Project Bootstrap (data_100529711)

Hsieh Boh Yang, Emmanuel Mathew Thomas

14 May 2024

Contents

Introduction

In this project, we explored the application of a robust linear regression model coupled with bootstrap resampling to analyze a dataset containing variables y, x1, x2, and x3. The primary objective was to build a reliable predictive model for the response variable y while accounting for outliers and non-normality in the error residuals.

Methodology

Robust Linear Regression Model: We began by constructing a robust linear regression model using the rlm() function from the robust base package in R. This model incorporates all three covariates (x1, x2, and x3) to predict the response variable y.

Bootstrap Confidence Intervals: To assess the significance of the regressors' coefficients, we employed bootstrap resampling techniques. The type of bootstrap intervals used here is the "Bias-Corrected and Accelerated" (BCa) intervals. These intervals adjust for both bias and skewness in the bootstrap distribution The boot() function was used to generate bootstrap samples, and bootstrap confidence intervals were calculated using the boot.ci() function. This approach provides robust estimates of the coefficients' significance, particularly in the presence of outliers.

Backward Elimination: We implemented backward elimination to refine the model by iteratively removing covariates that do not contribute significantly to the regression model. This process relied on bootstrap confidence intervals to determine the statistical significance of each covariate.

Confidence Intervals on Coefficients: For the final model obtained through backward elimination, we computed 95% confidence intervals on the regression coefficients using the confint() function. These intervals provide insights into the precision of the coefficient estimates.

Confidence Interval on Mean Response: Lastly, we constructed a 95% confidence interval on the mean response of the chosen model when x1, x2, and x3 are set to specific values (e.g., 14, 14, 14). This interval offers a range within which the true mean response is likely to lie.

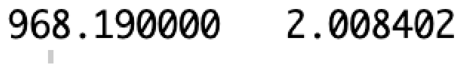
Conclusion

In conclusion, this project demonstrates the effectiveness of robust linear regression techniques in mitigating the impact of outliers and non-normality in regression analysis. By incorporating bootstrap resampling and backward elimination, we obtained a robust predictive model for the response variable y, along with reliable estimates of the coefficients' significance. This approach enhances the reliability and validity of statistical inference in the presence of challenging data conditions.

Results

CIs on the regression coefficients for the initial model

The 95% BCa intervals were found to (968.19000, 2.008402)

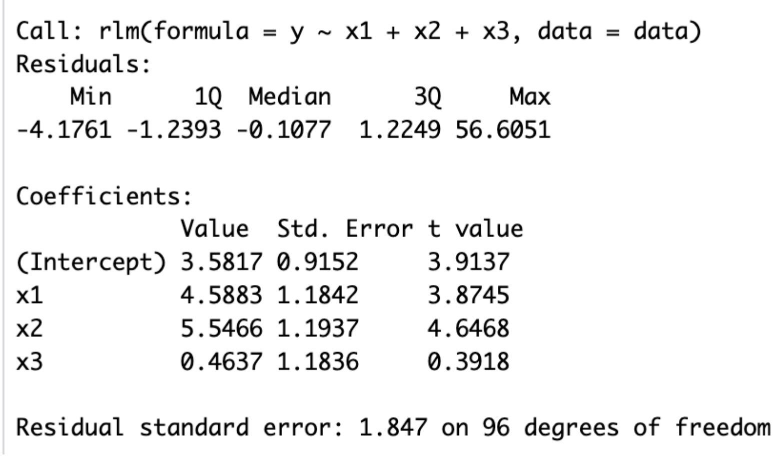


```
968.190000   2.008402
```

Backward elimination procedure

Covariate x2 was found to be not significantly contributing to the regression model; hence, it was removed.

Initial Model:

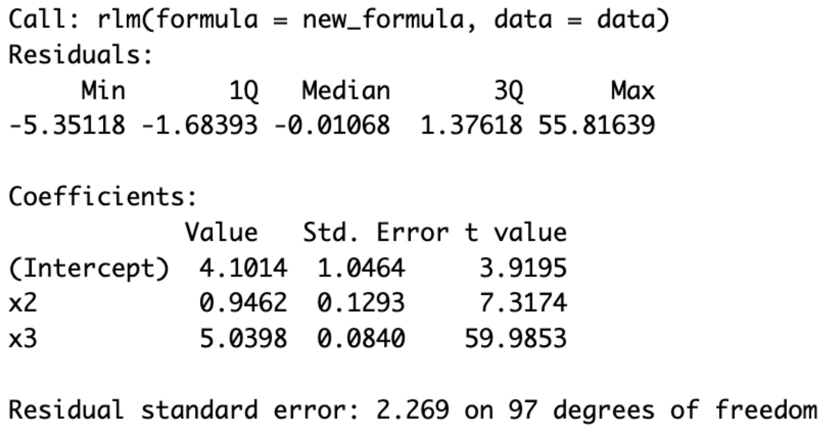


```
Call: rlm(formula = y ~ x1 + x2 + x3, data = data)
Residuals:
    Min      1Q  Median      3Q     Max 
-4.1761 -1.2393 -0.1077  1.2249 56.6051 

Coefficients:
            Value  Std. Error t value
(Intercept) 3.5817 0.9152     3.9137 
x1          4.5883 1.1842     3.8745 
x2          5.5466 1.1937     4.6468 
x3          0.4637 1.1836     0.3918 

Residual standard error: 1.847 on 96 degrees of freedom
```

