Problem Set 1

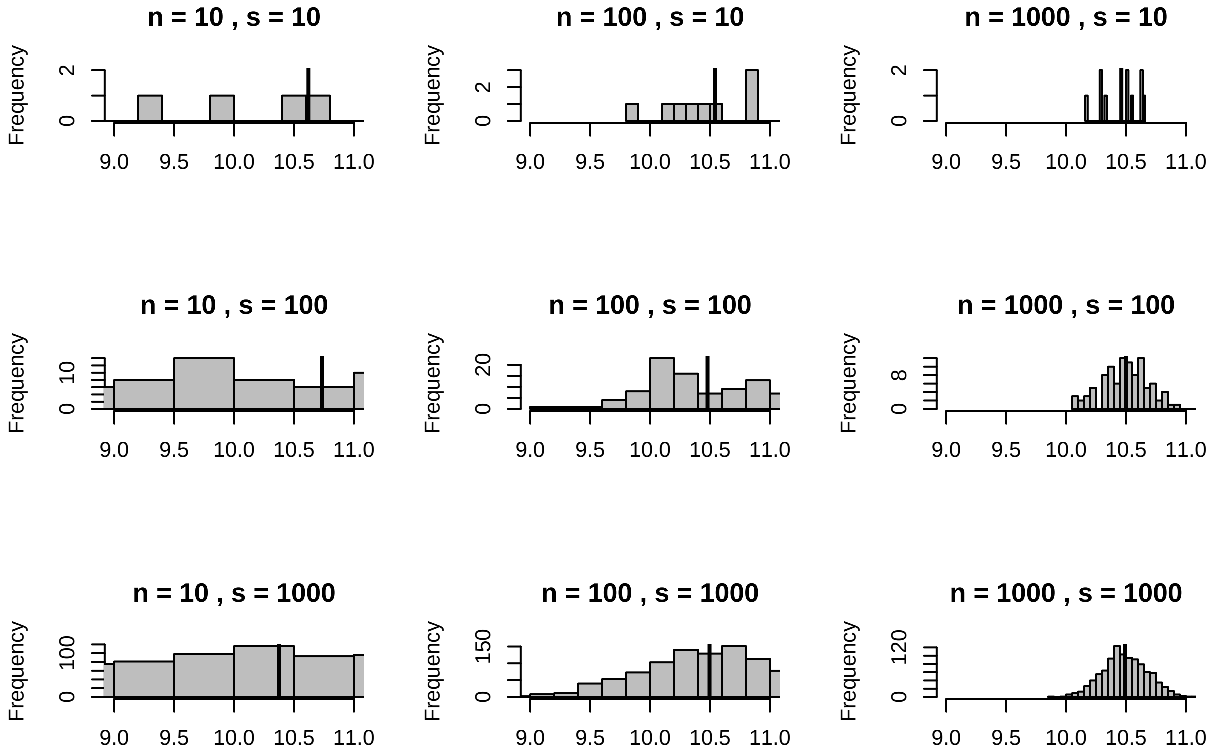
Yawen Dong

2025-01-08

Question 1

a

```r
n_values <- c(10, 100, 1000)
s_values <- c(10, 100, 1000)

par(mfrow = c(3,3))

for (s in s_values) {
  for (n in n_values) {
    sample_means <- replicate(s, mean(sample(1:20, n, replace = TRUE)))
    hist(
      sample_means,
      main = paste("n =", n, ", s =", s),
      xlab = "",
      xlim = c(9.0, 11.0),
      col = "gray",
      breaks = 20
    )
    abline(v = mean(sample_means), col = "black", lwd = 2)
  }
}
```



b

1. When the number of units within a group (n) increases, the variability in the sample means would decrease, making the estimates more precise. When the number of groups (s) increases, the sampling distribution would be smoother while the variability would not be affected. That is to say, for efficient sampling, larger number of units within a group (n) is more influential than the number of groups (s).
2. Central limit theorem. Under appropriate conditions, the distribution of a normalized version of the sample mean converges to a standard normal distribution.

