STAT 4360 (Introduction to Statistical Learning, Spring 2023)
Mini Project 3
Name: Sayema Rahman

---

1. (a) The correlation matrix tells us that the strongest positive correlation would be between variables Age and Pregnancies at approximately 0.54. This indicates that as women becomes older, they are more likely to become pregnant and vice versa. The second strongest correlation we have is between Glucose and Outcome at approximately 0.46.

```
> summary(diabetes)
  Pregnancies        Glucose      BloodPressure    SkinThickness       Insulin           BMI
 Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   Min.   : 0.00
 1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 63.50   1st Qu.:  0.00   1st Qu.:  0.00   1st Qu.:27.38
 Median : 3.000   Median :117.0   Median : 72.00   Median : 23.00   Median : 40.00   Median :32.30
 Mean   : 3.704   Mean   :121.2   Mean   : 69.15   Mean   : 20.93   Mean   : 80.25   Mean   :32.19
 3rd Qu.: 6.000   3rd Qu.:141.0   3rd Qu.: 80.00   3rd Qu.: 32.00   3rd Qu.:130.00   3rd Qu.:36.80
 Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :110.00   Max.   :744.00   Max.   :80.60
 DiabetesPedigreeFunction      Age            Outcome
 Min.   :0.0780           Min.   :21.00   Min.   :0.000
 1st Qu.:0.2440           1st Qu.:24.00   1st Qu.:0.000
 Median :0.3760           Median :29.00   Median :0.000
 Mean   :0.4709           Mean   :33.09   Mean   :0.342
 3rd Qu.:0.6240           3rd Qu.:40.00   3rd Qu.:1.000
 Max.   :2.4200           Max.   :81.00   Max.   :1.000
```

Figure 1: Summary of the full data set

```
Call:
glm(formula = Outcome ~ Age + BloodPressure + BMI + DiabetesPedigreeFunction +
    Glucose + Insulin + Pregnancies + SkinThickness, family = binomial(),
    data = diabetes)

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)
(Intercept)              -8.0264511  0.4306345 -18.639  < 2e-16 ***
Age                       0.0129414  0.0057020   2.270  0.02323 *
BloodPressure            -0.0096446  0.0032441  -2.973  0.00295 **
BMI                       0.0775549  0.0088819   8.732  < 2e-16 ***
DiabetesPedigreeFunction  0.8877583  0.1860275   4.772 1.82e-06 ***
Glucose                   0.0337202  0.0022258  15.150  < 2e-16 ***
Insulin                  -0.0012426  0.0005786  -2.148  0.03175 *
Pregnancies               0.1263845  0.0199997   6.319 2.63e-10 ***
SkinThickness             0.0005185  0.0042301   0.123  0.90244
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2569.4  on 1999  degrees of freedom
Residual deviance: 1914.3  on 1991  degrees of freedom
AIC: 1932.3

Number of Fisher Scoring iterations: 5
```

Figure 2: Summary of the full model

```
> summary(reduced_model)

Call:
glm(formula = Outcome ~ Age + BloodPressure + BMI + DiabetesPedigreeFunction +
    Glucose + Insulin + Pregnancies, family = binomial(), data = diabetes)

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)
(Intercept)              -8.0273146  0.4306244 -18.641  < 2e-16 ***
Age                       0.0128944  0.0056879   2.267  0.02339 *
BloodPressure            -0.0095806  0.0032013  -2.993  0.00276 **
BMI                       0.0778743  0.0084946   9.167  < 2e-16 ***
DiabetesPedigreeFunction  0.8894946  0.1855205   4.795 1.63e-06 ***
Glucose                   0.0336810  0.0022020  15.296  < 2e-16 ***
Insulin                  -0.0012123  0.0005228  -2.319  0.02042 *
Pregnancies               0.1263707  0.0199944   6.320 2.61e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2569.4  on 1999  degrees of freedom
Residual deviance: 1914.3  on 1992  degrees of freedom
AIC: 1930.3

Number of Fisher Scoring iterations: 5
```

