1.

a. log-likelihood function $\ln L(\beta, \sigma^2) = -\frac{n}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(Y_i - X_i'\beta)^2$

$$= -\frac{n}{2}\ln(2\pi\sigma^2) - SSE(b)\cdot\frac{1}{2\sigma^2}$$

$$\Rightarrow \tilde{\beta}_{CLS} = \underset{Rb=\theta_0}{\operatorname{argmin}}\ SSE(b)$$

so, in the constrain of $Rb=\theta_0$, we optimize $\tilde{\beta}_{CLS}$ by Lagrange function

$L(b,\lambda) = SEE(b) + \lambda'(Rb-\theta_0)$ $\quad\lambda$ is a $q\times 1$ vector

$$\Rightarrow \nabla_b L(b,\lambda) = \frac{\partial}{\partial b}\left(SSE(b) + \lambda'(Rb-\theta_0)\right)$$

$$= \frac{\partial}{\partial b}\left(\sum_{i=1}^{n}(Y_i - X_i'b)^2 + \lambda'(Rb-\theta_0)\right)$$

$b'X_iY_i = 1\times 1$ matrix

$\Rightarrow b'X_iY_i = Y_i'X_i'b$

$$= \frac{\partial}{\partial b}\left(\sum_{i=1}^{n}(Y_i^2 - b'X_iY_i - Y_i'X_i'b + b'X_iX_i'b) + \lambda'(Rb-\theta_0)\right)$$

$$= \frac{\partial}{\partial b}\left(Y'Y - 2b'X'Y + b'X'Xb + \lambda'Rb - \lambda'\theta_0\right)$$

$\lambda'R = 1\times 1$ matrix $\Rightarrow R\lambda' = \lambda'R$

$$= -2X'Y + 2X'Xb + \lambda'R = 0$$

$$\nabla_\lambda L(b,\lambda) = Rb - \theta_0 = 0$$

$\Rightarrow$ for optimized $b = \tilde{\beta}_{CLS}$, $\tilde{\beta}_{CLS} = -(X'X)^{-1}R'\lambda\cdot\frac{1}{2} + (X'X)^{-1}X'Y$

$$= \hat{\beta} - (X'X)^{-1}R'\lambda\cdot\frac{1}{2}$$

Also, $R\tilde{\beta}_{CLS} = R\hat{\beta} - R(X'X)^{-1}R'\lambda\frac{1}{2} = \theta_0$

$\Rightarrow \lambda = 2(R(X'X)R')^{-1}(R\cdot\hat{\beta} - \theta_0) \Rightarrow \tilde{\beta}_{CLS} = \hat{\beta} - (X'X)^{-1}\cdot R'(R(X'X)^{-1}R')^{-1}\cdot(R\cdot\hat{\beta} - \theta_0)$ #

b.

by H_0, $\theta_0 = R\beta$

$$n^{1/2}\cdot(\hat{\beta}-\hat{\beta}_{CLS}) = n^{1/2}(X'X)^{-1}R'(R(X'X)^{-1}R')^{-1}(R\hat{\beta}-\theta_0) \quad \nearrow R\beta$$

$$= (X'X)^{-1}R'(R(X'X)^{-1}R')^{-1}R\cdot\sqrt{n}\cdot(\hat{\beta}-\beta)$$

$$= \left(\tfrac{1}{n}X'X\right)^{-1}R'\left(R\left(\tfrac{1}{n}X'X\right)^{-1}R'\right)^{-1}R\cdot\sqrt{n}(\hat{\beta}-\beta)$$

converges to $E[X'X]^{-1}R'(R\cdot E[X'X]^{-1}R')^{-1}R\cdot\sqrt{n}(\hat{\beta}-\beta)$ as $n\to\infty$

Also, as $n\to\infty$, $\sqrt{n}(\hat{\beta}-\beta) \xrightarrow{d} N(0, \sigma^2 E[X'X]^{-1})$ since $var[\hat{\beta}|X] = \sigma^2(X'X)^{-1}$ under condition $E[ee'|X] = \sigma^2 I_n$

So we have asymptotic distribution as follow:

$$\Rightarrow n^{1/2}(\hat{\beta}-\hat{\beta}_{CLS}) = M\cdot\sqrt{n}(\hat{\beta}-\beta) \xrightarrow{d} N(0, \sigma^2 ME[X'X]^{-1}M')$$

where $M = E[X'X]^{-1}R'(R\cdot E[X'X]^{-1}R')^{-1}R$

c.

H_0: $R\hat{\beta}_{CLS} = \theta_0$ and test static is $\sqrt{n}(\hat{\beta}-\hat{\beta}_{CLS}) \xrightarrow{d} N(0, \sigma^2 ME[X'X]^{-1}M')$

$\sqrt{n}(\hat{\beta}-\hat{\beta}_{CLS}) = M\cdot\sqrt{n}\cdot(\hat{\beta}-\beta)$ $\quad V_\beta = \sigma^2\cdot E[X'X]^{-1}$ $\quad R = M$ $\quad V_\theta = M\hat{V}_\beta M'$

by Wald's test

$$\Rightarrow W: n(\hat{\beta}-\hat{\beta}_{CLS})'M\sigma^2\cdot E[X'X]^{-1}M'(\hat{\beta}-\hat{\beta}_{CLS}) \xrightarrow{d} \chi^2(q) \quad \text{as } n\to\infty$$

d.

