```
# ================================
# Data Cleaning for Bat vs Rat Project
# ================================

import pandas as pd
import numpy as np

# Load datasets (update with your correct file paths)
dataset1 = pd.read_csv("/content/dataset1.csv")
dataset2 = pd.read_csv("/content/dataset2.csv")


# 1. Inspect datasets

print("Dataset1 Info:")
print(dataset1.info())
print("\nDataset1 Head:")
print(dataset1.head())

print("\nDataset2 Info:")
print(dataset2.info())
print("\nDataset2 Head:")
print(dataset2.head())


# 2. Check for missing values

print("\nMissing values in Dataset1:")
print(dataset1.isna().sum())

print("\nMissing values in Dataset2:")
print(dataset2.isna().sum())


# 3. Remove duplicates

dataset1 = dataset1.drop_duplicates()
dataset2 = dataset2.drop_duplicates()
```

```
# 4. Convert columns to correct types
# (Adjust column names if needed)
# Convert time columns with day-first format
time_cols1 = ["start_time", "rat_period_start", "rat_period_end", "sunset_time"]
for col in time_cols1:
    if col in dataset1.columns:
        dataset1[col] = pd.to_datetime(dataset1[col], errors="coerce", dayfirst=True)

if "time" in dataset2.columns:
    dataset2["time"] = pd.to_datetime(dataset2["time"], errors="coerce", dayfirst=True)

# Convert time columns to datetime
#time_cols1 = ["start_time", "rat_period_start", "rat_period_end", "sunset_time"]
#for col in time_cols1:
 #   if col in dataset1.columns:
  #      dataset1[col] = pd.to_datetime(dataset1[col], errors="coerce")

#if "time" in dataset2.columns:
#    dataset2["time"] = pd.to_datetime(dataset2["time"], errors="coerce")##

# Ensure categorical variables are categorical
cat_cols1 = ["habit", "risk", "reward", "month", "season"]
for col in cat_cols1:
    if col in dataset1.columns:
        dataset1[col] = dataset1[col].astype("category")

cat_cols2 = ["month"]
for col in cat_cols2:
    if col in dataset2.columns:
        dataset2[col] = dataset2[col].astype("category")


# 5. Handle missing values

# Example: Fill missing numerical values with median
dataset1 = dataset1.fillna(dataset1.median(numeric_only=True))
dataset2 = dataset2.fillna(dataset2.median(numeric_only=True))

# Example: Drop rows where essential categorical values are missing
dataset1 = dataset1.dropna(subset=["risk", "reward", "season"])
dataset2 = dataset2.dropna(subset=["month"])
```

```
# 6. Save cleaned datasets

dataset1.to_csv("dataset1_cleaned.csv", index=False)
dataset2.to_csv("dataset2_cleaned.csv", index=False)

print("\nData cleaning completed! Cleaned files saved as 'dataset1_cleaned.csv' and 'dataset2_cleaned.csv'.")
```

```
habit                          41
```

```
habit                       41
rat_period_start             0
rat_period_end               0
seconds_after_rat_arrival    0
risk                         0
reward                       0
month                        0
sunset_time                  0
hours_after_sunset           0
season                       0
dtype: int64

Missing values in Dataset2:
time                  0
month                 0
hours_after_sunset    0
bat_landing_number    0
food_availability     0
rat_minutes           0
rat_arrival_number    0
dtype: int64

Data cleaning completed! Cleaned files saved as 'dataset1_cleaned.csv' and 'dataset2_cleaned.csv'.
```

```
pd.to_datetime(dataset1[col], errors="coerce", dayfirst=True)
```

	month
0	1970-01-01 00:00:00.000000000
1	1970-01-01 00:00:00.000000000
2	1970-01-01 00:00:00.000000000

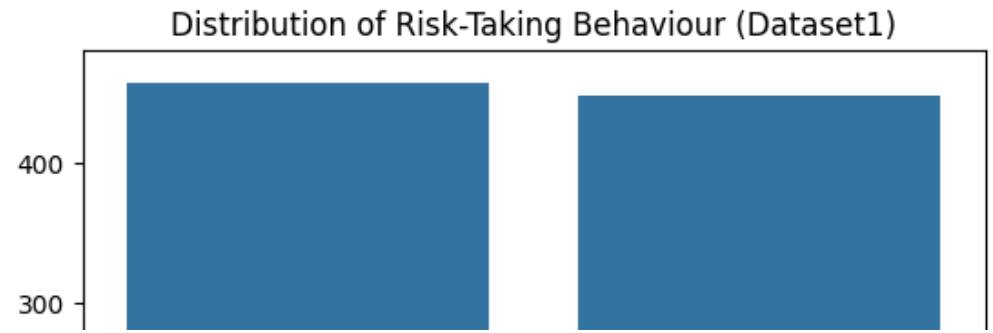
```
# ============================================
# Bat vs Rat Project - EDA & Statistical Analysis
# ============================================

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import chi2_contingency, ttest_ind, f_oneway
import statsmodels.api as sm

# ------------------------------
# Load cleaned datasets
# ------------------------------
dataset1 = pd.read_csv("dataset1_cleaned.csv")
dataset2 = pd.read_csv("dataset2_cleaned.csv")

# ------------------------------
# Exploratory Data Analysis (EDA)
# ------------------------------
print("\nDataset1 Summary:")
print(dataset1.describe(include="all"))
print("\nDataset2 Summary:")
print(dataset2.describe(include="all"))

# Risk-taking behaviour distribution
sns.countplot(x="risk", data=dataset1)
plt.title("Distribution of Risk-Taking Behaviour (Dataset1)")
plt.show()

# Reward vs Risk
sns.countplot(x="reward", hue="risk", data=dataset1)
plt.title("Reward Outcome vs Risk Behaviour (Dataset1)")
plt.show()
```