Figure 3: Summary of reduced model

```
> confint(reduced_model, level = 0.95)
Waiting for profiling to be done...
                                2.5 %        97.5 %
(Intercept)              -8.889630784 -7.2009252668
Age                       0.001711033  0.0240290378
BloodPressure            -0.015885768 -0.0033221648
BMI                       0.061474284  0.0947952879
DiabetesPedigreeFunction  0.527470753  1.2549028449
Glucose                   0.029435255  0.0380709843
Insulin                  -0.002241105 -0.0001893038
Pregnancies               0.087447559  0.1658700222
```

Figure 4: 95% confidence interval of reduced model

```
> print(correlation)
                           Outcome         Age BloodPressure        BMI DiabetesPedigreeFunction
Outcome                 1.00000000  0.23650925    0.07595808 0.27672554                0.15545908
Age                     0.23650925  1.00000000    0.23837508 0.03898737                0.02656950
BloodPressure           0.07595808  0.23837508    1.00000000 0.28154513                0.05133095
BMI                     0.27672554  0.03898737    0.28154513 1.00000000                0.12571935
DiabetesPedigreeFunction 0.15545908  0.02656950    0.05133095 0.12571935                1.00000000
Glucose                 0.45842130  0.25449621    0.13804400 0.22686443                0.12324343
Insulin                 0.12092362 -0.08587910    0.08738405 0.22301161                0.19271873
Pregnancies             0.22443699  0.53945719    0.14967246 0.01947503               -0.02545316
                          Glucose     Insulin Pregnancies
Outcome                 0.4584213  0.12092362  0.22443699
Age                     0.2544962 -0.08587910  0.53945719
BloodPressure           0.1380440  0.08738405  0.14967246
BMI                     0.2268644  0.22301161  0.01947503
DiabetesPedigreeFunction 0.1232434  0.19271873 -0.02545316
Glucose                 1.0000000  0.32037084  0.12040541
Insulin                 0.3203708  1.00000000 -0.07659977
Pregnancies             0.1204054 -0.07659977  1.00000000
```

Figure 5: Correlation matrix

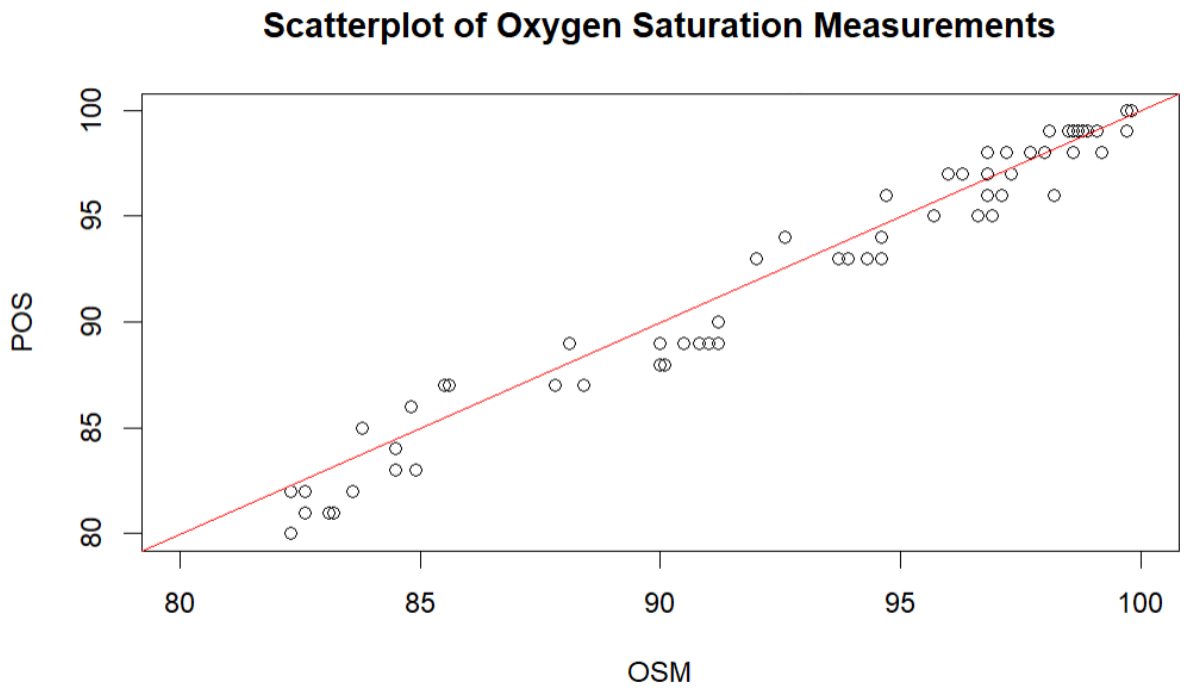
(b) I took out the variable SkinThickness from my model as it has a p-value of 0.698078, well above 0.05.
(c) The final model in equation form is

