

Stat 570 HW2

Dongyang Wang

2022-10-11

Question 1

```
## Loading required package: stats4
##
## Attaching package: 'sn'
##
## The following object is masked from 'package:stats':
## 
##     sd
```

The following tables include with sample size 15, the results for bias, variance of the standard deviation, sampling distribution estimate for variance, and coverage.

```
##           bias      var  var.hat cover
## [1,] 0.062121129 32.81684167 33.7666519 0.8028
## [2,] 0.002939479  0.08125637  0.0836765 0.8041
##
##           bias      var  var.hat cover
## [1,] 0.09776866 69.0714103 70.1947871 0.7994
## [2,] 0.00476274  0.1705574  0.1739998 0.8025
##
##           bias      var  var.hat cover
## [1,] 1.5699001501 3.289790979 3.274532106 0.6298
## [2,] 0.0003015705 0.008139971 0.008121563 0.8034
```

The following tables include with sample size 40, the results for bias, variance of the standard deviation, sampling distribution estimate for variance, and coverage.

```
##           bias      var  var.hat cover
## [1,] 0.054099577 10.91519966 10.90921282 0.8052
## [2,] 0.002734042  0.02698626  0.02701367 0.8056
##
##           bias      var  var.hat cover
## [1,] 0.07115467 22.31167010 22.7678050 0.8024
## [2,] 0.00409232  0.05504602  0.0563858 0.8049
##
##           bias      var  var.hat cover
## [1,] 1.5730022668 1.041788596 1.059416243 0.3941
## [2,] 0.0003786393 0.002583686 0.002622647 0.8089
```

Based on the results above,

a

The biases are 0.

b

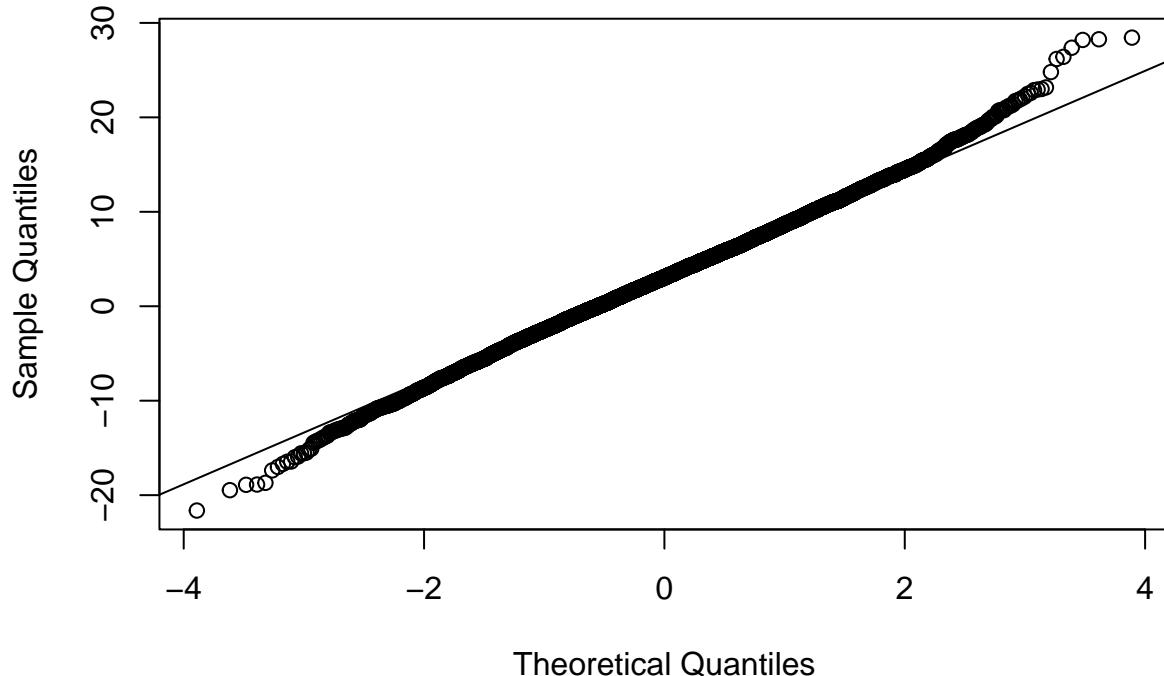
The variance converges.

c

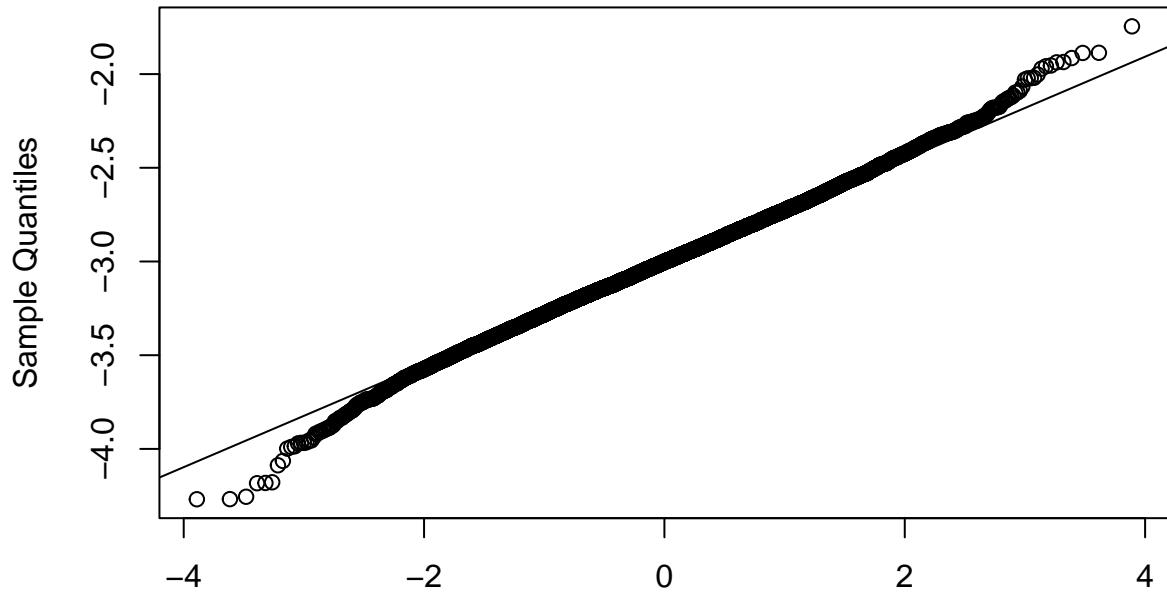
The coverage has been indicated by the variable cover. Normality will be checked as follows:

