Statistical Analysis and Parameter Estimation

Submitted By:

Gopi Charan

Roll Number: MA23BTECH11015

Varshith

Roll Number: MA23BTECH11014

Prajith

Roll Number: MA23BTECH11019

Course:

Applied Statistics

Table of Contents

1 Datasets

Here is a brief description of datasets. However, these datasets are attached to this PDF in the Google Classroom page (.csv files).

1.1 Paris 2024 Olympic Medals

This dataset comprises Olympic Medal counts and additional data like Population, GDP, and HDI of 89 participating countries that have won at least one medal in the Paris Olympics 2024.

1.2 Food and Nutrition

This dataset contains the information of 1698 individuals, like age, gender, height, weight, daily intake of dietary nutrients like sodium, carbohydrates, proteins, fats, etc, including their preferences for meals and snacks, and diseases caused.

1.3 Indian Cricket Matches

This dataset contains summary statistics for 243 international cricket matches played by the Indian team. Details related to the match outcomes, toss decisions, batting/bowling choices, scores, and opponents.

2 Parameter Estimation for Gamma Distribution Using MoM and MLE

We shall be using Olympic Medals dataset for this part. Note that all the values in the dataset are positive. Suppose we model the data using a Gamma(a,b) distribution. For a Gamma distribution with shape parameter a and rate parameter b, the mean and variance are:

$$\mathbb{E}[X] = \frac{a}{b}, \quad \text{Var}(X) = \frac{a}{b^2}$$

We analyze variance, standard deviation, mean of the data.

```
import pandas as pd
import numpy as np

variance = df["Total Medals"].var()
mean_medals = df["Total Medals"].mean()
```

```
print("Variance:", variance)
print("Mean:",mean_medals)
```

Output:

```
Variance: 395.125
Mean: 11.6179
```

2.1 Method of Moments

For Gamma-Distribution we know that,

$$\mathbb{E}[X] = \frac{a}{b}, \quad \text{Var}(X) = \frac{a}{b^2}$$

Using method of moments we get the following equations:

$$\texttt{mean\_medals} = \bar{x} = \frac{a}{b}$$
$$\texttt{variance} = s^2 = \frac{a}{b^2}$$

Solving the equations, we get:

$$b = \frac{\bar{x}}{s^2}$$
$$a = \bar{x}b = \frac{\bar{x}^2}{s^2}$$

Hence,

$$\texttt{MoM Estimates:} \quad \hat{a}_{\texttt{MoM}} = \frac{\bar{x}^2}{s^2} \approx 0.3447, \quad \hat{b}_{\texttt{MoM}} = \frac{\bar{x}}{s^2} \approx 0.0295$$

2.2 Maximum Likelihood Estimates

Let $X_1, X_2, \ldots, X_n \overset{iid}{\sim} \text{Gamma}(a, b)$, with density function :

$$f(x|a,b) = \frac{b^a}{\Gamma(a)} x^{a-1} e^{-bx}, \quad x > 0$$

The log-likelihood function for observed data $x_1, \ldots, x_n$ is:

$$\ell(a,b) = \sum_{i=1}^{n} \log f(x_i|a,b) = (a-1)\sum_{i=1}^{n} \log x_i - \sum_{i=1}^{n} bx_i + na \log b - n \log \Gamma(a)$$

Maximizing $\ell(a, b)$ and Partial Derivative w.r.t. a, b

$$\frac{\partial \ell}{\partial b} = b^2 \left(\sum_{i=1}^{n} x_i - \frac{na}{b} \right)$$

Setting $\frac{\partial \ell}{\partial b} = 0$, we get:

$$\hat{b} = \frac{1}{\sum_{i=1}^{n} x_i} na = \frac{a}{\bar{x}}$$

Substituting b and setting $\frac{\partial \ell}{\partial a} = 0$, we get:

$$\log a - \frac{\Gamma'(a)}{\Gamma(a)} = \log \bar{x} - \frac{1}{n} \sum_{i=1}^{n} \log x_i$$