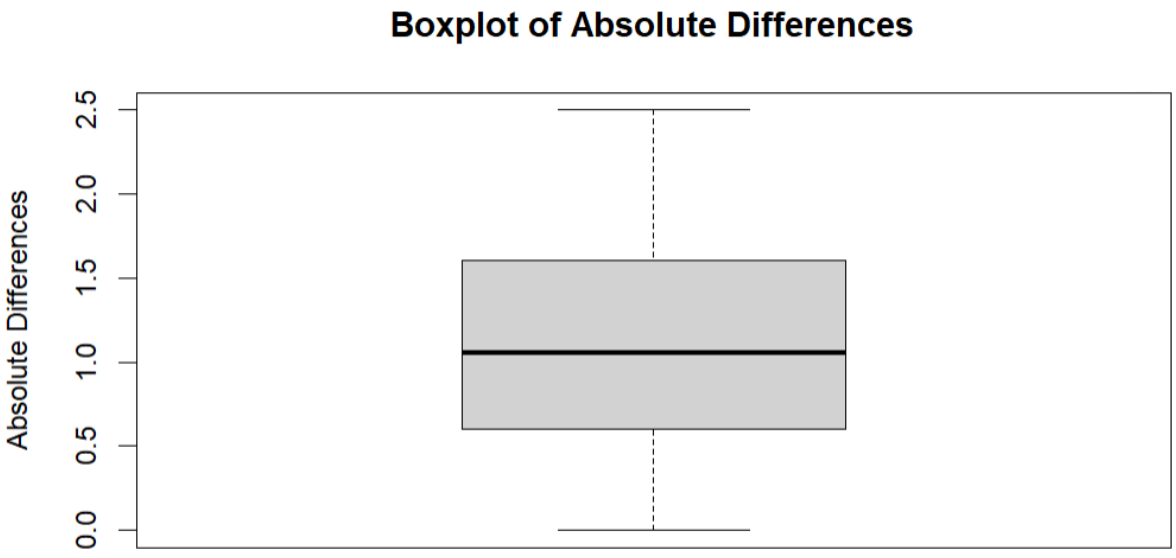
$$log(\frac{p}{1-p}) = -8.027 + 0.013 \times Pregnancies - 0.010 \times BloodPressure + 0.078 \times BMI + 0.890 \times DiabetesPredigreeFunction + 0.034 \times Glucose - 0.001 \times Insulin + 0.126 \times Pregnancies$$

The estimate for the variable Age is 0.0128944, the standard error is 0.0056879 and the 95% confidence interval is [0.00171, 0.02340]. The estimate for the variable BloodPressure is -0.00958, the standard error is 0.0032013 and the 95% confidence interval is [-0.01588, -0.00328]. The estimate for the variable BMI is 0.0778743, the standard error is 0.0084946 and the 95% confidence interval is [0.06147, 0.09480]. The estimate for the variable DiabetesPredigreeFunction is 0.8894946, the standard error is 0.1855205 and the 95% confidence interval is [0.52747, 1.25490]. The estimate for the variable Glucose is 0.0336810, the standard error is 0.0022020 and the 95% confidence interval is [0.02944, 0.03807]. The estimate for the variable Insulin is -0.0012123, the standard error is 0.0005228 and the 95% confidence interval is [-0.00224, -0.00019]. The estimate for Pregnancies is 0.12637, the standard error is 0.0199944 and the 95% confidence interval is [0.08745, 0.16587]. The training error rate is 0.216. For variable Age, every one-unit increase in Age, the response variable will increase by approximately 0.0129 units on average. For variable BloodPressure, the response variable will decrease by approximately 0.00958 units on average. For variable BMI, the response variable will increase by approximately 0.0779 units on average.

2. (a) The error rate for the logistic regression model is 0.216. The sensitivity for the logistic regression model is 0.5672515. The specificity for the logistic regression model is 0.8966565.