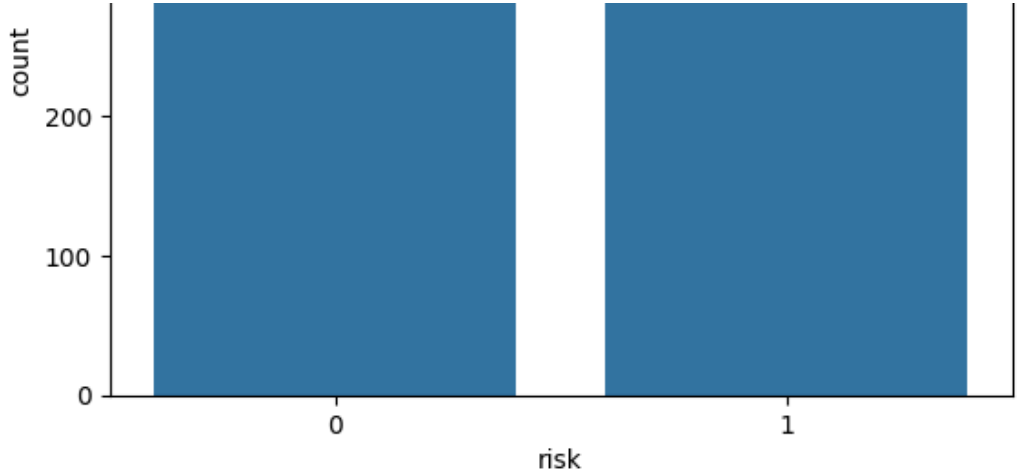
```
# Risk-taking across seasons
sns.countplot(x="season", hue="risk", data=dataset1)
plt.title("Risk-Taking Across Seasons (Dataset1)")
plt.show()

# Bat landings per month
sns.barplot(x="month", y="bat_landing_number", data=dataset2, ci=None)
plt.title("Bat Landings per Month (Dataset2)")
plt.show()

# Rat arrivals vs bat landings
sns.scatterplot(x="rat_arrival_number", y="bat_landing_number", data=dataset2)
plt.title("Rat Arrivals vs Bat Landings (Dataset2)")
plt.show()

# Correlation heatmap
sns.heatmap(dataset2.corr(numeric_only=True), annot=True, cmap="coolwarm")
plt.title("Correlation Heatmap (Dataset2)")
plt.show()

# ------------------------------
# Investigation A - Predator Perception
# ------------------------------
print("\n--- Investigation A ---")

# Chi-Square Test: Risk vs Habit
contingency = pd.crosstab(dataset1['risk'], dataset1['habit'])
chi2, p, dof, expected = chi2_contingency(contingency)
print("Chi-square test p-value (risk vs habit):", p)

# Logistic Regression: Predictors of Risk
X = dataset1[["seconds_after_rat_arrival", "hours_after_sunset", "season"]]
y = dataset1["risk"]
X = sm.add_constant(X)
logit_model = sm.Logit(y, X).fit()
print(logit_model.summary())

# ------------------------------
# Investigation B - Seasonal Changes
# ------------------------------
print("\n--- Investigation B ---")
```