We cannot solve for the exact value of a rather we can approximate it using **Newton-Raphson Method**. Note that $\frac{\Gamma'(a)}{\Gamma(a)}$ is known as **di-gamma** function and represented by $\Psi(a)$.

```
import numpy as np
import pandas as pd
from scipy.special import psi, polygamma

def newton_gamma_mle(data, max_iter=100, tol=1e-6):
    x_bar = data["Total Medals"].mean()
    s = np.log(x_bar) - np.mean(np.log(data["Total Medals"]))
    a = 0.5 / s  # initial guess

    for _ in range(max_iter):
        numerator = np.log(a) - psi(a) - s
        denominator = 1/a - polygamma(1, a)
        a_new = a - numerator / denominator
        if abs(a_new - a) < tol:
            break
        a = a_new
    return a
data = pd.read_csv('olympics_data.csv')
a_est = newton_gamma_mle(data)
b_est = a_est / np.mean(data["Total Medals"])

print("Shape (a):", a_est)
print("Rate (b):", b_est)
```

Output:

```
Shape (a): 0.7616447191533221
Scale (b): 0.06555742747064378
```

3 Confidence Interval for Variance Under Normality Assumption

Assume a sample $X_1, X_2, \ldots, X_n$ drawn from a normal distribution $\mathcal{N}(\mu, \sigma^2)$, where both μ and σ^2 are unknown. The $100(1-\alpha)\%$ confidence interval for the variance σ^2 is given by:

$$\left(\frac{(n-1)s^2}{\chi^2_{1-\alpha/2,n-1}}, \frac{(n-1)s^2}{\chi^2_{\alpha/2,n-1}} \right)$$

Where:

- $s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X})^2$ is the sample variance,
- $\bar{X}$ is the sample mean,
- $\chi^2_{\alpha/2,n-1}$ and $\chi^2_{1-\alpha/2,n-1}$ are critical values from the chi-square distribution with $n-1$ degrees of freedom.

For 95% confidence interval, $100(1-\alpha) = 95 \implies \alpha = 0.05$. On substituting the values: $n = 89, s^2 = 395.125, \alpha = 0.05$,
Lower bound $= \frac{(88)(395.125)}{\chi^2_{0.975,88}} = 310.3943$, Upper bound $= \frac{(88)(395.125)}{\chi^2_{0.025,88}} = 520.789$

So, the 95% confidence interval for variance is: $(310.3943, 520.789)$

4 Confidence Interval for the Difference of Means Between two Populations

We shall be using `Food and Nutrition` dataset for this part. The non-overlapping population sets which we are going to consider are `Male` and `Female` and the attribute which we are taking is `Calorie Intake`. Let $M_1, M_2, \ldots, M_n$ be a sample of size n from a normal population having mean μ_1 and variance σ_1^2, and let $F_1, \ldots, F_m$ be a sample of size m from a different normal population having mean μ_2 and variance σ_2^2.

Since we are modeling these populations as independent normal distributions with unknown means and variances, let

$$\bar{M} = \sum_{i=1}^{n} \frac{M_i}{n}, \quad S_1^2 = \frac{\sum_{i=1}^{n} (M_i - \bar{M})^2}{n-1}$$

$$\bar{F} = \sum_{i=1}^{n} \frac{F_i}{n}, \quad S_2^2 = \frac{\sum_{i=1}^{n} (F_i - \bar{F})^2}{n-1}$$

$$S_p^2 = \frac{(n-1)S_1^2 + (m-1)S_2^2}{n+m-2}$$

We know that $100(1-\alpha)$ percent confidence interval for $\mu_1 - \mu_2$ is given by,