Question 2

```r
f <- function(x){exp(-x) * sin(x)}
f_result <- integrate(f, lower = 2, upper = 5)
print(f_result)
```

```
## 0.03564528 with absolute error < 8.3e-16
```

Question 3

a

- Systematic Component: $y_i = 1 + 0.5x_{i1} - 2.2x_{i2} + x_{i3}$
- Stochastic Component: $\epsilon_i \sim N(\mu = 0, \sigma^2 = 1.5)$

b

```
xmat <- read.csv("xmat.csv")
print(dim(xmat))
```

i

```
## [1] 1000    3
```

Dimensions: 1000 rows and 3 columns

```
set.seed(10825)
n <- nrow(xmat)

# Generate random noise
epsilon <- rnorm(n, mean = 0, sd = sqrt(1.5))

# Define coefficients for the linear model
coef_0 <- 1
coef_1 <- 0.5
coef_2 <- -2.2
coef_3 <- 1

# Extract predictor variables
X1 <- xmat$X1
X2 <- xmat$X2
X3 <- xmat$X3

# Generate the dependent variable
y <- coef_0 + coef_1 * X1 + coef_2 * X2 + coef_3 * X3 + epsilon

# Fit a linear regression model with y and X1, X2, X3.
model <- lm(y ~ X1 + X2 + X3)
summary(model)
```

ii

```
##
## Call:
## lm(formula = y ~ X1 + X2 + X3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5330 -0.8196  0.0124  0.8168  4.5651
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.06651    0.05084   20.98   <2e-16 ***
## X1           0.48024    0.01925   24.95   <2e-16 ***
## X2          -2.26451    0.07852  -28.84   <2e-16 ***
## X3           0.95040    0.03822   24.86   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 1.225 on 996 degrees of freedom
## Multiple R-squared:  0.6792, Adjusted R-squared:  0.6782
## F-statistic: 702.9 on 3 and 996 DF,  p-value: < 2.2e-16
```

Question 4

```
library(haven)
cox <- read_dta("coxappend.dta")
attributes(cox)
```

```
## $class
## [1] "tbl_df"     "tbl"        "data.frame"
##
## $row.names
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## [26] 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
## [51] 51 52 53 54
##
## $names
##  [1] "var12"    "drop"     "year"     "enpv"     "enps"     "eneth"
##  [7] "ml"       "upper"    "enpres"   "proximit" "lnml"     "lmleneth"
## [13] "smdp"     "smdpeth"  "multi"    "enpvlml"  "enpvUpp"  "multiV"
## [19] "enpvQ"    "enpvmult" "enpvsmdp" "proxpres" "drop2"
```