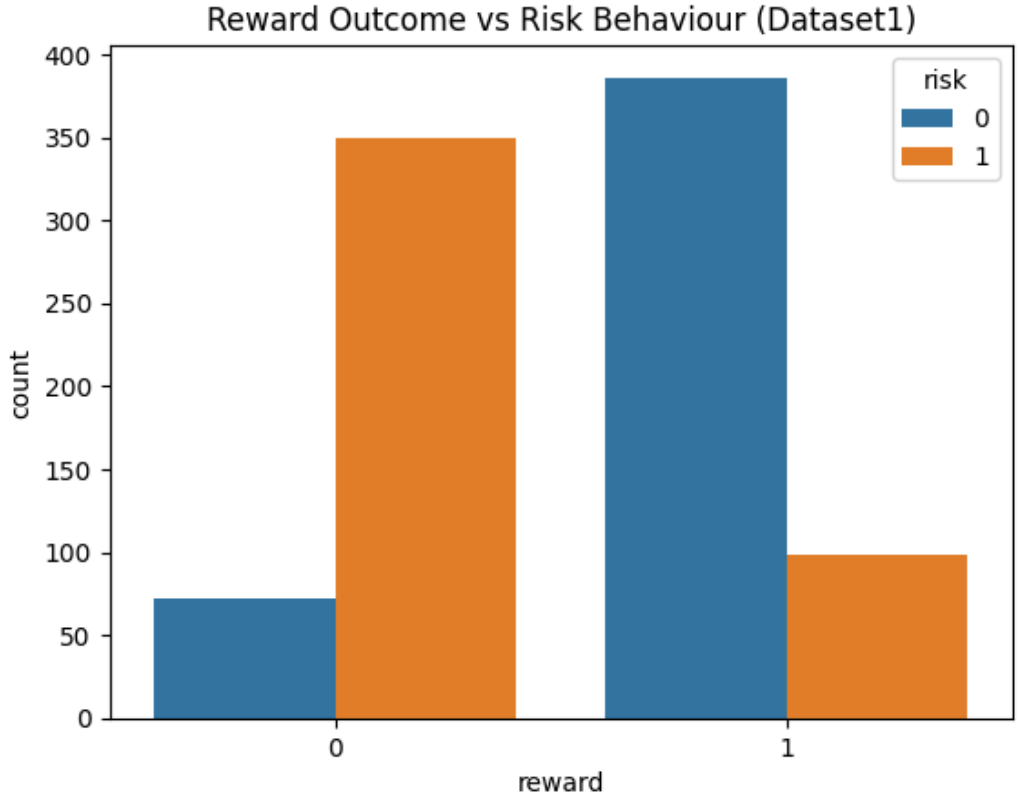
```
# T-test (Risk by Season: Winter=0 vs Spring=1)
season0 = dataset1[dataset1["season"]==0]["risk"]
season1 = dataset1[dataset1["season"]==1]["risk"]
t_stat, p_val = ttest_ind(season0, season1, equal_var=False)
print("T-test Risk (Winter vs Spring) p-value:", p_val)

# ANOVA example (if >2 seasons available)
anova = f_oneway(*[dataset1[dataset1["season"]==s]["risk"] for s in dataset1["season"].unique()])
print("ANOVA Risk by Season p-value:", anova.pvalue)

# Seasonal trend in bat landings
sns.barplot(x="month", y="bat_landing_number", data=dataset2, errorbar=None)
plt.title("Monthly Trends in Bat Landings (Dataset2)")
plt.show()


print("\nAnalysis Complete ✅")
```

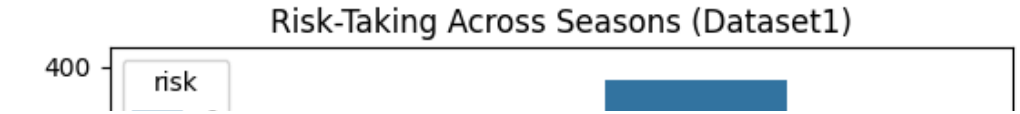
```
Dataset1 Summary:
                start_time  bat_landing_to_food habit     rat_period_start  \
count                  906           906.000000   865                  906
unique                 628                  NaN    81                  268
top    2018-01-28 20:05:00                  NaN  fast  2018-04-26 22:25:00
freq                     6                  NaN   245                   29
mean                   NaN            11.720544   NaN                  NaN
std                    NaN            27.658777   NaN                  NaN
min                    NaN             0.010238   NaN                  NaN
25%                    NaN             1.000000   NaN                  NaN
50%                    NaN             4.000000   NaN                  NaN
75%                    NaN            11.750000   NaN                  NaN
max                    NaN           443.000000   NaN                  NaN

            rat_period_end  seconds_after_rat_arrival        risk  \
count                  906                 906.000000  906.000000
unique                 268                        NaN         NaN
top    2018-04-26 22:36:00                        NaN         NaN
freq                    29                        NaN         NaN
mean                   NaN                 282.786976    0.494481
std                    NaN                 241.092545    0.500246
min                    NaN                   0.000000    0.000000
25%                    NaN                  89.250000    0.000000
50%                    NaN                 206.000000    0.000000
75%                    NaN                 447.250000    1.000000
max                    NaN                 949.000000    1.000000

           reward       month          sunset_time  hours_after_sunset  \
count   906.000000  906.000000                  906          906.000000
unique         NaN         NaN                   65                 NaN
top            NaN         NaN  2018-04-26 19:17:00                 NaN
freq           NaN         NaN                  101                 NaN
mean      0.534216    3.800221                  NaN            5.532579
std       0.499103    1.199834                  NaN            2.415383
min       0.000000    0.000000                  NaN           -0.261667
25%       0.000000    4.000000                  NaN            3.775069
50%       1.000000    4.000000                  NaN            5.627083
75%       1.000000    5.000000                  NaN            7.406250
max       1.000000    5.000000                  NaN           12.091944

           season
count   906.000000
unique         NaN
top            NaN
```