With sample size 15,

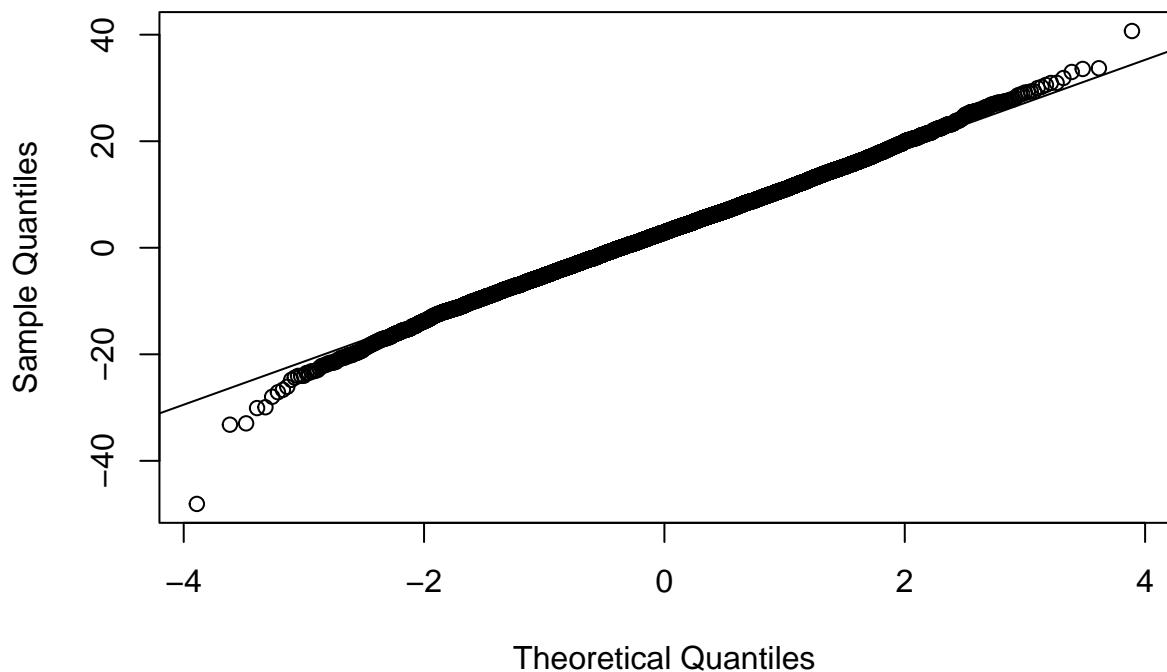
Normal Error Beta 0 Q-Q Plot



Normal Error Beta 1 Q-Q Plot

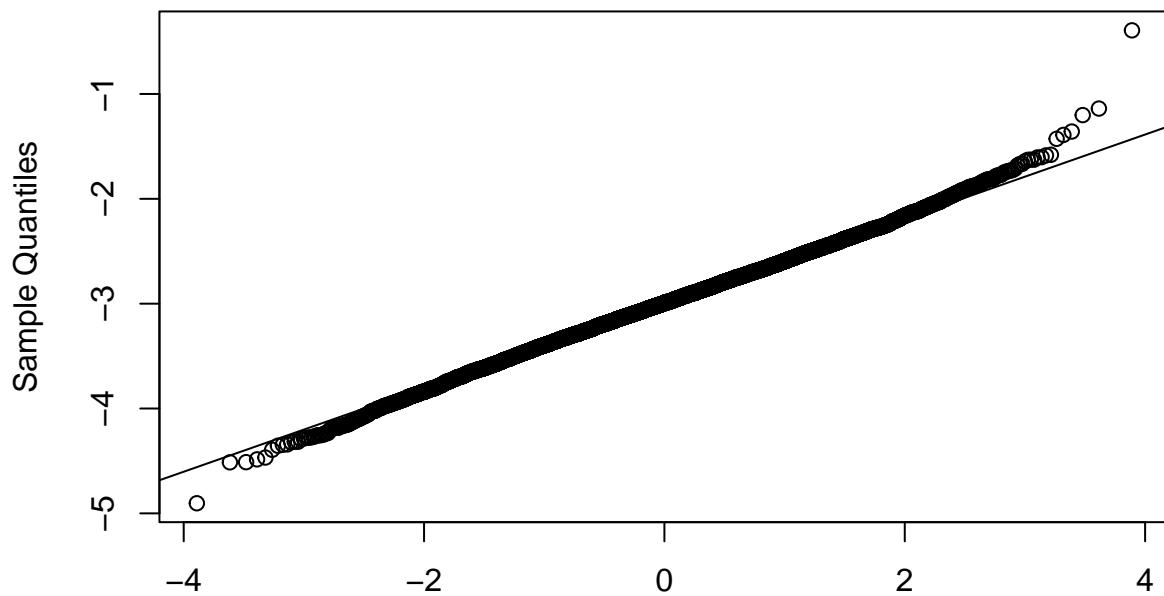


Theoretical Quantiles
Uniform Error Beta 0 Q-Q Plot

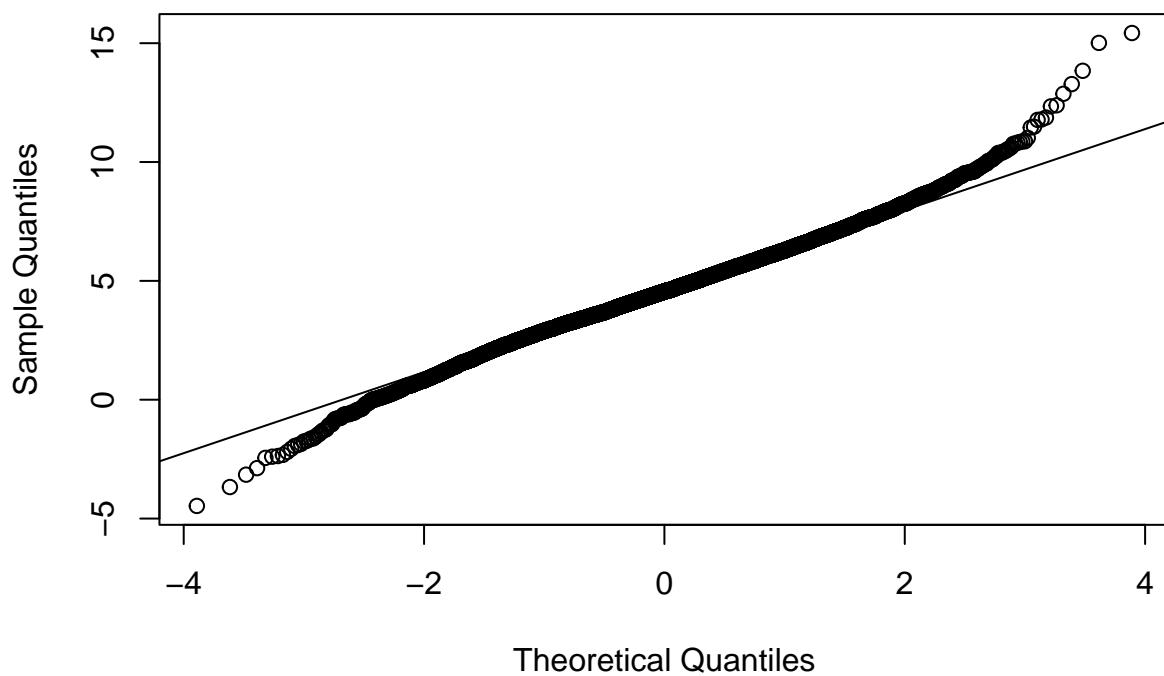


Theoretical Quantiles

Uniform Error Beta 1 Q–Q Plot

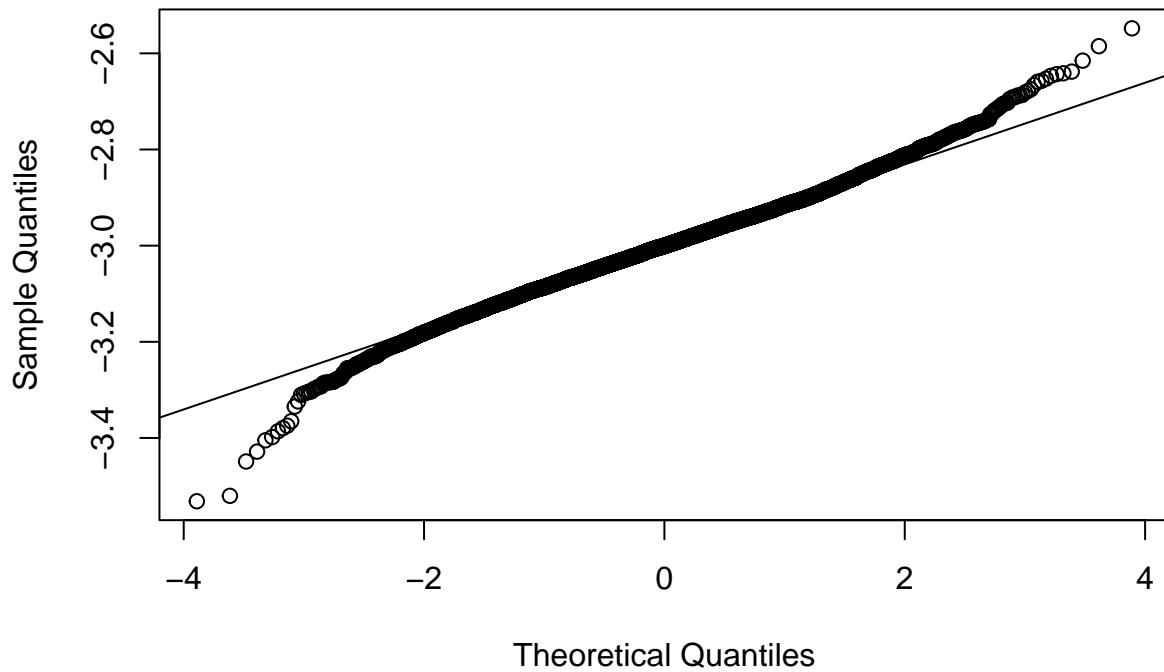


Theoretical Quantiles
Skew Error Beta 0 Q–Q Plot



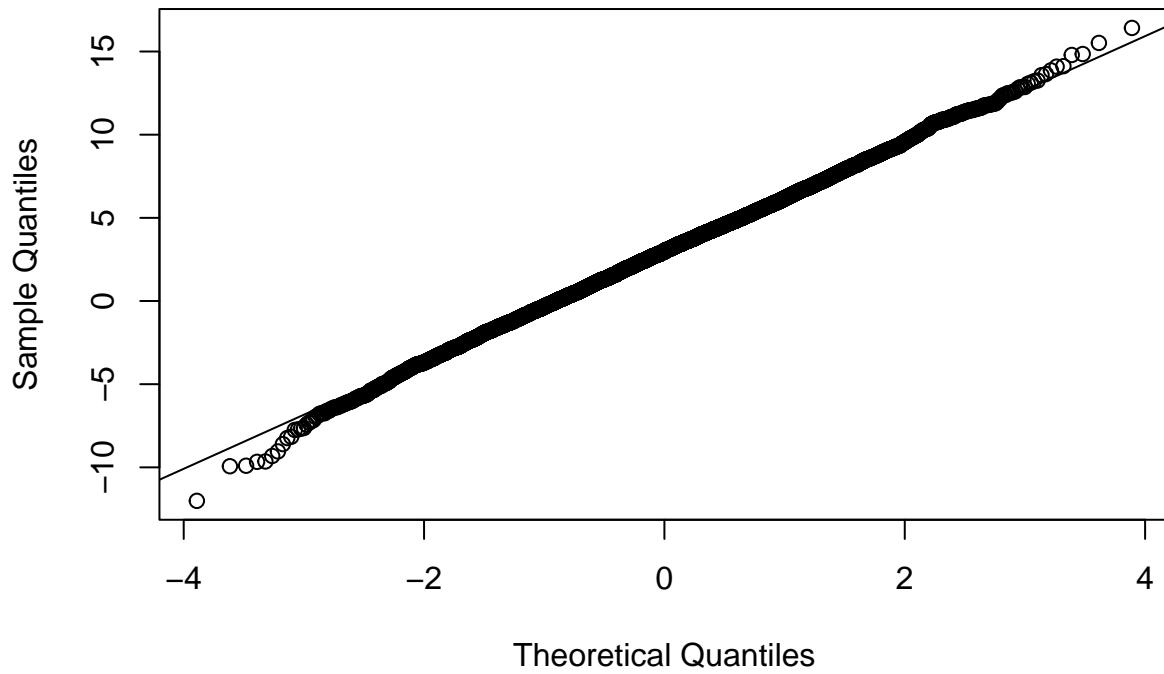
Theoretical Quantiles

Skew Error Beta 1 Q-Q Plot

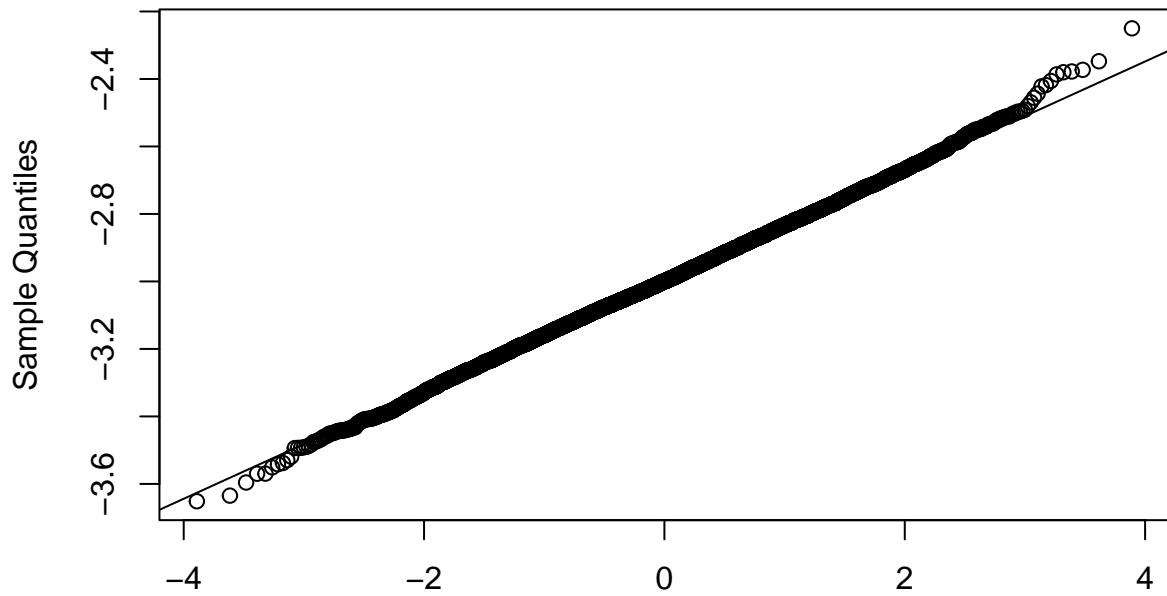


With sample size 40,

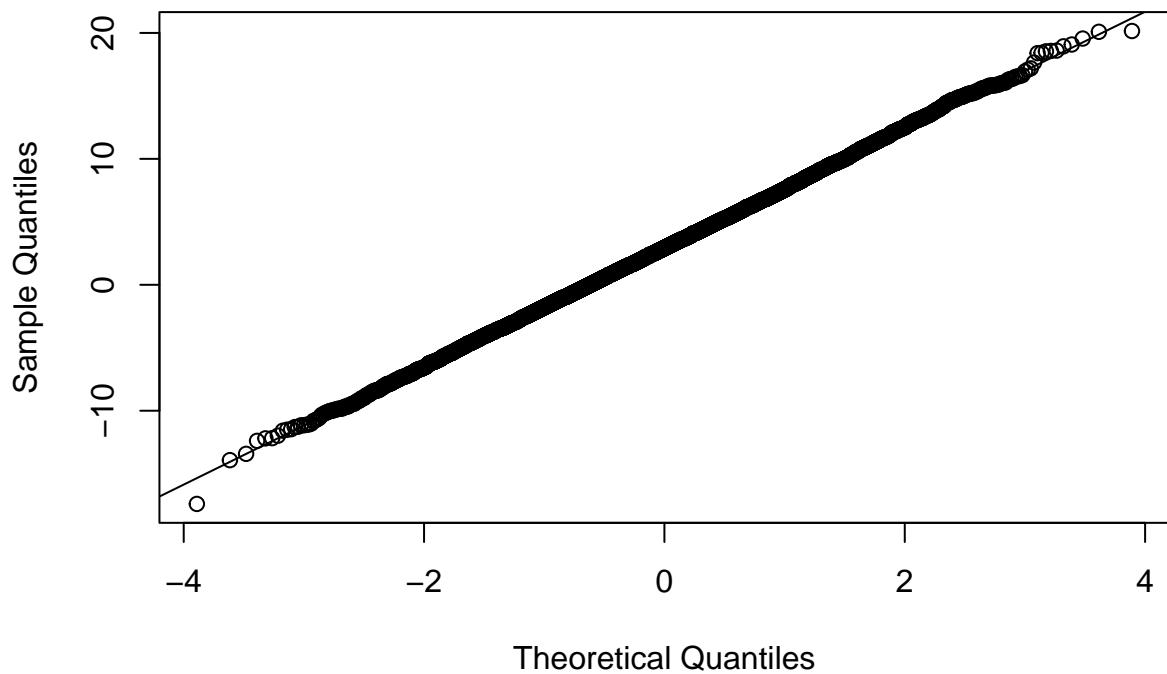
Normal Error Beta 0 Q-Q Plot



