Lab Report: Lab5 - Computational Statistics

Dhanush Kumar Reddy Narayana Reddy (dhana004), Udaya Shanker Mohanan Nair (udamo524)

2025-02-25

Introduction

Implementation of 2 Assignment questions of Computational Statistics Lab 5 .

Contributions

Member: Dhanush Kumar Reddy Narayana Reddy, Liu Id: dhana004, Contribution: Report writing and coding of question 1.

Member: Udaya Shanker Mohanan Nair, Liu Id: udamo524, Contribution: Report writing and coding of question 2.

Question 1

Part A

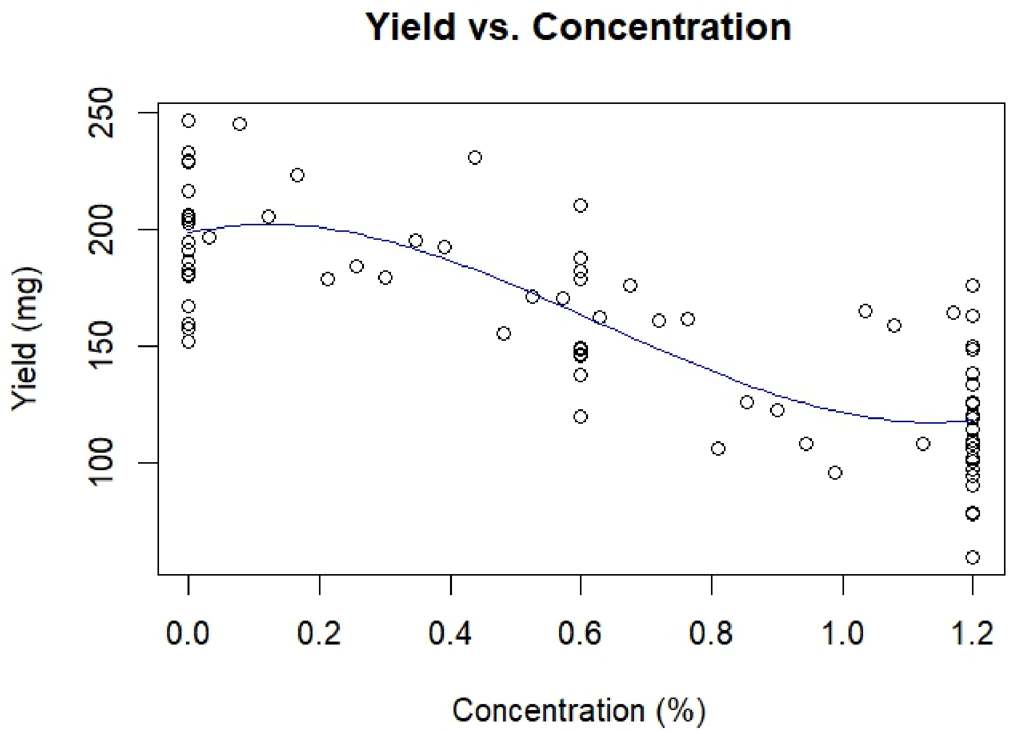
$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \epsilon$$

The summary of Cubic regression model:

```
Call:
lm(formula = yield ~ poly(concentration, 3, raw = TRUE), data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-58.909 -17.239  -3.878  18.138  58.091

Coefficients:
                                    Estimate Std. Error t value Pr(>|t|)
(Intercept)                          198.568      5.583  35.564   <2e-16 ***
poly(concentration, 3, raw = TRUE)1   66.714     72.748   0.917   0.3620
poly(concentration, 3, raw = TRUE)2 -305.221    166.937  -1.828   0.0714 .
poly(concentration, 3, raw = TRUE)3  161.634     92.076   1.755   0.0832 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 26.58 on 76 degrees of freedom
Multiple R-squared:  0.6406,    Adjusted R-squared:  0.6264
F-statistic: 45.15 on 3 and 76 DF,  p-value: < 2.2e-16
```