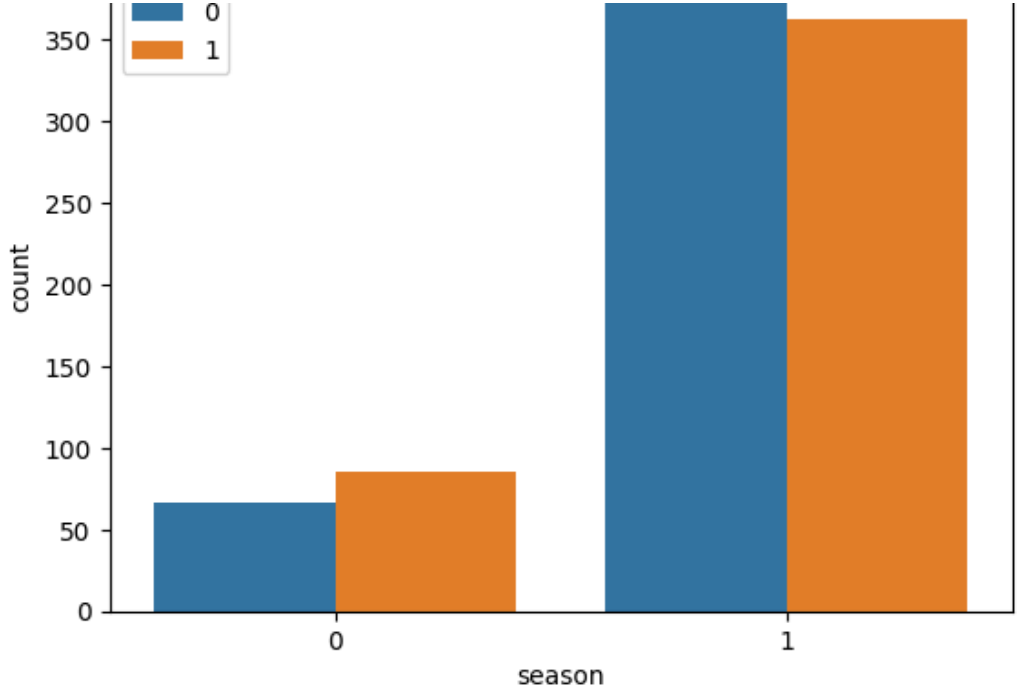
```
freq           NaN
mean      0.833333
std       0.372884
min       0.000000
25%       1.000000
50%       1.000000
75%       1.000000
max       1.000000

Dataset2 Summary:
                        time        month  hours_after_sunset  \
count                   2123  2123.000000         2123.000000
unique                  2123          NaN                 NaN
top     2018-06-01 05:41:00          NaN                 NaN
freq                       1          NaN                 NaN
mean                     NaN     3.083844            5.265426
std                      NaN     1.642261            4.076188
min                      NaN     0.000000           -2.000000
25%                      NaN     2.000000            2.000000
50%                      NaN     4.000000            5.000000
75%                      NaN     4.000000            8.500000
max                      NaN     6.000000           13.500000

        bat_landing_number  food_availability  rat_minutes  rat_arrival_number
count          2123.000000        2123.000000  2123.000000         2123.000000
unique                 NaN                NaN          NaN                 NaN
top                    NaN                NaN          NaN                 NaN
freq                   NaN                NaN          NaN                 NaN
mean             32.083373           2.445874     1.994442            0.444654
std              25.614431           1.218353     6.793397            1.019195
min               0.000000           0.000000     0.000000            0.000000
25%              11.000000           1.962206     0.000000            0.000000
50%              27.000000           2.951877     0.000000            0.000000
75%              48.000000           3.105873     0.158333            1.000000
max             178.000000           4.000000   120.000000           17.000000
```



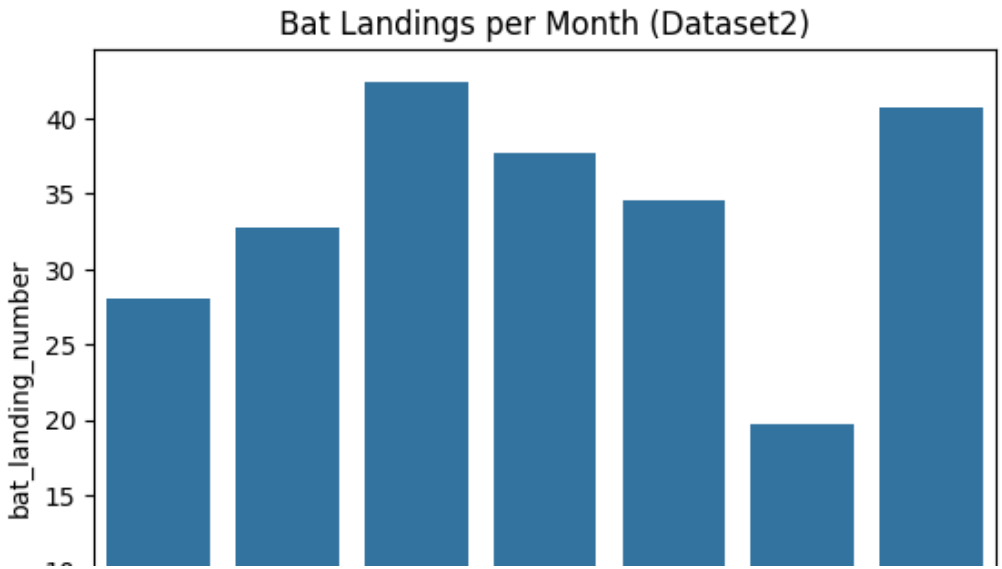

count
200
100
0
0
1
risk

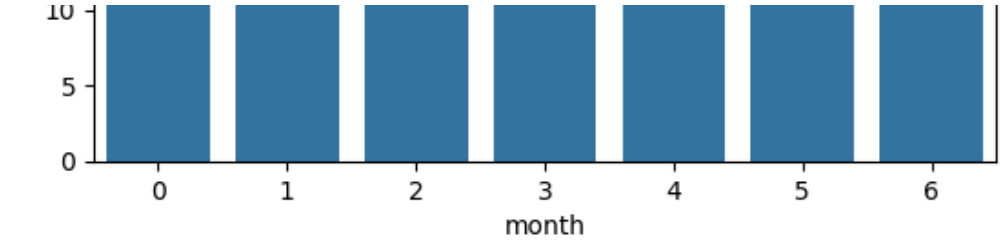

Reward Outcome vs Risk Behaviour (Dataset1)
risk
0
1
count
400
350
300
250
200
150
100
50
0
0
1
reward