Normal Error Beta 1 Q-Q Plot

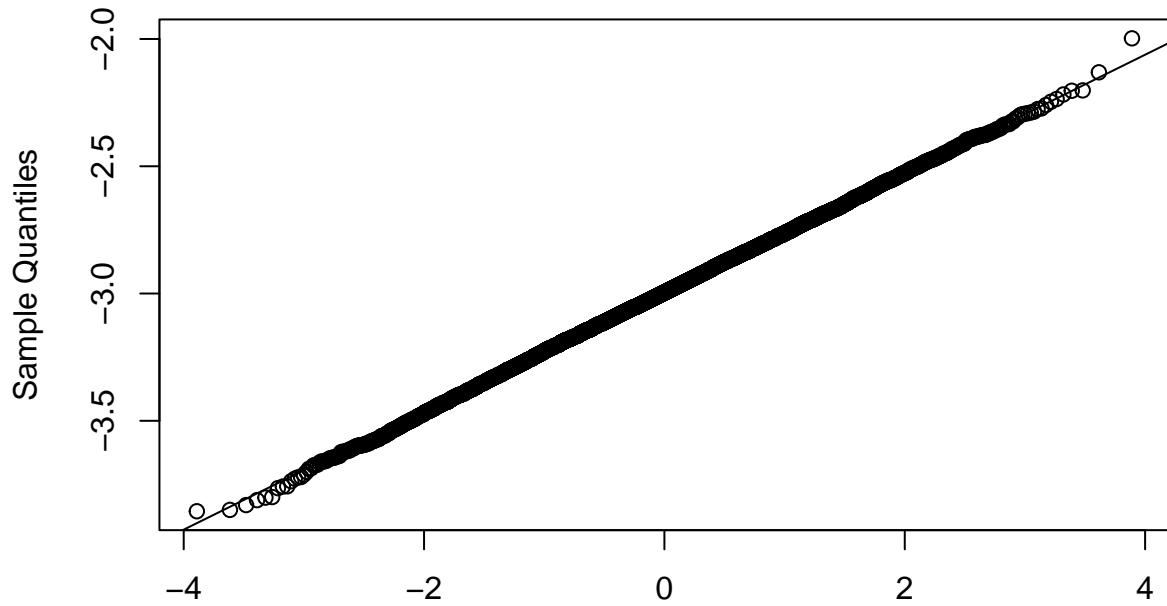


Theoretical Quantiles
Uniform Error Beta 0 Q-Q Plot

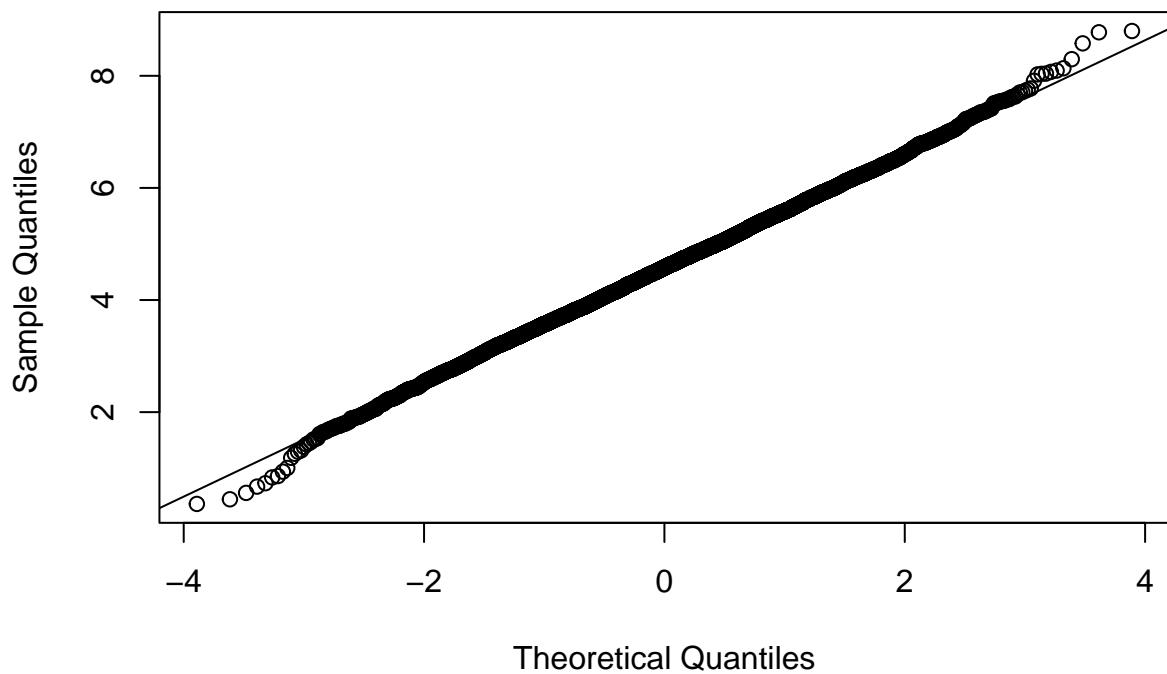


Theoretical Quantiles

Uniform Error Beta 1 Q–Q Plot

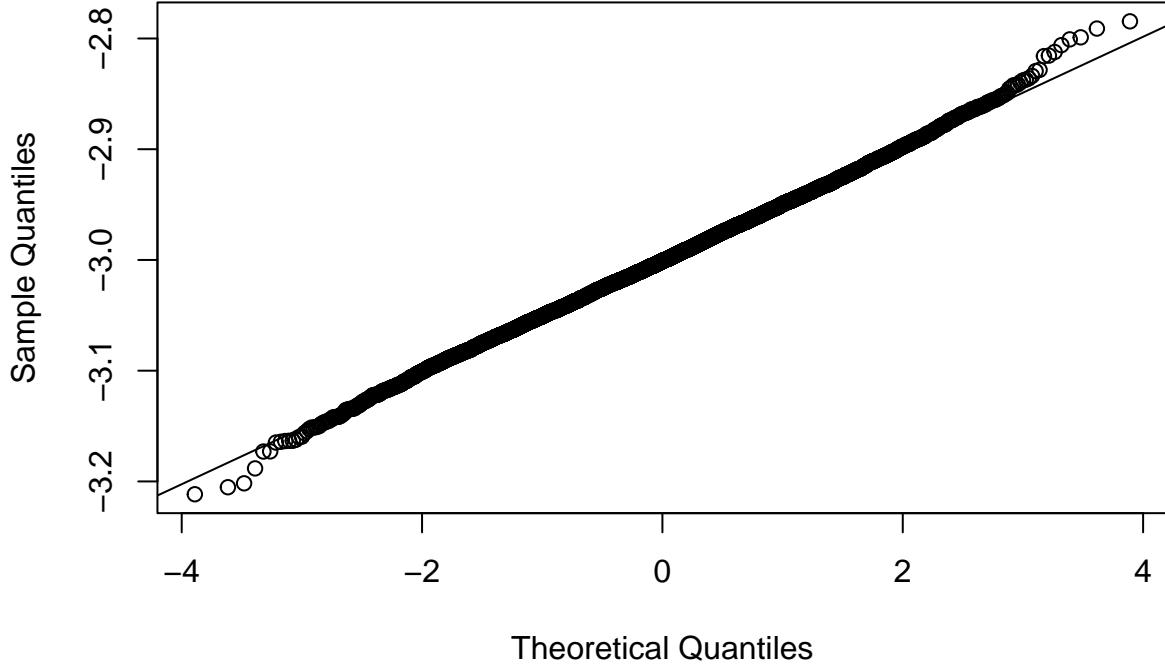


Theoretical Quantiles
Skew Error Beta 0 Q–Q Plot



Theoretical Quantiles

Skew Error Beta 1 Q-Q Plot



d

The skewed normal errors seem to break the least squares, although the expectation is 0 but the confidence interval fails to cover the true parameters 80% of the time.

Question 2

a