3. (a) The scatterplot shows the measurements from the two methods. The boxplot of absolute differences shows the inconsistency between the methods. There is a slight distance between each point and the 45 degree line. This shows that the methods do not have perfect agreement.





(b) To argue that smaller values for theta imply better agreement, we need to understand what the total deviation index means. Total deviation index, also known as TDI is the measure of the difference between the two methods, OSM and POS. Using the 90 percent quantile value we can look at the greeted 10 percent of the absolute difference between the two methods.
(c) The 90 percent sample quantile to obtain the point estimate theta hat of theta is 2.
(d) The bias is 0 because there is only a slight difference between the mean of theta hat from the bootstrap samples and the original estimate theta hat. The standard error is 0 which shows that

there is a high precision in estimating theta. The 95% upper confidence bound is 2 which shows the range in which we can conclude 95% confidence in the true value of theta.
(e) The bootstrap method gave me the same results as what I got in part D.
(f) I think that the methods do agree but they cannot be used interchangeably. There are significant differences between the two methods.

4)a) $E(Y) = \int_0^\infty y \cdot f_Y(y;\theta,\phi)\,dy$ $\qquad b'(\theta) = \frac{d\,b(\theta)}{d\theta}$

$= \int_0^\infty y \cdot \exp\{(y\cdot\theta - \exp(\theta))/1\}\,dy$

$= \int_0^\infty y \cdot \exp(y\cdot\theta - \exp(\theta))\,dy$

$u = y \qquad dv = \exp(y\cdot\theta - \exp(\theta))\,dy$

$du = dy \qquad v = \frac{1}{\theta}\exp(y\cdot\theta - \exp(\theta))$

integration by parts

$= y\cdot\frac{1}{\theta}\exp(y\cdot\theta - \exp(\theta)) - \int\frac{1}{\theta}\exp(y\cdot\theta - \exp(\theta))\,dy$

$= y\cdot\frac{1}{\theta}\exp(y\cdot\theta - \exp(\theta)) - \frac{1}{\theta^2}\exp(y\cdot\theta - \exp(\theta)) + C$

$= \left(y\cdot\frac{1}{\theta} - \frac{1}{\theta^2}\right)\exp(y\cdot\theta - \exp(\theta)) + C$

$= \lim_{a\to\infty}\left(a\cdot\frac{1}{\theta} - \frac{1}{\theta^2}\right)\exp(a\cdot\theta - \exp(\theta)) - \left(0\cdot\frac{1}{\theta} - \frac{1}{\theta^2}\right)\exp(0\cdot\theta - \exp(\theta))$

$= \frac{1}{\theta^2}$

$b'(\theta) = \frac{\partial}{\partial\theta}(\exp(\theta))$

$= \exp(\theta)$

This shows that $E(Y) = b'(\theta)$

These expressions are the same as the standard expressions for the poisson distribution.

b) $b(\theta) = \exp(\theta)$

$= \exp(\log(\mu)) = \mu$

$a(\phi) = \phi = 1$

$c(y,\phi) = -\log(y!)$

Python Code (or R Code)

```
```{r}
library(ggplot2)
library(MASS)
diabetes <- read.table("C:/Users/sayem/Downloads/diabetes.csv",
sep=",", header = T)
# renaming columns to get rid of .. at the end of each variable
names(diabetes) <- c("Pregnancies", "Glucose", "BloodPressure",
"SkinThickness",
                     "Insulin", "BMI", "DiabetesPedigreeFunction",
"Age", "Outcome")
head(diabetes)
```
Question 1
(a) Perform an exploratory analysis of data.
```{r}
# make the full model
full_model <-
glm(Outcome~Age+BloodPressure+BMI+DiabetesPedigreeFunction+
                   Glucose+Insulin+Pregnancies+SkinThickness,
                 family = binomial(), data = diabetes)
summary(full_model)
# make a reduced model without SkinThickness as it has a p-value
above 0.05
reduced_model <-
glm(Outcome~Age+BloodPressure+BMI+DiabetesPedigreeFunction+
                     Glucose+Insulin+Pregnancies, family =
binomial(),
                   data = diabetes)
summary(reduced_model)
# confidence interval
confint(reduced_model, level = 0.95)
data <- data.frame(Age = 40, BloodPressure = 70, BMI = 30,
                   DiabetesPedigreeFunction = 0.5, Glucose = 120,
Insulin = 100,
                   Pregnancies = 5)
predict(reduced_model, newData = data, interval = 'predict')
# make a correlation
correlation <- cor(diabetes[,c("Outcome", "Age", "BloodPressure",
"BMI",
                               "DiabetesPedigreeFunction", "Glucose",
"Insulin",
                               "Pregnancies")])