H_1: $R\beta = \theta_0 + n^{-1/2}\cdot h$

Test static $T = (R\hat{\beta}-\theta_0)'(R(X'X)^{-1}R')^{-1}(R\hat{\beta}-\theta_0)\cdot n$

since $\sqrt{n}(\hat{\beta}-\beta) \xrightarrow{d} N(0, \sigma^2(X'X)^{-1})$ as $n\to\infty$

$R\hat{\beta}-\theta_0 = R\hat{\beta} - R\beta + n^{-1/2}h$

$$\Rightarrow \sqrt{n}(R\hat{\beta}-\theta_0) = \sqrt{n}R(\hat{\beta}-\beta) + h \xrightarrow{d} N(h, \sigma^2 R(X'X)^{-1}R')$$

$\Rightarrow T = (\sqrt{n}(\hat{\beta}-\beta)+h)'(R(X'X)^{-1}R')^{-1}(\sqrt{n}(\hat{\beta}-\beta)+h)$

$\Rightarrow$ since $\sqrt{n}(\hat{\beta}-\beta) \xrightarrow{d} N(0, \sigma^2(X'X)^{-1})$ as $n\to\infty$, by Wald's test, T converges to a non central chi-square distribution with $\lambda = h'(R(X'X)^{-1}R')^{-1}h$

$\Rightarrow$ asymptotic distribution of T : $\chi^2_q(\lambda)$ with $\lambda = h'(R(X'X)^{-1}R')^{-1}h$ #

2. Simulation results

Code for data setup

```r
set.seed(12345)

n_values <- c(50, 100, 200, 500)
k <- 5
R <- matrix(c(1, 0, 0, 0, 0,
              1, 1, 0, 0, 0), nrow = 2, byrow = TRUE)
theta_0 <- c(1, 2)
alpha <- 0.05
replications <- 1000

library(MASS)

generate_data <- function(n, k) {
    data <- mvrnorm(n, mu = rep(0, k + 1), Sigma = diag(k + 1))
    #the first k columns are the X's, the last column is the e
    X <- data[, 1:k]
    e <- data[, k + 1]
    list(X = X, e = e)
}

compute_test_statistic <- function(X, Y, R, theta_0) {
  beta_hat <- solve(t(X) %*% X) %*% t(X) %*% Y
  sigma_hat <- sum((Y - X %*% beta_hat)^2) / (nrow(X) - ncol(X))
  V_theta_hat <- sigma_hat * R %*% solve(t(X) %*% X) %*% t(R)

  T <- t(R %*% beta_hat - theta_0) %*% solve(V_theta_hat) %*% (R %*% beta_hat - theta_0)
  as.numeric(T)
}

critical_value <- qchisq(1 - alpha, df = 2)
```

Simulation code

```r
simulation <- function(n, R, theta_0, beta, critical_value, replications) {
  rejections <- 0
  for (i in 1:replications) {
    data <- generate_data(n, k)
    X <- data$X
    e <- data$e
    Y <- X %*% beta + e
    T <- compute_test_statistic(X, Y, R, theta_0)
    if (T > critical_value) {
      rejections <- rejections + 1
    }
  }
  rejections / replications
}
```

```r
# Run the simulation for each n to get the empirical size
beta_alt_2a <- rep(1, k)
size_results <- sapply(n_values, function(n) simulation(n, R, theta_0, beta_alt_2a, critical_value, replications))
results_2a <- data.frame(n = n_values, size = size_results)
cat("Empirical size for each n:\n")
print(results_2a)

# Run the simulation for each n with beta = (1, 2, 3, ..., k)
beta_alt_2b <- 1:k
power_results_2b <- sapply(n_values, function(n) simulation(n, R, theta_0, beta_alt_2b, critical_value, replications))
cat("Empirical power for each n with beta = (1, 2, 3, ..., k):\n")
print(data.frame(n = n_values, power = power_results_2b))

# Range of h values for Problem 2c
h_values <- 1:10

# Run the simulation for each n and each h in 2c
power_results_2c <- sapply(n_values, function(n) {
  sapply(h_values, function(h) {
    beta <- rep(1 + h / sqrt(n), k)  # Construct beta for each h and n
    simulation(n, R, theta_0, beta, critical_value, replications)
  })
})

power_results_2c <- t(power_results_2c)

# Convert power results to a more readable format
colnames(power_results_2c) <- paste("h =", h_values)
rownames(power_results_2c) <- paste("n =", n_values)
cat("Empirical power for each n and h with beta = (1 + n^(-1/2) * h, 2, 3, ..., k):\n")
print(data.frame(power_results_2c))
```

Results

```
Empirical size for each n:
    n  size
1  50 0.059
2 100 0.059
3 200 0.040
4 500 0.051
Empirical power for each n with beta = (1, 2, 3, ..., k):
    n power
1  50     1
2 100     1
3 200     1
4 500     1
Empirical power for each n and h with beta = (1 + n^(-1/2) * h, 2, 3, ..., k):
        h...1 h...2 h...3 h...4 h...5 h...6 h...7 h...8 h...9 h...10
n = 50  0.220 0.672 0.944 0.998     1     1     1     1     1      1
n = 100 0.230 0.690 0.961 1.000     1     1     1     1     1      1
n = 200 0.215 0.696 0.961 1.000     1     1     1     1     1      1
n = 500 0.227 0.730 0.975 0.999     1     1     1     1     1      1
```

- The empirical size results show that the test maintains an appropriate Type I error rate across different sample sizes.

- The test has high power against a clear alternative hypothesis, suggesting it can reliably detect substantial deviations from the null hypothesis.

- For the local alternative, power depends on both h (the magnitude of the deviation) and n (sample size):
  - Small deviations (low h) result in low power, especially for smaller sample sizes.
  - As h or n increases, power improves, indicating that larger sample sizes allow the test to detect smaller deviations from the null.

These results are consistent with expectations in statistical hypothesis testing, where larger sample sizes increase the power to detect subtle differences.