Likelihood function $L(\beta) = \prod_{i=1}^n \lambda_i e^{-\lambda_i y_i} = \prod_{i=1}^n \exp(\beta_0 + \beta_1 x_i - e^{\beta_0 + \beta_1 x_i} y_i)$

Log likelihood function $l(\beta) = \sum_{i=1}^n \log(\lambda_i) - \lambda_i y_i = \sum_{i=1}^n \beta_0 + \beta_1 x_i - e^{\beta_0 + \beta_1 x_i} y_i = n\beta_0 + \beta_1 \sum_{i=1}^n x_i - e^{\beta_0} \sum_{i=1}^n e^{\beta_1 x_i} y_i$

Score function Taking derivative wrt β_0 , $\frac{dl(\beta)}{d\beta_0} = n - e^{\beta_0} \sum_{i=1}^n e^{\beta_1 x_i} y_i$; Taking derivative wrt β_1 , $\frac{dl(\beta)}{d\beta_1} = \sum_{i=1}^n x_i - e^{\beta_0} \sum_{i=1}^n x_i e^{\beta_1 x_i} y_i$

FIM With $E(y_i) = \frac{1}{\lambda_i} = e^{-\beta_0 - \beta_1 x_i}$,

Taking derivative wrt score function: $I_{11} = -E(-e^{\beta_0} \sum_{i=1}^n e^{\beta_1 x_i} y_i) = e^{\beta_0} \sum_{i=1}^n e^{\beta_1 x_i} E(y_i) = n$,

$I_{12} = -E(-e^{\beta_0} \sum_{i=1}^n x_i e^{\beta_1 x_i} y_i) = e^{\beta_0} \sum_{i=1}^n x_i e^{\beta_1 x_i} E(y_i) = \sum_{i=1}^n x_i$,

$I_{21} = -E(-e^{\beta_0} \sum_{i=1}^n x_i e^{\beta_1 x_i} y_i) = e^{\beta_0} \sum_{i=1}^n x_i e^{\beta_1 x_i} E(y_i) = \sum_{i=1}^n x_i$,

$I_{22} = -E(-e^{\beta_0} \sum_{i=1}^n x_i^2 e^{\beta_1 x_i} y_i) = e^{\beta_0} \sum_{i=1}^n x_i^2 e^{\beta_1 x_i} E(y_i) = \sum_{i=1}^n x_i^2$

b

Setting score functions to 0, $n - e^{\beta_0} \sum_{i=1}^n e^{\beta_1 x_i} y_i = 0$ renders $\hat{\beta}_0 = \log \frac{n}{\sum_{i=1}^n e^{\beta_1 x_i} y_i}$ and $0 = \sum_{i=1}^n x_i - e^{\beta_0} \sum_{i=1}^n x_i e^{\beta_1 x_i} y_i$ and there is no closed form for $\hat{\beta}_1$

c

The likelihood is

mle

```
##      [,1]      [,2]
## [1,] -2.82535 0.3012588
```

The Covariance (theoretical) and confidence interval are

theoretical_cov

```
##      [,1]      [,2]
## [1,] 0.34504173 -0.046292970
## [2,] -0.04629297  0.007698387
```

theoretical_ci

```
##      [,1]      [,2]
## [1,] -3.9766378 -1.6740622
## [2,]  0.1292906  0.4732269
```