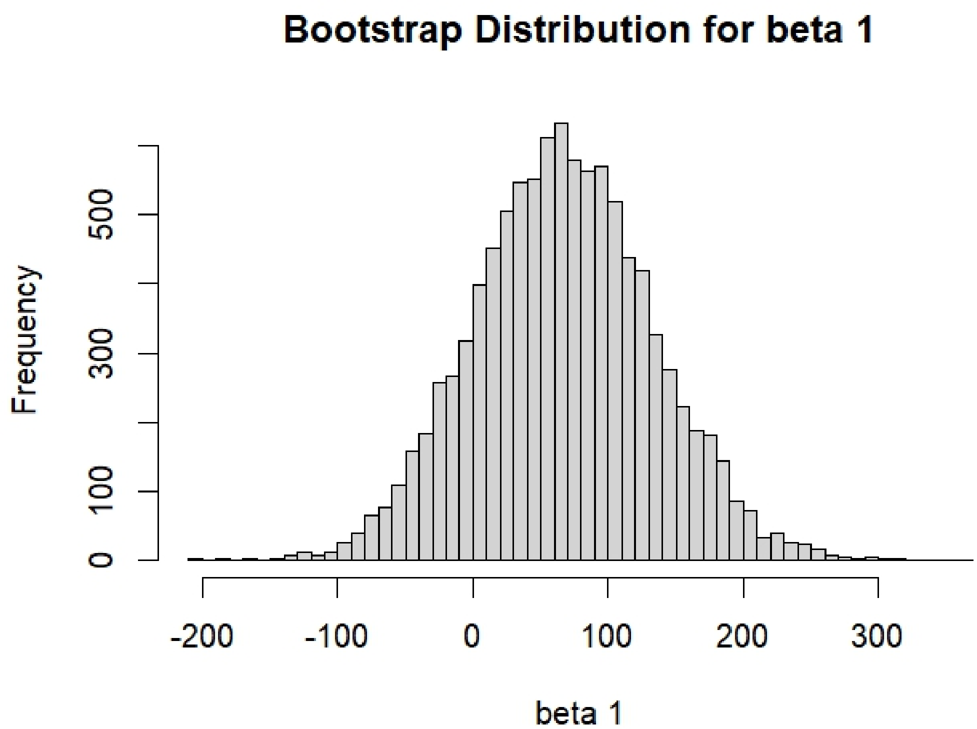
$$\text{yield} = 198.568 + 66.714x - 305.221x^2 + 161.634x^3 + \varepsilon$$

Part B



95%-Confidence Interval Analytical (lm): -78.17667 211.605

Part C



95%-Bootstrap Confidence Interval Bootstrap Percentile (manual): -61.1966 198.2414

Part D

```
# BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
# Based on 10000 bootstrap replicates
#
# CALL :
#   boot.ci(boot.out = bs_result, type = c("perc", "bca"))
#
# Intervals :
#   Level      Percentile            BCa
# 95%   (-63.34, 199.09 )   (-59.78, 200.63 )
# Calculations and Intervals on Original Scale
```

The 95% confidence interval for beta1 using the percentile method is (-63.34, 199.09), while the BCa method provides a slightly adjusted interval of (-59.78, 200.63).

Part E

Confidence Interval Comparison

Method	95% Confidence Interval for beat1
Analytical (lm)	[-78.17667, 211.60502]
Bootstrap Percentile (manual)	[-61.1966, 198.2414]
Bootstrap Percentile (boot package)	[-63.34, 199.09]
Bootstrap BCa (boot package)	[-59.78, 200.63]

Comparision:

Analytical vs. Bootstrap Approaches: (1)The analytical confidence interval (from lm) is wider than all the bootstrap-based intervals.

(2)The analytical method assumes normality of residuals, which might not be valid if the error distribution is skewed or has outliers.

Bootstrap Methods (Percentile and BCa): (1)The manual percentile method and boot package percentile method give almost the same interval, which suggests that the bootstrap implementation is correct.

(2)The BCa (bias-corrected and accelerated) interval is slightly different from the percentile intervals, particularly at the lower bound (-59.78 vs -63.34). This suggests that the bootstrap distribution is slightly skewed, and the BCa method accounts for this bias.

Width of Confidence Intervals: (1)The analytical CI [-78.17667, 211.60502] is the widest.

(2)The BCa method [-59.78, 200.63] is slightly narrower than the other bootstrap CIs, adjusting for bias and skewness.

(3)The bootstrap CIs provide more precise estimates than the analytical CI, which suggests that the normality assumption in the analytical approach might not hold perfectly.

Conclusion: (1)The bootstrap confidence intervals are more reliable because they do not rely on normality assumptions and directly estimate variability from resampling.

(2)The BCa interval is likely the best choice, as it adjusts for potential bias and skewness in the bootstrap distribution.

(3)The analytical method overestimates uncertainty, possibly due to non-normal residuals or influential points.

Question 2

Given a Gumbel distribution with scale parameter 1 and location parameter $\mu + c$, where $c = \log(\log(2))$. And this distribution has the following distribution function. For this distribution, median of a random variable is .

Now we are gone to generate random variables(Gumbel) using Inverse Transformation Method.

