

Lab5

Cui Qingxuan, Nisal Amashan

2025-02-23

Contents

1 Collaborations	1
2 Question 1	1
3 Question 2	1
3.1 Inverse Transformation Method Generating Random Values	1
3.2 Sign Test	2
4 Appendix	3

1 Collaborations

Nisal Amashan: Responsible for the question 1.

Cui Qingxuan: Responsible for the question 2.

2 Question 1

3 Question 2

3.1 Inverse Transformation Method Generating Random Values

X obeys Gumbel distribution whose cumulative distribution function is:

$$F(x) = \exp(-\exp(-(x - \mu - c))), \quad c = \log(\log(2))$$

.

Then for a uniform distribution $U \sim U(0, 1)$, $X = F^{-1}(U)$ will obey that distribution.

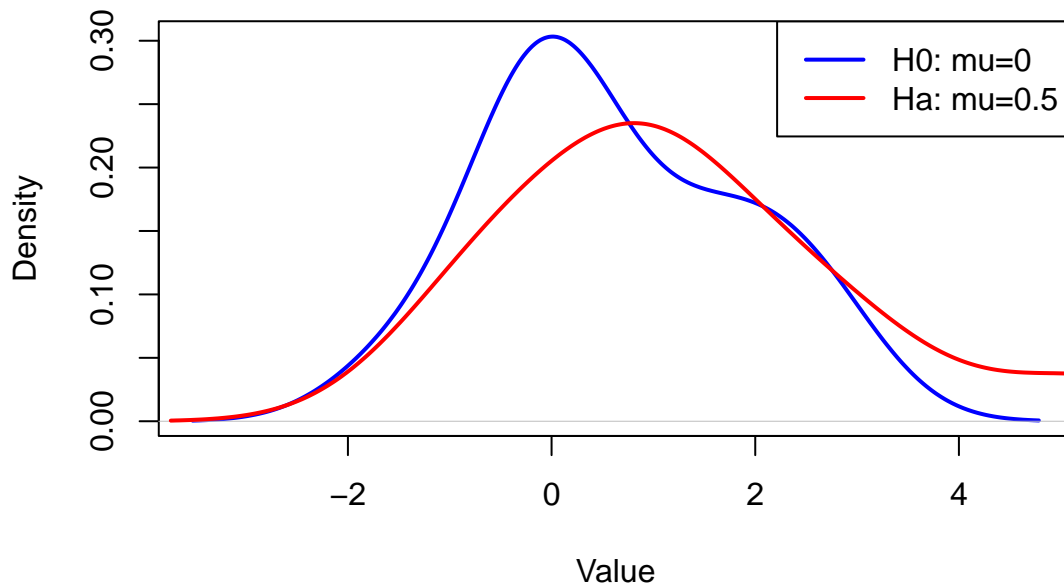
Based on that, the solution of equation $U = \exp(-\exp(-(x - \mu - c)))$ is:

$$X = -\ln(-\ln(U)) + \mu + c$$

where, $U \sim U(0, 1)$, $c = \log(\log(2))$. So we can generate random values in term of this.

We set $\mu = [0, 0.5]$, and we get random values like the plot below.

Smoothed Density Curves for both Data

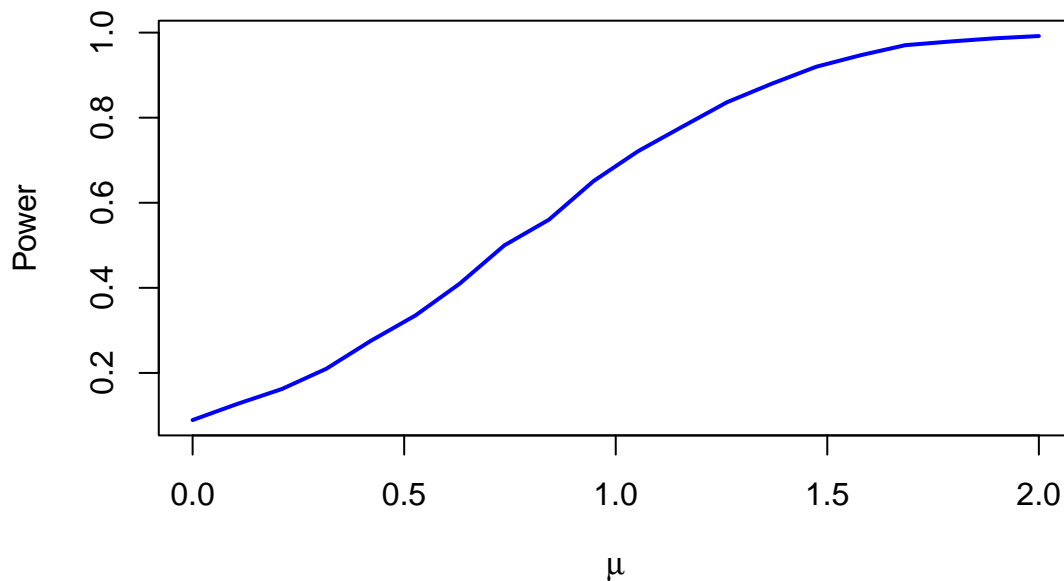


3.2 Sign Test

We selected 20 values for μ and set the number of repetitions to 10,000.

To determine an appropriate number of repetitions, we initially tested values of 10, 100, 1,000, and 10,000. However, lower repetition counts resulted in noticeable fluctuations in the power values, which should ideally exhibit a smooth and stable increase as the plot below. Since our null hypothesis states that $\mu = 0$, the power is expected to be lower when μ is closer to 0. Therefore, we chose 10,000 repetitions to ensure the reliability and consistency of our results.

Power Curve



4 Appendix

```
generate_gumbel = function(n, mu, c = log(log(2))) {
  U <- runif(n)
  X <- mu + c - log(-log(U))
  return(X)
}

sign_test = function(mu){
  x = generate_gumbel(n=13, mu)
  s = sum(x>0)
  p_value = binom.test(s, n, p = 0.5, alternative = "greater")$p.value
  return(p_value)
}

# power = P(reject H0| Ha is true)
compute_power = function(mu, n = 13, reps=10000, alpha = 0.05) {
  reject_count = sum(replicate(reps, sign_test(generate_gumbel(n, mu)))) < alpha)
  # repeat the simulation for reps times
  return(reject_count / reps)
}

# H0: mu=0
set.seed(123)
n = 13
data_H0 = round(generate_gumbel(n, mu = 0), 3)
# Ha: mu>0
data_Ha = round(generate_gumbel(n, mu = 0.5), 3)
# a appropriate grid values for mu

# calculate density
# work for presentation
denH0 = density(data_H0)
denHa = density(data_Ha)

plot(denH0, col = "blue", lwd = 2, main = "Smoothed Density Curves for both Data", xlab = "Value", ylab = "Density")
lines(denHa, col = "red", lwd = 2)
legend("topright", legend = c("H0: mu=0", "Ha: mu=0.5"), col = c("blue", "red"), lwd = 2)

# sign test
# set grid values
mu = seq(from=0, to=2, length.out=20)
power_values = sapply(mu, compute_power)
# reps has been defined in the definition of compute power function as 1000
plot(mu, power_values, type = "l", col = "blue", lwd = 2,
      xlab = expression(mu), ylab = "Power", main = "Power Curve")
```