The Covariance (empirical) and confidence interval are

empirical_cov

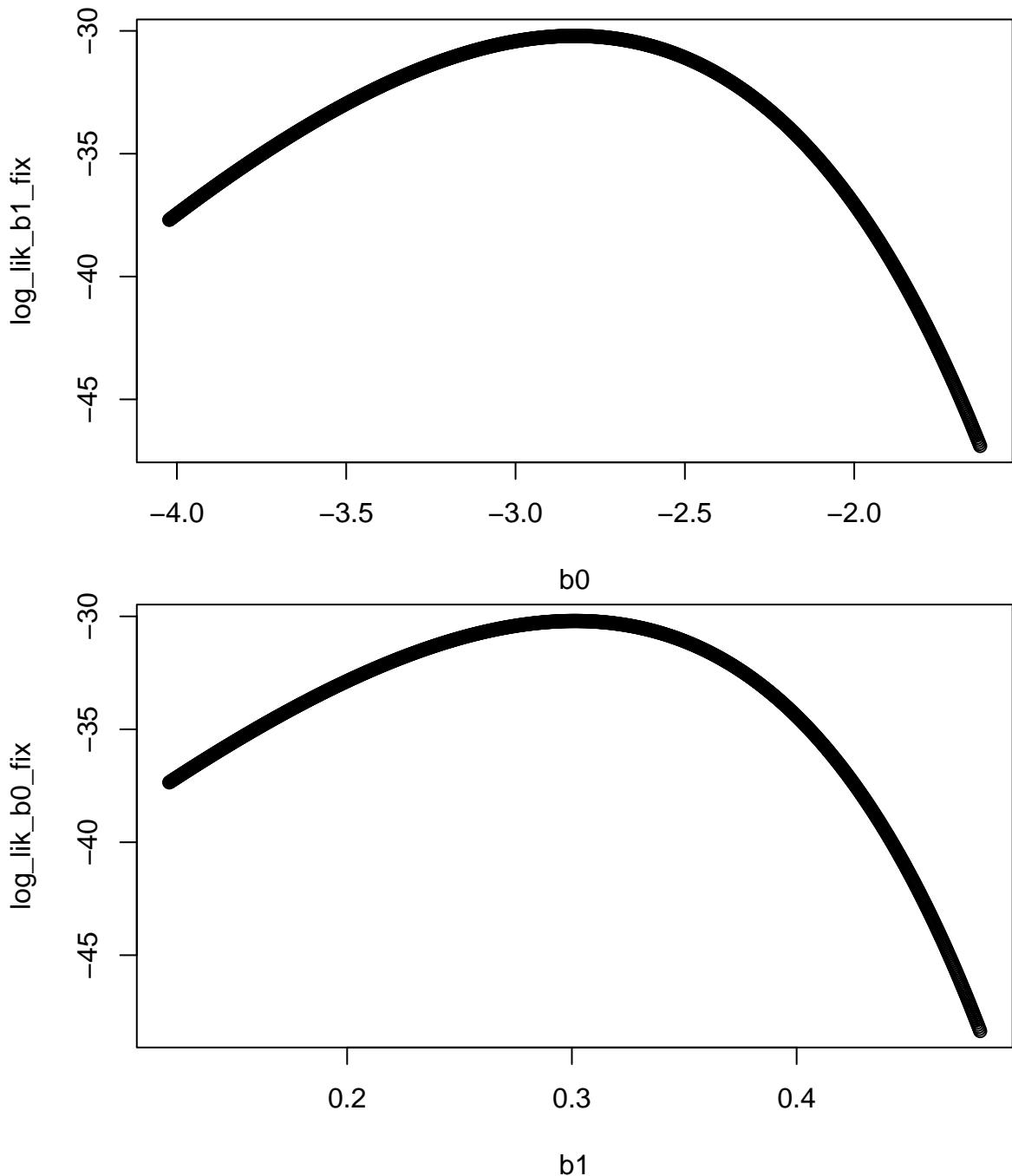
```
##      [,1]      [,2]
## [1,] 0.37309517 -0.050951186
## [2,] -0.05095119  0.008472349
```

empirical_ci

```
##      [,1]      [,2]
## [1,] -4.0225257 -1.6281742
## [2,]  0.1208532  0.4816644
```

d

We can observe the distribution of the log likelihood as we fix beta 0 or beta 1.



e

Plugging in, $\hat{\beta}_0 = \log \frac{n}{\sum_{i=1}^n e^{\beta_1 x_i} y_i} = \log \frac{n}{\sum_{i=1}^n y_i}$ with value
`## [1] -1.444563`

f

The following are chi-squared, LR test, and Wald test p-values.

`chi_pvalue`

```

## [1] 0
likeli_pvalue

## [1] 0.000323146
wald_pvalue

## [1] 0.0001348559

```

g

Based on the results above, three tests all show that the predictor significantly matters and we reject the null hypothesis that the contaminant is not associated with survival time. In terms of estimation, one unit increase in the concentration of the contaminant is associated with a 0.3012588 increase in $\log \lambda_i$, i.e., a $e^{-0.3012588} = 0.7398863$ proportion of the original survival time (without that unit of contamination).

Appendix

Q1

```

rm(list = ls())
library(sn)
set.seed(42)

n1 <- 15
n2 <- 40
beta <- c(3, -3)
methods = c("norm", "unif", "skew")

gen_bias_var <- function(n = n1, method, b0 = beta[1], b1 = beta[2]){

  x <- rnorm(n, 20, 2)

  # Epsilon
  if (method == "norm"){
    epsilon <- rnorm(n, 0, 2)
  }
  else if (method == "unif"){
    epsilon <- runif(n, -5, 5)
  }
  else if (method == "skew"){
    epsilon <- rsn(n = n, xi=5/sqrt(1+5^2)*sqrt(2/pi), omega=1, alpha=5)
  }

  y <- b0 + b1 * x + epsilon

  X <- cbind(1, x)

  # Q1
  beta_hat <- solve(t(X) %*% X) %*% t(X) %*% y

  # Q2
  y_hat <- X %*% beta_hat
  sig_hat <- sum((y-y_hat)^2)/(n-2)
  beta_sd <- sqrt(diag(sig_hat * solve(t(X)%*%X)))
}

```

```

    return(list(beta = beta_hat, sd = beta_sd))
}

gen_one <- function(n){
  result = matrix(NA, nrow = 4, ncol = 3)
  colnames(result) = methods

  for (i in methods){
    res = gen_bias_var(n, method = i)
    result[,i] = c(res$beta, res$sd)
  }

  return(as.vector(result))
}

result_15 <- replicate(10000, gen_one(15))
result_40 <- replicate(10000, gen_one(40))

gen_result <- function(result, method, n){
  if(method=="norm") {
    res <- result[1:4,]
  }else if(method=="unif") {
    res <- result[5:8,]
  }else if(method=="skew") {
    res <- result[9:12,]
  }

  t.stat <- res[1:2,]/res[3:4,]
  ci1 <- res[1,] + res[3,] %o% c(qt(0.1,df=n-2), qt(0.9,df=n-2))
  ci2 <- res[2,] + res[4,] %o% c(qt(0.1,df=n-2), qt(0.9,df=n-2))
  cover1 <- beta[1] >= ci1[,1] & beta[1] <= ci1[,2]
  cover2 <- beta[2] >= ci2[,1] & beta[2] <= ci2[,2]

  mean <- apply(res,1,mean)
  var <- apply(res,1,var)
  var.hat <- apply(res^2,1,mean)

  final_res <- cbind(bias=abs(mean[1:2]-c(beta[1], beta[2])), var=var[1:2],
                      var.hat=var.hat[3:4], cover=c(mean(cover1),mean(cover2)))
}

return(final_res)
}

gen_result(result_15, method = "norm", n =n1)
gen_result(result_15, method = "unif", n =n1)
gen_result(result_15, method = "skew", n =n1)

gen_result(result_40, method = "norm", n =n2)
gen_result(result_40, method = "unif", n =n2)
gen_result(result_40, method = "skew", n =n2)

qqnorm(result_15[1,], main = "Normal Error Beta 0 Q-Q Plot")
qqline(result_15[1,])