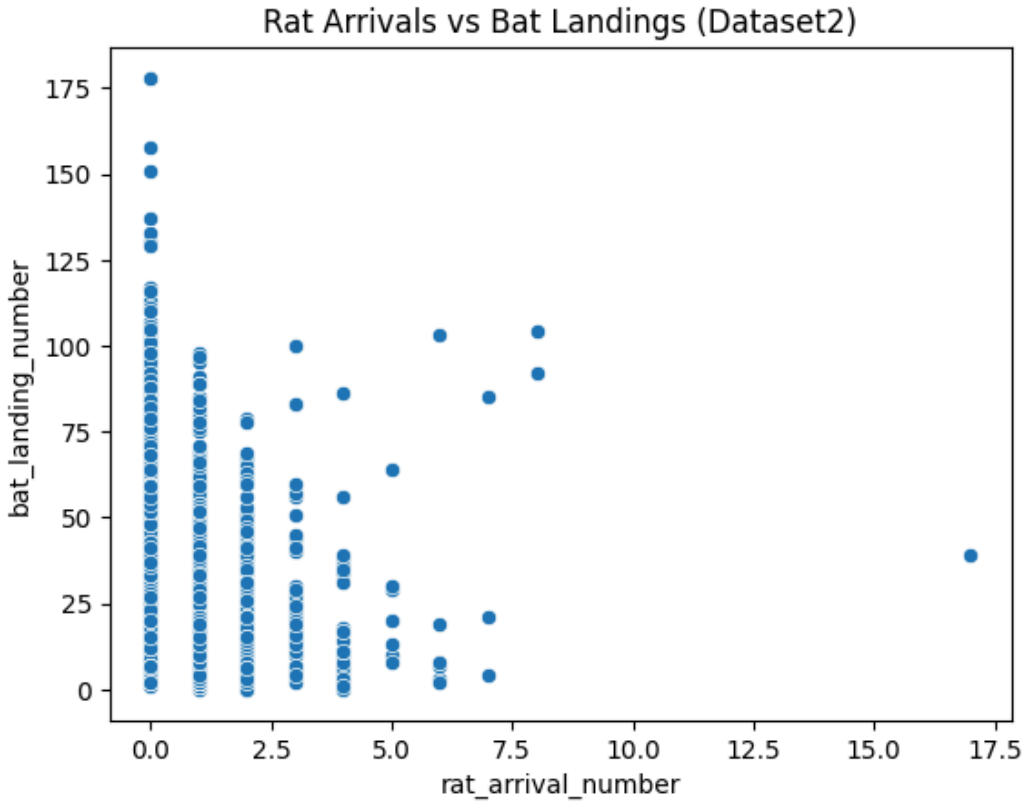

Risk-Taking Across Seasons (Dataset1)
400
risk

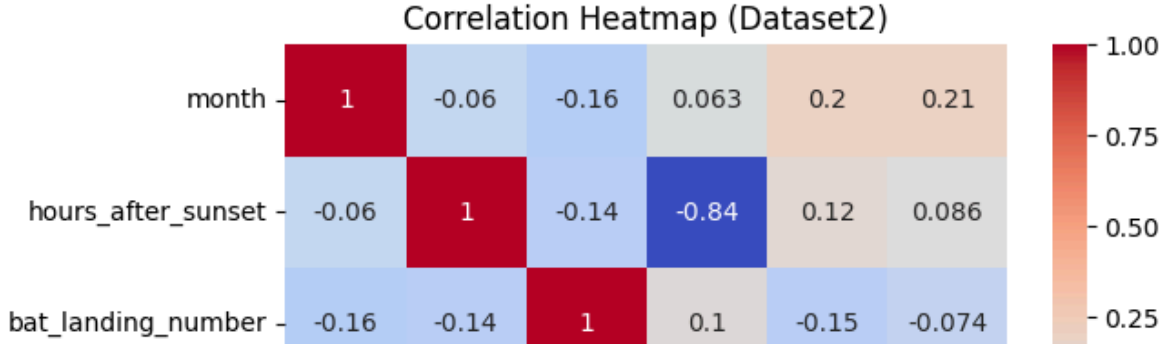


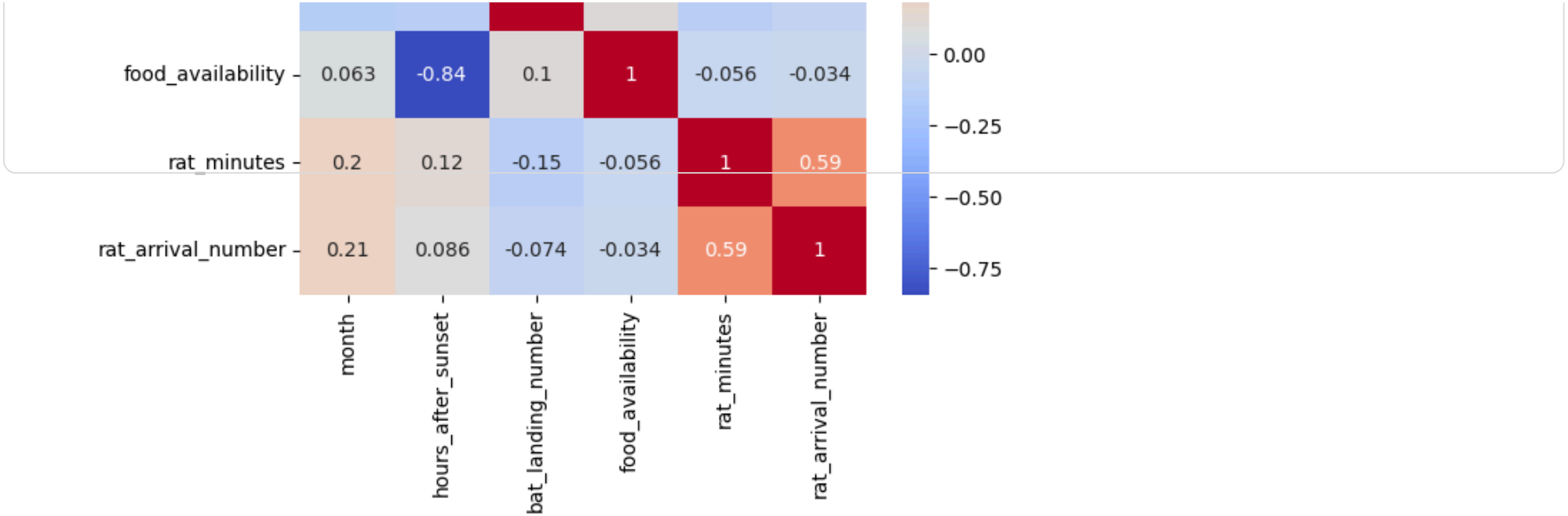

```
/tmp/ipython-input-384532454.py:41: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x="month", y="bat_landing_number", data=dataset2, ci=None)
```




10
5
0
0
1
2
3
4
5
6
month


Rat Arrivals vs Bat Landings (Dataset2)
bat_landing_number
175
150
125
100
75
50
25
0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
rat_arrival_number


Correlation Heatmap (Dataset2)
month
hours_after_sunset
bat_landing_number
1
-0.06
-0.16
0.063
0.2
0.21
-0.06
1
-0.14
-0.84
0.12
0.086
-0.16
-0.14
1
0.1
-0.15
-0.074
1.00
0.75
0.50
0.25




```
--- Investigation A ---
Chi-square test p-value (risk vs habit): 5.040138482485281e-132
Optimization terminated successfully.
         Current function value: 0.688802
         Iterations 4
                           Logit Regression Results
==============================================================================
Dep. Variable:                   risk   No. Observations:                  906
Model:                          Logit   Df Residuals:                      902
Method:                           MLE   Df Model:                            3
Date:                Tue, 09 Sep 2025   Pseudo R-squ.:                0.006182
Time:                        12:00:42   Log-Likelihood:                -624.05
converged:                       True   LL-Null:                       -627.94
Covariance Type:            nonrobust   LLR p-value:                   0.05116
================================================================================================
                                   coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------------------------
const                            0.4814      0.231      2.086      0.037       0.029       0.934
seconds_after_rat_arrival        0.0003      0.000      1.071      0.284      -0.000       0.001
hours_after_sunset              -0.0520      0.028     -1.869      0.062      -0.107       0.003
season                          -0.3609      0.182     -1.982      0.047      -0.718      -0.004
================================================================================================
```