$$\left(\bar{M} - \bar{F} - t_{\alpha/2,\, n+m-2} \cdot S_p \sqrt{\frac{1}{n} + \frac{1}{m}},\ \bar{M} - \bar{F} + t_{\alpha/2,\, n+m-2} \cdot S_p \sqrt{\frac{1}{n} + \frac{1}{m}} \right)$$

```
import pandas as pd
import numpy as np
from scipy.stats import t

data = pd.read_csv("Food_and_Nutrition__.csv")
male_calories = data[data["Gender"] == "Male"]["Calories"]
female_calories = data[data["Gender"] == "Female"]["Calories"]

n1 = len(male_calories)
n2 = len(female_calories)
mean1 = np.mean(male_calories)
mean2 = np.mean(female_calories)
var1 = np.var(male_calories, ddof=1)
var2 = np.var(female_calories, ddof=1)
alpha = 0.05

combined_variance = (((n1 - 1) * var1 + (n2 - 1) * var2) / (n1 + n2 - 2))
combined_std_dev = np.sqrt(combined_variance)
standard_error = combined_std_dev * np.sqrt((1 / n1) + (1 / n2))
n = n1 + n2 - 2
t_critical = t.ppf(1-alpha/2, n)

mean_diff = mean1 - mean2
```

```
margin_of_error = t_critical * standard_error
lower_bound = mean_diff - margin_of_error
upper_bound = mean_diff + margin_of_error

print("95% Confidence Interval for the difference in Means:")
print(f"Lower Bound: {lower_bound:.2f}")
print(f"Upper Bound: {upper_bound:.2f}")
```

Output:

```
95% Confidence Interval for the difference in Means:
Lower Bound: 205.44
Upper Bound: 311.39
```

5 Hypothesis Testing on a Bernoulli Distributed Binary Response Variable

We shall be using the result of the Indian Cricket team as the data set. We model the result data as a Bernoulli $B(n, p)$ distribution with parameters n is the total number of matches and p is the true probability of India winning a match. The Null Hypothesis H_0 and the alternative Hypothesis H_1 are

$$H_0 : p \leq 0.5 \qquad H_1 : p > 0.5$$

The Null Hypothesis probability is $p_0 = 0.5$ and the sample proportion of data is $\hat{p} = \frac{X}{n}$ where X is the number of Successful trials of the results and n is the number of trials in the data. With the level of Significance ($\alpha = 0.05$), we conduct a level α test that rejects H_0 if:

- $X \geq k^*$ where k^* is critical value and

$$k^* = \min \left\{ k \,\middle|\, \sum_{i=k}^{n} \binom{n}{i} p_0^i (1 - p_0)^{n-i} \leq \alpha \right\}$$

- $p - value \leq \alpha$ where $p - value$ is a probability measure of the likelihood of results greater than observed values assuming the Null Hypothesis is true and

$$p - value = \sum_{i=x}^{n} \binom{n}{i} p_0^i (1 - p_0)^{n-i}$$

with x, the observed number of successes.

We solve this using the code below.

```
import numpy as np
import pandas as pd
from scipy.stats import binomtest

data = pd.read_csv("indian_match_summary.csv")
data['Result_Binary'] = data['Result'].apply(lambda x: 1 if x == 'won' else 0)
num_successes = data['Result_Binary'].sum()
total_trials = data['Result_Binary'].count()

sample_proportion = num_successes/total_trials
p0 = 0.5
alpha = 0.05
result = binomtest(num_successes, total_trials, p=p0, alternative='greater')
p_value = result.pvalue
reject_null = p_value < alpha

critical_values = []
for k in range(total_trials + 1):
    test_result = binomtest(k, total_trials, p=p0, alternative='greater')
    if test_result.pvalue <= alpha:
        critical_values.append(k)
critical_value = min(critical_values) if critical_values else None

print(f"Sample size (n): {total_trials}")
print(f"Number of successes (X): {num_successes}")
print(f"Sample proportion (p^): {sample_proportion:.4f}")
print(f"Significance level (alpha): {alpha}")
print(f"Critical value (k*): {critical_value}")
print(f"p-value: {p_value:.8f}")
print(f"Result: {'Successful in Rejecting H0' if reject_null
else 'Failed to reject H0'}")
```

Output:

```
Sample size (n): 243
Number of successes (X): 158
Sample proportion (p^): 0.6502
```

```
Significance level (alpha): 0.05
Critical value (k*): 135
p-value: 0.00000164
Result: Successful in Rejecting H0
```

It is statistically evident that $\hat{p} > p_0$ and $p - value \ll \alpha$. This result suggests that the Indian cricket team is winning over half of their matches. Hence, only hypothesis $H1$ is true.