print(correlation)
# make a scatter plot
pairs(diabetes[, c("Outcome", "Age", "BloodPressure", "BMI",
                   "DiabetesPedigreeFunction", "Glucose", "Insulin",
                   "Pregnancies")], main = "Scatterplots")
```
```

```
# getting the summary of the data set
summary(diabetes)
```
(b) Build a “reasonably good” logistic regression model for these data. There is
no need to explore interactions. Carefully justify all the choices you make in
building the model.
```{r}
# took out SkinThickness as it has a p-value above 0.05
reduced_model <-
glm(Outcome~Age+BloodPressure+BMI+DiabetesPedigreeFunction+
                       Glucose+Insulin+Pregnancies, family =
binomial(),
                     data = diabetes)
summary(reduced_model)
```
(c) Write the final model in equation form. Provide a summary of estimates of
the regression coefficients, the standard errors of the estimates, and 95%
confidence intervals of the coefficients. Interpret the estimated coefficients
of at least two predictors. Provide training error rate for the model.
```{r}
final_model <-
glm(Outcome~Age+BloodPressure+BMI+DiabetesPedigreeFunction+
                    Glucose+Insulin+Pregnancies, family = binomial(),
                  data = diabetes)
summary(final_model)
# confidence intervals
confint(final_model)
fitted_results <- predict(final_model, type = "response")
predicted_outcome <- ifelse(fitted_results > 0.5, 1, 0)
misclassError <- mean(predicted_outcome != diabetes$Outcome)
cat("Training Error Rate: ", misclassError)
```
2. Consider the diabetes dataset from #1. Use all predictors for all the models
considered for this problem.
(a) Fit a logistic regression model using all predictors in the data. Provide
its error rate, sensitivity, and specificity based on training data.
```{r}
# logistic regression model
full_model <-
glm(Outcome~Age+BloodPressure+BMI+DiabetesPedigreeFunction+
                    Glucose+Insulin+Pregnancies+SkinThickness,
                  family = binomial(), data = diabetes)
# Predict probabilities
predict_probability <- predict(full_model, type = "response")
```

```
# Convert probabilities to binary predictions (0 or 1)
predicted_classes <- ifelse(predict_probability > 0.5, 1, 0)
outcome <- diabetes$Outcome
# Calculate confusion matrix
conf_matrix <- table(outcome, predicted_classes)
# Calculate accuracy
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
# Calculate sensitivity (true positive rate)
sensitivity <- conf_matrix[2, 2] / sum(conf_matrix[2, ])
cat("Sensitivity:", sensitivity, "\n")
# Calculate specificity (true negative rate)
specificity <- conf_matrix[1, 1] / sum(conf_matrix[1, ])
cat("Specificity:", specificity, "\n")
# Calculate error rate
error_rate <- 1 - accuracy
cat("Error Rate:", error_rate, "\n")
```
(b) Write your own code to estimate the test error rate of the model in (a)
using LOOCV.
```{r}
# getting number of observations
observations <- nrow(diabetes)
predicted_classes <- rep(NA, observations)
full_model <-
glm(Outcome~Age+BloodPressure+BMI+DiabetesPedigreeFunction+
                    Glucose+Insulin+Pregnancies+SkinThickness,
                  family = binomial(), data = diabetes)
# LOOCV
for (i in 1:observations) {
  training_data <- diabetes[-i, ]
  # Fitting logistic regression model on training data
  model <- glm(Outcome ~ ., family = binomial(), data =
training_data)

  # Predict class for observation i
  predicted_classes[i] <- ifelse(predict(model, newdata = diabetes[i,
],
                                         type = "response") > 0.5, 1,
0)
}
conf_matrix <- table(diabetes$Outcome, predicted_classes)

# Calculate accuracy
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)

# Calculate test error rate
error_rate <- 1 - accuracy
cat("Test Error Rate:", error_rate, "\n")
```
(c) Verify your results in (b) using a package. You can use cv.glm in
R, or you

may use the caret package (https://topepo.github.io/caret/) for doing so as it
is not restricted to the GLMs.
```{r}
# Load required library
library(boot)

# Define the logistic regression model function
glm_func <- function(data, indices) {
  fit <- glm(Outcome ~ ., family = binomial(), data = data[indices, ])
  return(fit)
}

# Perform LOOCV
cv_result <- cv.glm(data = diabetes, glmfit = full_model)

# Extract the estimated test error rate
test_error_rate <- cv_result$delta[1]

# Print the estimated test error rate
cat("Estimated Test Error Rate using LOOCV:", test_error_rate, "\n")