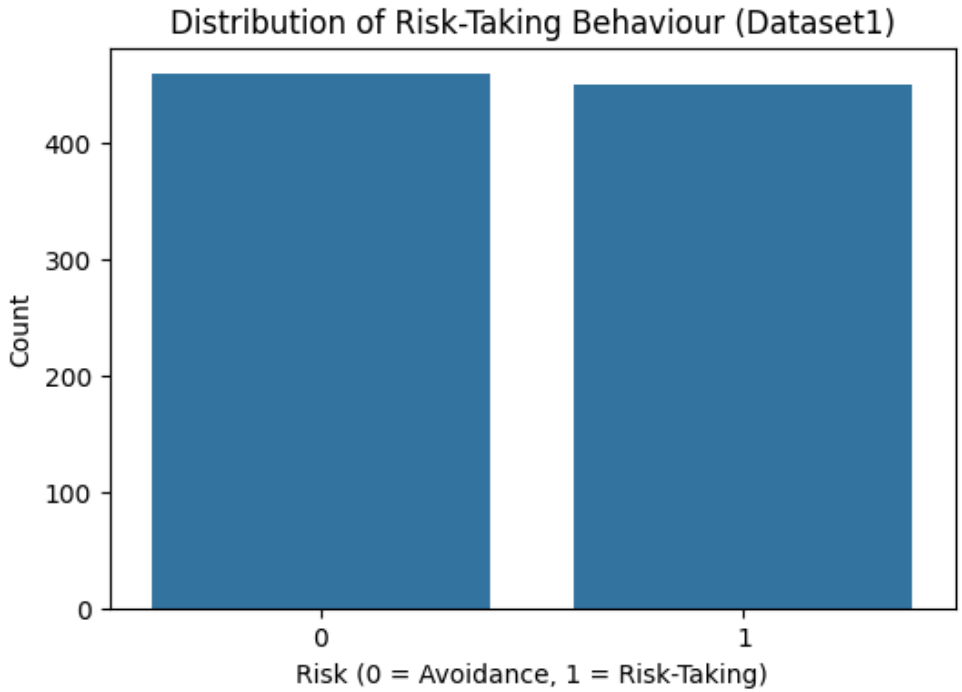
```
--- Investigation B ---
```

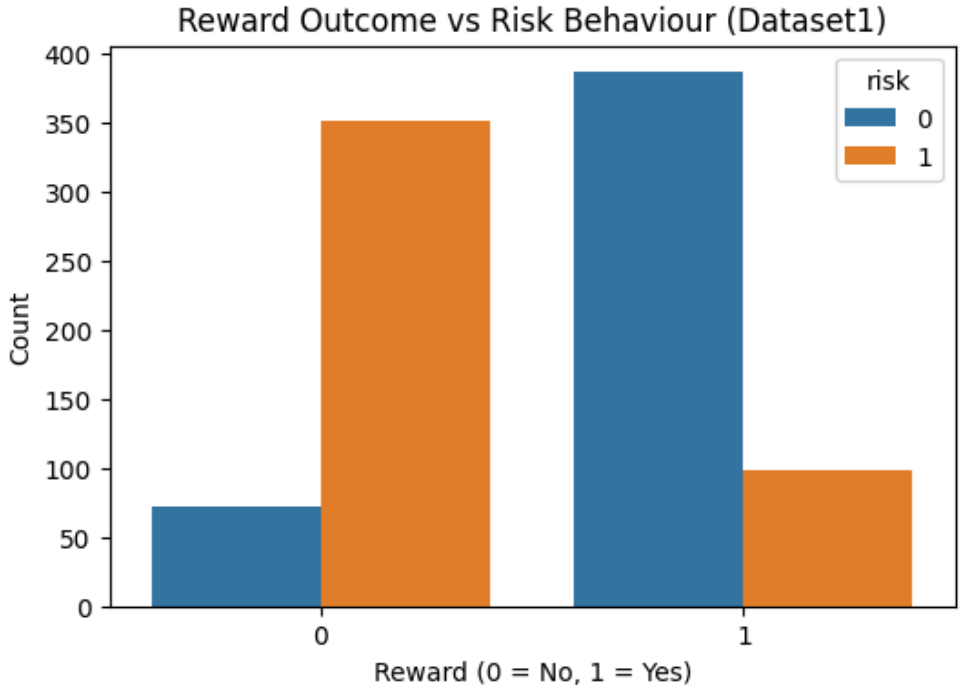
Start coding or generate with AI.

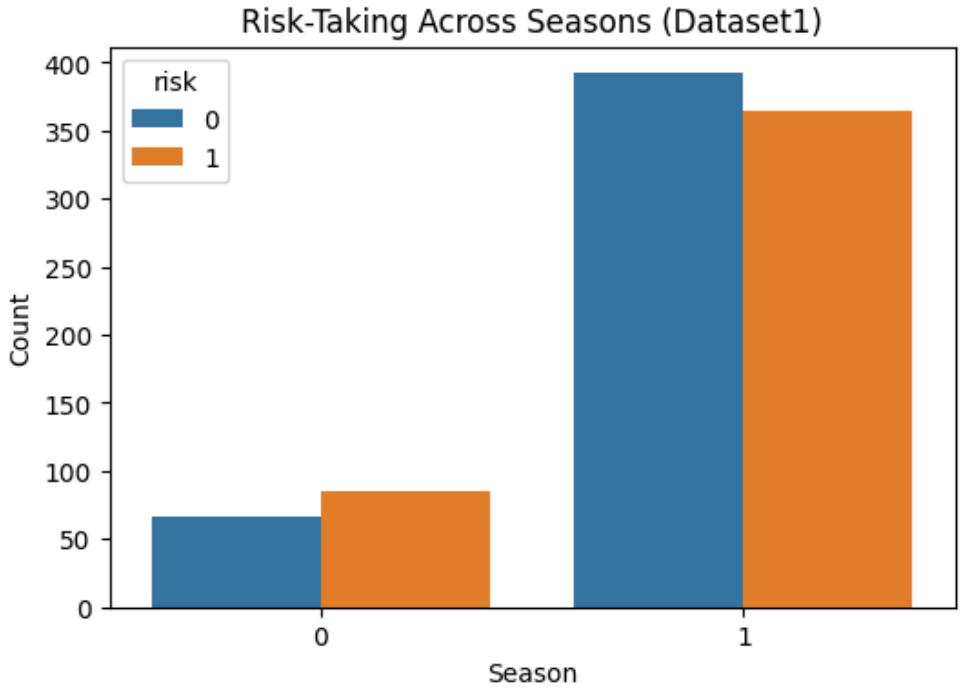
Monthly Trends in Bat Landings (Dataset2)

