values.

2. Descriptive statistics to summarise bat behaviours.

3. Chi-square test to assess the relationship between risk-taking and reward.

4. Independent samples T-test to compare bat feeding time with and without rats present.

5. Logistic regression to model risk-taking as a function of rat arrival time and night timing.

Results

**- Chi-square Test**: χ² = 435.99, p < 0.001.

→ Strong association between risk-taking and reward. Bats that take risks are significantly more likely to obtain food.

**- T-test:** t = -0.20, p = 0.838.

→ No significant difference in feeding time with vs without rats. Rat presence does not delay bat feeding.

**- Logistic Regression**:

- `hours\_after\_sunset`: coefficient = -0.06, p = 0.037 (significant).

- `seconds\_after\_rat\_arrival`: coefficient ≈ 0.0004, p = 0.211 (not significant).

→ Risk-taking decreases later at night, but rat arrival timing itself is not significant.

**Discussion**

The results suggest that bats perceive rats primarily as competitors for food rather than as predators.

- Risk-taking is strongly linked to reward, showing that challenging rats provides access to food.

- However, bats do not delay feeding behaviour when rats are present, indicating that they are not overly fearful.

- Night timing is more influential: bats are less likely to take risks later into the night, suggesting fatigue or reduced motivation.

Overall, rats affect competition but do not trigger strong avoidance or predator-like responses from bats.

**Conclusion**

- Rats influence competition outcomes by forcing bats into risk-taking to access food.

- No evidence that rats are perceived as direct predators, since avoidance was not significantly higher when rats were present.

- Time of night plays a more critical role in risk-taking behaviour.

**Code Explanation**

Code:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from scipy import stats

import statsmodels.api as sm

import statsmodels.formula.api as smf

Explanation:

* These are **libraries**:
  + pandas → handles tables (CSV files).
  + numpy → math & arrays.
  + matplotlib & seaborn → graphs.
  + scipy.stats → statistical tests (chi-square, t-test, etc.).
  + statsmodels → advanced stats models (like regression).

Code:

d1 = pd.read\_csv("C:\\Users\\Administrator\\Desktop\\project\\data\\dataset1.csv")

d2 = pd.read\_csv("C:\\Users\\Administrator\\Desktop\\project\\data\\dataset2.csv")

Explanation:

* Load both datasets into memory.
* d1 = bat landings data.
* d2 = rat arrivals data.

Code:

print("Dataset1 shape:", d1.shape)

print("Dataset2 shape:", d2.shape)

Explanation:

Prints how many rows & columns each dataset has.

Example: (866, 12) means 866 rows, 12 columns.

Code:

print(d1.info())

print(d2.info())

Explanation:

Shows details about each dataset (column names, data types, missing values).

Code:

d1 = d1.dropna()

d2 = d2.dropna()

Explanation:

Removes rows with missing values.

Because “dirty” or incomplete rows can break analysis.

Code:

print("\nDescription of  Dataset1:\n", d1.describe())

print("\nDescription of Dataset2:\n", d2.describe())

Explanation:

Shows summary stats (mean, min, max, std deviation) for all numeric columns.

Code:

risk\_counts = d1['risk'].value\_counts()

print("\nRisk behaviour counts:\n", risk\_counts)

Explanation:

Counts how many bats were **risk-takers (1)** vs **avoiders (0)**.

Code:

sns.countplot(x="risk", data=d1)

plt.title("Bat Risk-taking vs Avoidance")

plt.savefig("output/figures/risk\_counts.png")

plt.close()

Explanation:

Draws a bar chart of risk-taking vs avoidance.

Saves the image in your output folder.

Code:

sns.boxplot(x="risk", y="bat\_landing\_for\_food", data=d1)

plt.title("How quickly bats approach food (by risk behaviour)")

plt.savefig("output/figures/landing\_time.png")

plt.close()

Explanation:

Boxplot → compares how quickly bats start eating depending on risk behaviour.

Code:

sns.scatterplot(x="rat\_arrival", y="bat\_landing", data=d2)

plt.title("Rat arrivals vs Bat landings")

plt.savefig("output/figures/rat\_vs\_bat.png")

plt.close()

Explanation:

Scatterplot → shows if more rats = fewer bat landings (competition).