```

```

qqnorm(result_15[2,], main = "Normal Error Beta 1 Q-Q Plot")
qqline(result_15[2,])

qqnorm(result_15[5,], main = "Uniform Error Beta 0 Q-Q Plot")
qqline(result_15[5,])

qqnorm(result_15[6,], main = "Uniform Error Beta 1 Q-Q Plot")
qqline(result_15[6,])

qqnorm(result_15[9,], main = "Skew Error Beta 0 Q-Q Plot")
qqline(result_15[9,])

qqnorm(result_15[10,], main = "Skew Error Beta 1 Q-Q Plot")
qqline(result_15[10,])

qqnorm(result_40[1,], main = "Normal Error Beta 0 Q-Q Plot")
qqline(result_40[1,])

qqnorm(result_40[2,], main = "Normal Error Beta 1 Q-Q Plot")
qqline(result_40[2,])

qqnorm(result_40[5,], main = "Uniform Error Beta 0 Q-Q Plot")
qqline(result_40[5,])

qqnorm(result_40[6,], main = "Uniform Error Beta 1 Q-Q Plot")
qqline(result_40[6,])

qqnorm(result_40[9,], main = "Skew Error Beta 0 Q-Q Plot")
qqline(result_40[9,])

qqnorm(result_40[10,], main = "Skew Error Beta 1 Q-Q Plot")
qqline(result_40[10,])

```

Q2

```

x <- c( 6.2, 4.2, 0.5, 8.8, 1.5, 9.2, 8.5, 8.7, 6.7, 6.5, 6.3, 6.7, 0.2, 8.7, 7.5 )
y <- c( 0.8, 3.5, 12.4, 1.1, 8.9, 2.4, 0.1, 0.4, 3.5, 8.3, 2.6, 1.5, 16.6, 0.1, 1.4)

log_likelihood <- function(x = x, y = y, beta0 = 0, beta1 = 0){
  n <- length(x)
  n*beta0 + beta1*sum(x) - sum(exp(beta0+beta1*x)*y)
}

max_lik <- function(beta){
  -log_likelihood(x,y,beta[1],beta[2])
}

# mle
mle <- matrix(optim(par=c(0,0),fn=max_lik)$par, nrow =1)

# asymptotic cov
theoretical_cov = solve(matrix(c(length(x), sum(x), sum(x), sum(x^2)), nrow = 2, ncol = 2))

```

```

theoretical_ci = matrix(c(mle - matrix(qnorm(.975)*sqrt(diag(theoretical_cov)), nrow = 1), mle + matrix(qnorm(.975)*sqrt(diag(theoretical_cov)), nrow = 1)), nrow = 2, ncol = 2)

# empirical cov

empirical_cov = solve(matrix(c(sum(exp(mle[1,1]+mle[1,2]*x)*y),
                                sum(x*y*exp(mle[1,1]+mle[1,2]*x)),
                                sum(x*y*exp(mle[1,1]+mle[1,2]*x)),
                                sum(x^2*y*exp(mle[1,1]+mle[1,2]*x))), nrow = 2, ncol = 2))
empirical_ci = matrix(c(mle - matrix(qnorm(.975)*sqrt(diag(empirical_cov)), nrow = 1), mle + matrix(qnorm(.975)*sqrt(diag(empirical_cov)), nrow = 1)), nrow = 2, ncol = 2)

b0 = seq(empirical_ci[1,1], empirical_ci[1,2], length.out = 1000)
b1 = seq(empirical_ci[2,1], empirical_ci[2,2], length.out = 1000)

log_lik_b1_fix = c()
log_lik_b0_fix = c()

for (i in 1:1000){
  log_lik_b1_fix = c(log_lik_b1_fix, log_likelihood(x,y, beta0 = b0[i], beta1 = mle[1,2]))
  log_lik_b0_fix = c(log_lik_b0_fix, log_likelihood(x,y, beta0 = mle[1,1], beta1 = b1[i]))
}

plot(b0, log_lik_b1_fix)
plot(b1, log_lik_b0_fix)

beta_0_null = log(length(x)/sum(y))
beta_0_null

# null mle

null_mle = matrix(c(beta_0_null, 0), nrow = 2)

score <- function(x, y, beta0 = beta_0_null, beta1 = 0){
  n <- length(x)
  S1 <- n - sum(exp(beta0+beta1*x)*y)
  S2 <- sum(x) - sum(x*y*exp(beta0+beta1*x))
  return(c(S1, S2))
}

score_function = matrix(score(x, y), nrow = 2)

# score test

null_cov = solve(matrix(c(sum(exp(null_mle[1,1]+null_mle[2,1]*x)*y),
                           sum(x*y*exp(null_mle[1,1]+null_mle[2,1]*x)),
                           sum(x*y*exp(null_mle[1,1]+null_mle[2,1]*x)),
                           sum(x^2*y*exp(null_mle[1,1]+null_mle[2,1]*x))), nrow = 2, ncol = 2))
chi_stat = as.vector(t(score_function) %*% solve(null_cov) %*% score_function)
chi_pvalue = 1 - pchisq(chi_stat, 1)

# likelihood ratio

likeli_stat = -2 * (log_likelihood(x,y,beta_0_null,0) - log_likelihood(x,y,mle[1,1],mle[1,2]))
likeli_pvalue = 1 - pchisq(likeli_stat, 1)

# Wald

```

```
diff_mle = t(mle) - null_mle
walt_stat = as.vector(t(diff_mle) %*% solve(theoretical_cov) %*% diff_mle)
wald_pvalue = 1 - pchisq(walt_stat, 1)

chi_pvalue
likeli_pvalue
wald_pvalue
```