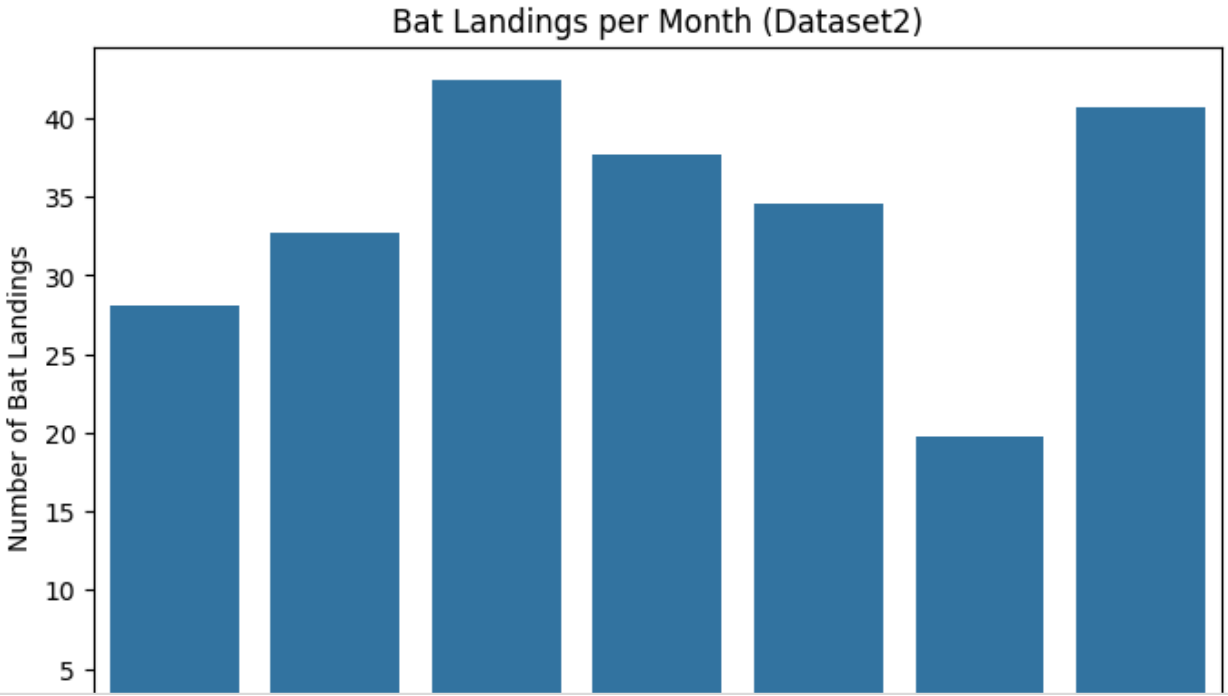
```
# ============================================
# Bat vs Rat Project - Visualisations
# ============================================

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load your cleaned datasets
dataset1 = pd.read_csv("/content/dataset1.csv")
dataset2 = pd.read_csv("/content/dataset2.csv")

# ------------------------------
# 1. Risk-taking behaviour distribution (Dataset1)
# ------------------------------
plt.figure(figsize=(6,4))
sns.countplot(x="risk", data=dataset1)
plt.title("Distribution of Risk-Taking Behaviour (Dataset1)")
plt.xlabel("Risk (0 = Avoidance, 1 = Risk-Taking)")
plt.ylabel("Count")
plt.show()

# ------------------------------
# 2. Reward vs Risk (Dataset1)
# ------------------------------
plt.figure(figsize=(6,4))
sns.countplot(x="reward", hue="risk", data=dataset1)
plt.title("Reward Outcome vs Risk Behaviour (Dataset1)")
plt.xlabel("Reward (0 = No, 1 = Yes)")
plt.ylabel("Count")
plt.show()

# ------------------------------
# 3. Risk-taking across seasons (Dataset1)
# ------------------------------
plt.figure(figsize=(6,4))
sns.countplot(x="season", hue="risk", data=dataset1)
plt.title("Risk-Taking Across Seasons (Dataset1)")
```

```
plt.xlabel("Season")
plt.ylabel("Count")
plt.show()

# ------------------------------
# 4. Bat landings per month (Dataset2)
# ------------------------------
plt.figure(figsize=(8,5))
sns.barplot(x="month", y="bat_landing_number", data=dataset2, errorbar=None)
plt.title("Bat Landings per Month (Dataset2)")
plt.xlabel("Month")
plt.ylabel("Number of Bat Landings")
plt.show()

# ------------------------------
# 5. Rat arrivals vs Bat landings (Dataset2)
# ------------------------------
plt.figure(figsize=(6,4))
sns.scatterplot(x="rat_arrival_number", y="bat_landing_number", data=dataset2)
plt.title("Rat Arrivals vs Bat Landings (Dataset2)")
plt.xlabel("Rat Arrivals")
plt.ylabel("Bat Landings")
plt.show()

# ------------------------------
# 6. Correlation heatmap (Dataset2)
# ------------------------------
plt.figure(figsize=(8,6))
corr = dataset2.corr(numeric_only=True)
sns.heatmap(corr, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Correlation Heatmap (Dataset2)")
plt.show()
```


Distribution of Risk-Taking Behaviour (Dataset1)
Count
0
100
200
300
400
0
1
Risk (0 = Avoidance, 1 = Risk-Taking)


Reward Outcome vs Risk Behaviour (Dataset1)
Count
0
50
100
150
200
250
300
350
400
0
1
Reward (0 = No, 1 = Yes)
risk
0
1


Risk-Taking Across Seasons (Dataset1)
risk
0
1
Count
400
350
300
250
200
150
100
50
0
0
1
Season


Bat Landings per Month (Dataset2)
Number of Bat Landings
40
35
30
25
20
15
10
5