Code:

contingency = pd.crosstab(d1['risk'], d1['reward'])

chi2, p, dof, expected = stats.chi2\_contingency(contingency)

Explanation:

Makes a **contingency table** { *A contingency table is just a table that shows the counts (frequencies) of two categorical variables, side by side. It’s used when you want to see if two variables are related. This table is the basis for the Chi-square test.}* between risk (0/1) and reward (0/1).

Runs **Chi-square test** → checks if risk-taking and reward are related.

Code:

with\_rats = d1[d1['seconds\_after\_rat\_arrival'] > 0]['bat\_landing\_to\_food']

without\_rats = d1[d1['seconds\_after\_rat\_arrival'] == 0]['bat\_landing\_to\_food']

t\_stat, p\_val = stats.ttest\_ind(with\_rats, without\_rats)

Explanation:

Splits bat landings into **with rats present** vs **without rats**.

Runs a **T-test** to check if the time to approach food is different.

Code:

formula = "risk ~ seconds\_after\_rat\_arrival + hours\_after\_sunset"

model = smf.logit(formula, data=d1)

result = model.fit()

print(result.summary())

print("\nLogistic Regression Result:\n", model.summary())

Explanation:

Runs a **logistic regression**:

* Dependent variable: risk (0/1).
* Predictors: seconds\_after\_rat\_arrival and hours\_after\_sunset.

Tells us if rat arrival time or time of night predicts risk-taking.

Code:

f= open("outputs/tables/results.txt", "w")

f.write("Chi-square Test between risk and reward:\n")

f.write(f"Chi2={chi2}, p={p}\n\n")

f.write("T-test for landing time:\n")

f.write(f"t={t\_stat}, p={p\_val}\n\n")

f.write(str(model.summary()))

Explanation:

Writes all results into a text file for later reporting.

**Why used chi square?**

**1. Chi-square Test (Risk vs Reward)**

**What it does:**  
Chi-square checks if there’s a **relationship between two categorical variables** (like Yes/No, 0/1).

**In our case:**

* Variable 1 = risk (0 = avoidance, 1 = risk-taking)
* Variable 2 = reward (0 = no food, 1 = got food)

We asked: **“Is risk-taking linked to getting food?”**

* If bats take risks, do they get rewarded more often?
* Chi-square is perfect here because both are yes/no type variables.

**Result:** Significant → Risk-takers got food more often.  
So, **risk behaviour matters**.

Why used T- test?

**2. T-test (Landing-to-food time with vs without rats)**

**What it does:**  
T-test compares the **average of two groups** to see if they are different.

**In our case:**

* Group 1 = bats that landed when rats were already present.
* Group 2 = bats that landed when no rats were there.
* Variable = bat\_landing\_to\_food (time in seconds).

We asked: **“Do bats delay eating if rats are around?”**

* If yes, we’d expect a longer landing-to-food time when rats are present.
* T-test checks if those two averages are significantly different.

**Result:** Not significant → Bats do **not** delay feeding when rats are around.  
So, they don’t seem “scared” of rats.

Why used logical regression?

**3. Logistic Regression (Risk predicted by rat timing & night hours)**

**What it does:**  
Logistic regression predicts the probability of a **binary outcome** (0/1) based on other variables.  
It also tells us which predictors matter more.

**In our case:**

* Outcome = risk (0 = avoid, 1 = risk-taking).
* Predictors =
  + seconds\_after\_rat\_arrival (how soon after rats appeared the bat landed)
  + hours\_after\_sunset (time of night).

We asked: **“Can we predict risk-taking based on rat arrival or time of night?”**

* Does arriving right after rats make bats more cautious?
* Do bats behave differently earlier vs later at night?

**Result:**

* Rat arrival time → not significant.
* Time of night → significant (later in the night, bats take fewer risks).

So, **time of night affects risk behaviour more than rat presence**.

**Easy “Student Voice” Explanation (what you can say)**

* *“We used a chi-square test because we had two yes/no variables — risk and reward — and we wanted to check if they are related. It showed that risk-taking is strongly linked to getting food.”*
* *“We used a t-test to compare feeding time with and without rats. This was to see if bats delay eating when rats are around. The result was not significant, meaning rats don’t actually slow them down.”*
* *“We used logistic regression because our outcome, risk-taking, is also yes/no. Regression let us test if rat presence or time of night predicts risk-taking. The result showed that night-time matters, but rats themselves don’t.”*