```
3. (a) Make a scatterplot of the data and superimpose the 45 degree line. Next, make a boxplot of absolute values of differences in the measurements
from the two methods. Comment on the extent of agreement between the methods.
Note that the methods would have perfect agreement if all the points in the
scatterplot fell on the 45 degree line, or equivalently, all the differences
were zero.
```{r}
# Load the data
oxygen_saturation <- read.delim("~/oxygen_saturation.txt")
plot(oxygen_saturation$osm, oxygen_saturation$pos,
     xlab = "OSM", ylab = "POS",
     main = "Scatterplot of Oxygen Saturation Measurements",
     xlim = c(80, 100), ylim = c(80, 100))
# 45 degree line
abline(a = 0, b = 1, col = "red")
oxygen_saturation$pos <- as.double(oxygen_saturation$pos)
# Calculate absolute differences
abs_diff <- abs(oxygen_saturation$osm - oxygen_saturation$pos)

# Boxplot of absolute differences
boxplot(abs_diff,
        main = "Boxplot of Absolute Differences",
        ylab = "Absolute Differences")
```

```

(c) Provide a point estimate ˆθ of θ.
```{r}
# doing point estimate
calculate_theta_hat <- function(sample_data) {
  D <- sample_data$pos - sample_data$osm
  abs_dff <- abs(D)
  return(quantile(abs_diff, 0.90))
}
theta_hat <- calculate_theta_hat(oxygen_saturation)
theta_hat

```
(d) Write your own code to compute (nonparametric) bootstrap
estimates of bias
and standard error of ˆθ, and a 95% upper confidence bound for θ
computed using
the percentile method. Interpret the results.
```{r}
# Given data
B <- 1000  # Number of bootstrap samples

# Bootstrap resampling
bootstrap_theta_hat <- replicate(B, {
  # Generate bootstrap sample indices
  bootstrap_indices <- sample(nrow(sample_data), replace = TRUE)

  # Calculate theta_hat for bootstrap sample
  bootstrap_theta <-
calculate_theta_hat(sample_data[bootstrap_indices, ])

  # Return theta_hat for bootstrap sample
  return(bootstrap_theta)
})

# Compute bias
bias <- mean(bootstrap_theta_hat) - theta_hat

# Compute standard error
standard_error <- sd(bootstrap_theta_hat)

# Compute 95% upper confidence bound using percentile method
upper_confidence_bound <- quantile(bootstrap_theta_hat, 0.95)

# Print results
print(paste("Bias:", bias))
print(paste("Standard Error:", standard_error))
print(paste("95% Upper Confidence Bound:", upper_confidence_bound))
```
(e) Repeat the computation in (d) using boot package in R, or check
bootstrap function from the
```

SciPy library (scipy.stats.bootstrap) in Python, and compare your results.

```{r}
#install.packages("boot")
library(boot)
# Define a function to calculate theta_hat
calculate_theta_hat <- function(data, indices) {
  D <- data[indices, "OSM"] - data[indices, "POS"]
  abs_D <- abs(D)
  return(quantile(abs_D, 0.90))
}

# Bootstrap resampling using boot() function
boot_results <- boot(data, statistic = calculate_theta_hat, R = 1000)

# Compute bias
bias <- mean(boot_results$t) - calculate_theta_hat(boot_results)

# Compute standard error
standard_error <- sd(boot_results$t)

# Compute 95% upper confidence bound using percentile method
upper_confidence_bound <- quantile(boot_results$t, c(0.95))

# Print results
print(paste("Bias:", bias))
print(paste("Standard Error:", standard_error))
print(paste("95% Upper Confidence Bound:", upper_confidence_bound))
```