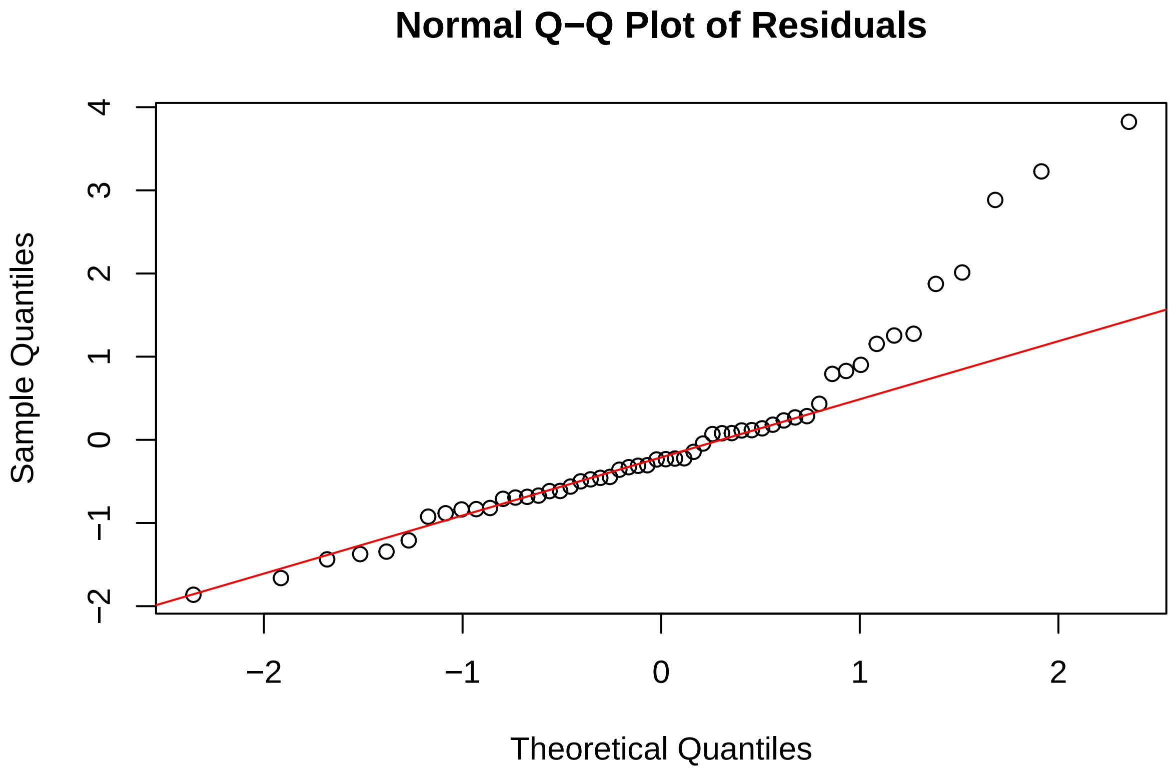
a

```
# Define the ols function
ols_regression <- function(y, X) {
  # Add intercept column
  X <- cbind(1, X)
  # Calculate the coefficients
  beta <- solve(t(X) %*% X) %*% t(X) %*% y
  # Calculate residuals
  residuals <- y - X %*% beta
  # Estimate the variance of residuals
  sigma_squared <- sum(residuals^2) / (nrow(X) - ncol(X))
  # Calculate riance-covariance matrix of coefficients
  var_beta <- sigma_squared * solve(t(X) %*% X)
  # Calculate standard errors
  std_errors <- sqrt(diag(var_beta))

  # Return results
  return(list(
    coefficients = beta,
    std_errors = std_errors
  ))
}

# Prepare the actual data
y <- cox$enps
X <- as.matrix(cox[, c("eneth",
                       "lnml",
                       "lmleneth")])

ols_results <- ols_regression(y, as.matrix(X))

# Create a table with coefficients and standard errors
results_table <- data.frame(
  Coefficient = ols_results$coefficients,
  Std_Error = ols_results$std_errors
)

# Set row names
rownames(results_table) <- c("Intercept", "eneth", "lnml", "lmleneth")

print(results_table)
```

```
##           Coefficient Std_Error
## Intercept   2.6713672 0.6072149
## eneth      -0.3619712 0.3486305
## lnml       -0.1911174 0.2967357
## lmleneth    0.4833254 0.1805094
```

b

```
X <- cbind(1, X)
residuals <- y - X %*% ols_results$coefficients

# Q-Q Plot for residuals
qqnorm(residuals, main = "Normal Q-Q Plot of Residuals")
qqline(residuals, col = "red")
```



The plot shows that the residuals mostly align with the theoretical normal distribution but deviate significantly in the upper tail. This suggests outliers or non-normality in the residuals distribution.