The CDF of the Gumbel distribution is given by

$$F(x) = \exp(-\exp(-(x - \mu - c)))$$

where

$$c = \log(\log(2))$$

Inverse Transformation of this function for a random variable is given by $Y \sim Y(0,1)$.

$$X = F^{-1}(Y)$$

$$F(x) = \exp(-\exp(-(x - \mu - c)))$$

Set $Y = F(x)$, where $Y \sim Y(0,1)$

Take the natural logarithm on both sides:

$$\log(Y) = -\exp(-(x - \mu - c))$$

Take the logarithm again:

$$\log(-\log(Y)) = -(x - \mu - c)$$

Solve for x:

$$x = -\log(-\log(Y)) + \mu + c$$

```
## Loading required package: lattice
```

```
##
## Attaching package: 'BSDA'
```

```
## The following object is masked from 'package:datasets':
##
##     Orange
```

Now we need to find out power of the test for median values ranging from 0 to 2, where for each value we will using number of observations, n as 13 and then used sign test(used SIGN.test in this case).

I have tested for 50 different values of median.

Jotting Power of few median values.

```
## Median Value:  0   Power: 0.03
```

```
## Median Value:  0.5306122   Power: 0.196
```

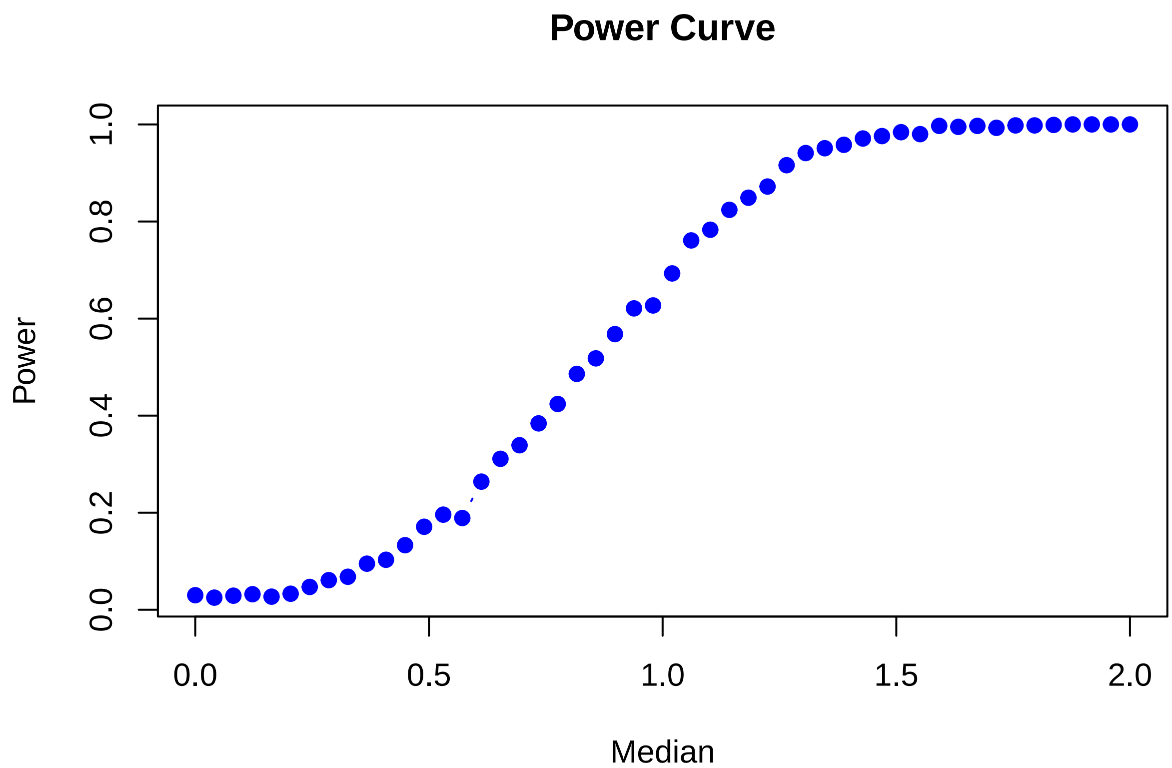
```
## Median Value:  1.020408   Power: 0.693
```

```
## Median Value:  1.510204   Power: 0.984
```

```
## Median Value:  2   Power: 1
```

from this we understand as the value of median increases the power also increases.