After Backward Elimination:



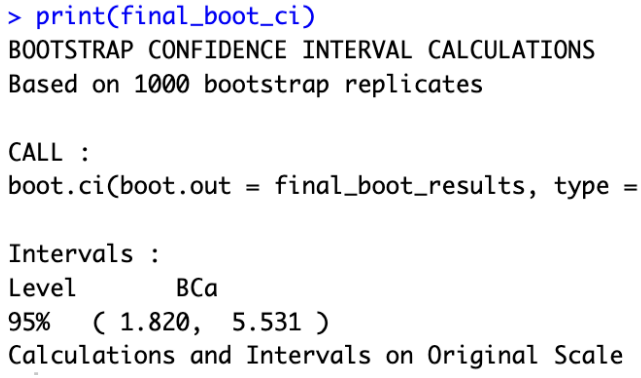
```
Call: rlm(formula = new_formula, data = data)
Residuals:
     Min       1Q   Median       3Q      Max 
-5.35118 -1.68393 -0.01068  1.37618 55.81639 

Coefficients:
            Value   Std. Error t value
(Intercept)  4.1014  1.0464     3.9195
x2           0.9462  0.1293     7.3174
x3           5.0398  0.0840    59.9853

Residual standard error: 2.269 on 97 degrees of freedom
```

CIs on the regression coefficients for the final model

The final 95% BCa interval was found to be (1.820, 5.31)



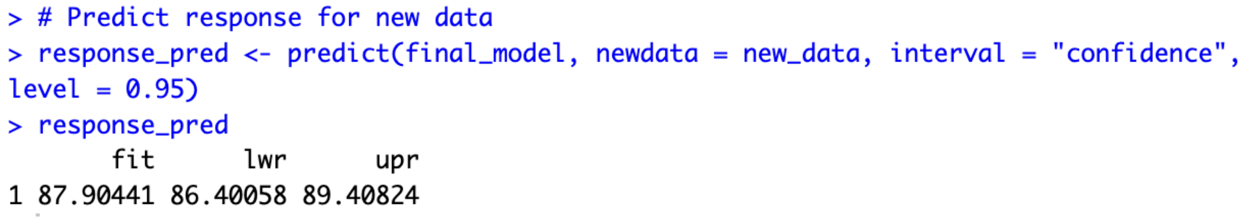
```
> print(final_boot_ci)
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL : 
boot.ci(boot.out = final_boot_results, type = "bca")

Intervals : 
Level       BCa          
95%   ( 1.820,  5.531 )  
Calculations and Intervals on Original Scale
```

CI(s) on the mean response

The CI for the mean response based on final regression model was found to be (86.40058, 89.40824)



```
> # Predict response for new data
> response_pred <- predict(final_model, newdata = new_data, interval = "confidence", 
level = 0.95)
> response_pred
       fit      lwr      upr
1 87.90441 86.40058 89.40824
```

Appendix 1: Project Code

```
# Load necessary libraries
library(MASS)
library(boot)

# Load dataset
data <- read.csv("/Users/emmanuel/Desktop/GEM Explorer - UC3M/Simulation in Prob&Stats/Bootstrap Project/data_100529711.csv", header=TRUE)

# 1. linear regression model with the three covariates
# Fit robust linear regression model
robust_model <- rlm(y ~ x1 + x2 + x3, data=data)
summary(robust_model)

# Bootstrap for confidence intervals
boot_func <- function(data, indices) {
  fit <- rlm(y ~ x1 + x2 + x3, data=data[indices, ])
  return(coef(fit))
}
boot_results <- boot(data, boot_func, R=1000)
boot_ci <- boot.ci(boot_results, type="bca")

# Extract bootstrap confidence intervals for coefficients
conf_intervals <- boot_ci$bca[1, c(3, 4)]
print(conf_intervals)
```

```
# Extract lower and upper bounds of confidence intervals
lower_bound <- conf_intervals[1]
upper_bound <- conf_intervals[2]

# 2. Backward elimination
# Identify non-significant coefficients
non_sig_vars <- which(lower_bound > 0 | upper_bound < 0)

# Identify significant variables
sig_vars <- paste("x", setdiff(1:3, non_sig_vars), sep = "")

# Construct new formula
new_formula <- as.formula(paste("y ~", paste(sig_vars, collapse = " + ")))

# Fit model with significant variables only
final_model <- rlm(new_formula, data = data)
summary(final_model)

# Step 3: Confidence intervals on the regression coefficients of final model
final_boot_func <- function(data, indices) {
  fit <- rlm(y ~ x1 + x2 + x3, data=data[indices, ])
  return(coef(fit))
}

final_boot_results <- boot(data, final_boot_func, R=1000)
```

```
final_boot_ci <- boot.ci(final_boot_results, type="bca")

print(final_boot_ci)


# Step 4: Confidence interval on the mean response

# Define new data point

new_data <- data.frame(x1 = 14, x2 = 14, x3 = 14)

# Predict response for new data

response_pred <- predict(final_model, newdata = new_data, interval = "confidence", level = 0.95)

response_pred
```