c

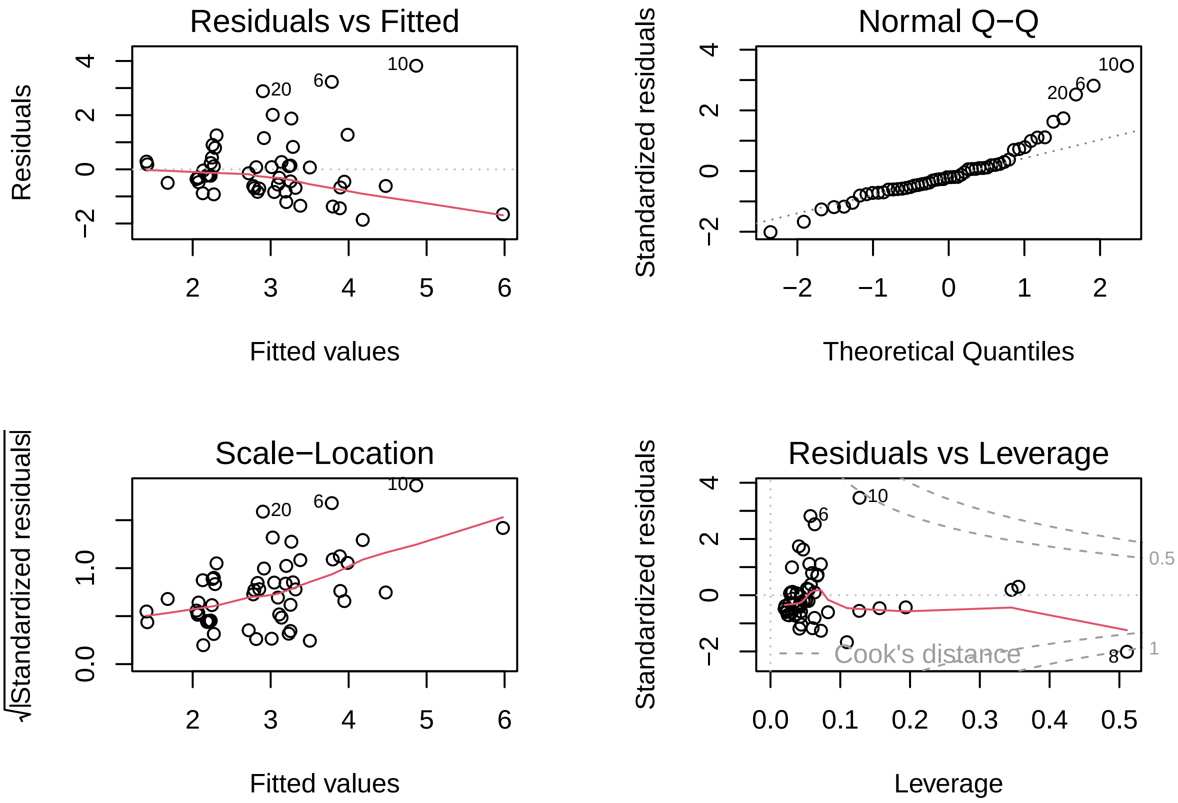
```r
ols_model <- lm(enps ~ eneth + lnml + lmleneth, data = cox)
summary(ols_model)
```

```
##
## Call:
## lm(formula = enps ~ eneth + lnml + lmleneth, data = cox)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8627 -0.6818 -0.2346  0.2605  3.8235
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.6714     0.6072   4.399 5.69e-05 ***
## eneth        -0.3620     0.3486  -1.038    0.304
## lnml         -0.1911     0.2967  -0.644    0.522
## lmleneth      0.4833     0.1805   2.678    0.010 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.181 on 50 degrees of freedom
## Multiple R-squared:  0.3629, Adjusted R-squared:  0.3247
## F-statistic: 9.493 on 3 and 50 DF,  p-value: 4.541e-05
```

d

```
par(mfrow = c(2, 2), mar = c(6, 6, 1.5, 1.5))
plot(ols_model)
```



e

The regression outputs and plots suggests that the model is statistically significant overall, but may not fully fit with the dataset. The F-statistics and interaction term suggests the overall significance. The plots illustrates potential non-linearity, non-normality, and variance in residuals, which means the model is not the perfect fit.

Appendix

I used ChatGPT in this assignment. I used the tool to write annotations fore question 2 and debug the codes for question 4, and I think it is generally helpful in this case. https://chatgpt.com/share/6785e39b-2fa0-8013-8470-426366a30ddc