Now plotting the power curve



Appendix

Question 1

```
# Question 1: Bootstrap for regression
# Part A
# Load the dataset
data <- read.table("C:/Users/dhanu/OneDrive/Documents/Computational_Statistics/Lab5/kresseertrag.dat", 
colnames(data) <- c("observation", "concentration", "yield")

# Fit a cubic regression model
model <- lm(yield ~ poly(concentration, 3, raw = TRUE), data = data)

# Display model summary
summary(model)

# Part B
# The coefficients and 95%-confidence intervals
coefficients <- coef(model)
conf_intervals <- confint(model, level = 0.95)
cat("95%-Confidence Interval Analytical (lm): \n")
cat(conf_intervals[2, ])
# -78.17667 211.605
```

```r
# Plot
plot(data$concentration, data$yield, main = "Yield vs. Concentration",
     xlab = "Concentration (%)", ylab = "Yield (mg)")
curve(coefficients[1] + coefficients[2]*x + coefficients[3]*x^2 +
        coefficients[4]*x^3, add = TRUE, col = "blue")

# Part C
set.seed(123)
b0 <- 10000

#  bootstrap resampling function
bootstrap <- function(data, b0, parameter_index) {
  bs_beta1 <- numeric(b0)
  for (i in 1:b0) {
    bs_data <- data[sample(nrow(data), replace = TRUE), ]
    bs_model <- lm(yield ~ poly(concentration, 3, raw = TRUE), data = bs_data)
    bs_beta1[i] <- coef(bs_model)[2]
  }
  return(bs_beta1)
}
bs_beta1 <- bootstrap(data, b0, 2)

# 95%-bootstrap confidence interval 95% percentile interval
bs_ci <- quantile(bs_beta1, probs = c(0.025, 0.975))
cat("95%-Bootstrap Confidence Interval Bootstrap Percentile (manual): \n")
cat(bs_ci)
# -61.1966 198.2414

# Plot histogram of bootstrap distribution
hist(bs_beta1, main = "Bootstrap Distribution for beta 1", xlab = "beta 1",
     breaks = 50)

# Part D
library(boot)

# Function to fit the model and extract the parameter of interest
bs_fn <- function(data, indices) {
  bs_data <- data[indices, ]
  bs_model <- lm(yield ~ poly(concentration, 3, raw = TRUE), data = bs_data)
  return(coef(bs_model)[2])
}
bs_result <- boot(data, statistic = bs_fn, R = b0)

# Percentile and BCa confidence intervals
ci_perc_bca <- boot.ci(bs_result, type = c("perc", "bca"))
cat("95%-Percentile and 95%-BCa Confidence Interval: \n")
ci_perc_bca
# BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
# Based on 10000 bootstrap replicates
#
# CALL :
#   boot.ci(boot.out = bs_result, type = c("perc", "bca"))
#
```

```
# Intervals :
#   Level     Percentile            BCa
# 95%   (-63.34, 199.09 )   (-59.78, 200.63 )
# Calculations and Intervals on Original Scale

# Part E
# Comparision:
# Analytical vs. Bootstrap Approaches:
# (1)The analytical confidence interval (from lm) is wider than all the bootstrap-based intervals.
# (2)The analytical method assumes normality of residuals, which might not be valid if the error distri
#
# Bootstrap Methods (Percentile and BCa):
# (1)The manual percentile method and boot package percentile method give almost the same interval, whi
# (2)The BCa (bias-corrected and accelerated) interval is slightly different from the percentile interv
#
# Width of Confidence Intervals:
#
# (1)The analytical CI [-78.17667, 211.60502] is the widest.
# (2)The BCa method [-59.78, 200.63] is slightly narrower than the other bootstrap CIs, adjusting for b
# (3)The bootstrap CIs provide more precise estimates than the analytical CI, which suggests that the n
#
# Conclusion:
# (1)The bootstrap confidence intervals are more reliable because they do not rely on normality assumpt
# (2)The BCa interval is likely the best choice, as it adjusts for potential bias and skewness in the b
# (3)The analytical method overestimates uncertainty, possibly due to non-normal residuals or influenti
```

Question 2

```
library(BSDA)
c <- log(log(2))
gumbel_fn <- function(median,n = 13) {
  Y <- runif(n)
  X <- -log(-log(Y)) + median + c
  return(X)
}
median_values <- seq(0, 2, length.out = 50)
len <- length(median_values)
power <- numeric()
for (j in seq_along(median_values)) {
  median <- median_values[j]
  count <- 0
  for (i in 1:1000) {
    data <- gumbel_fn(median)
    test <- SIGN.test(data)
    if(test$p.value < 0.05){
      count <- count + 1
    }
  }
  power[j] <- count / 1000
}
cat("Median Value: ", median_values[1],"  Power:",power[1])
```

```r
cat("Median Value: ", median_values[14],"  Power:",power[14])
cat("Median Value: ", median_values[26],"  Power:",power[26])
cat("Median Value: ", median_values[38],"  Power:",power[38])
cat("Median Value: ", median_values[50],"  Power:",power[50])

plot(median_values, power, type = "b", pch = 19, col = "blue",
     xlab = "Median", ylab = "Power", main = "Power Curve")
```