Stat 570 HW4

Dongyang Wang

2022-10-21

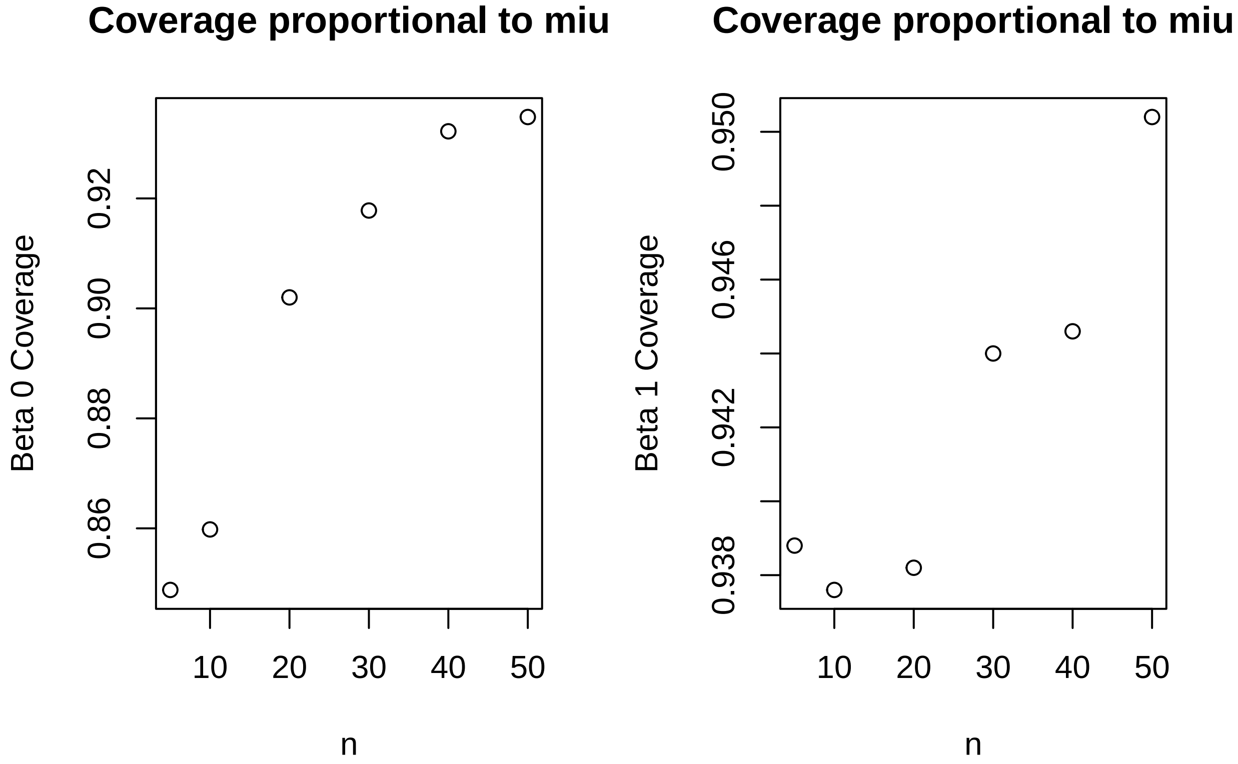
Q1

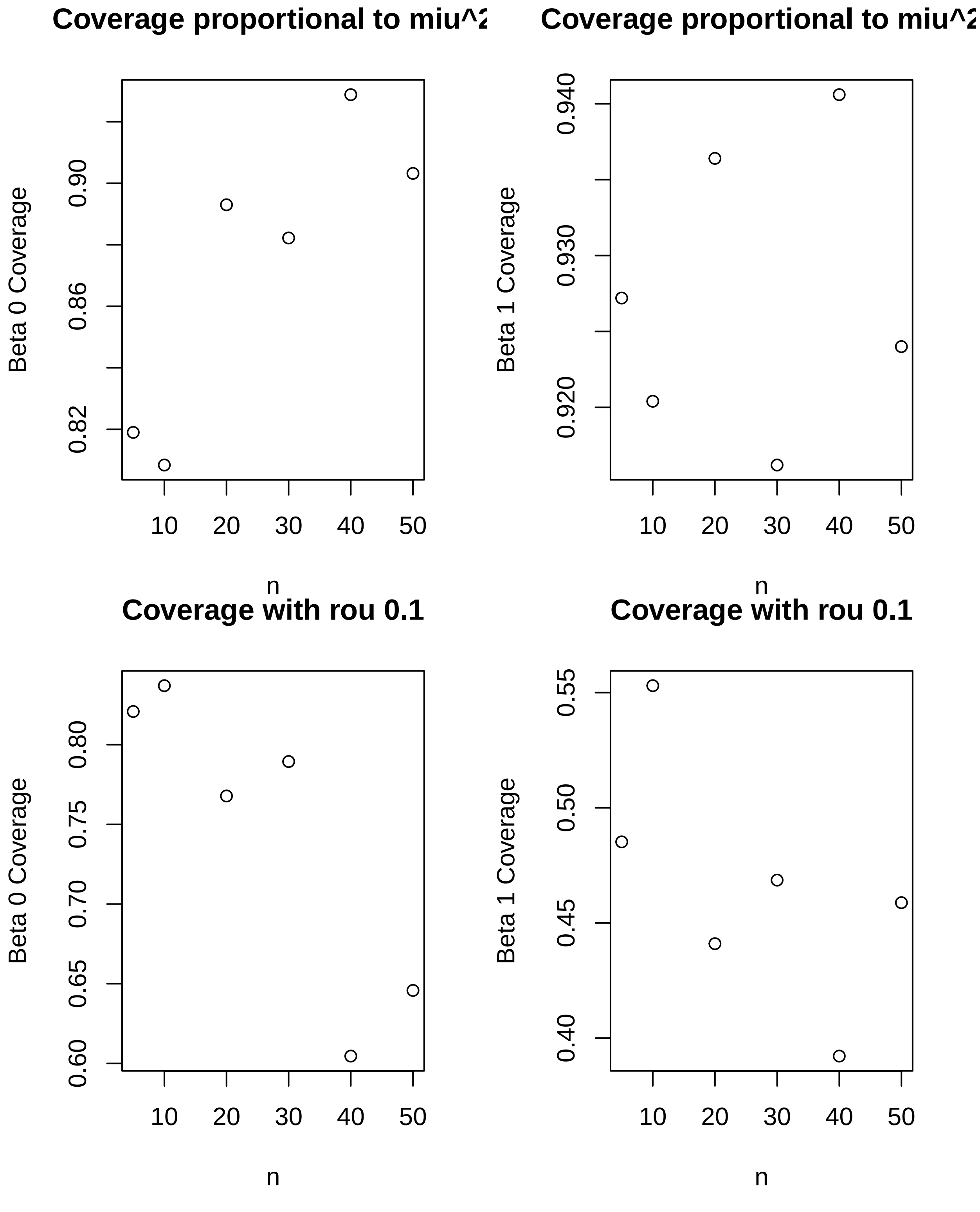
a

```
## Spam version 2.8-0 (2022-01-05) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.
```

```
##
## Attaching package: 'spam'
```

```
## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve
```



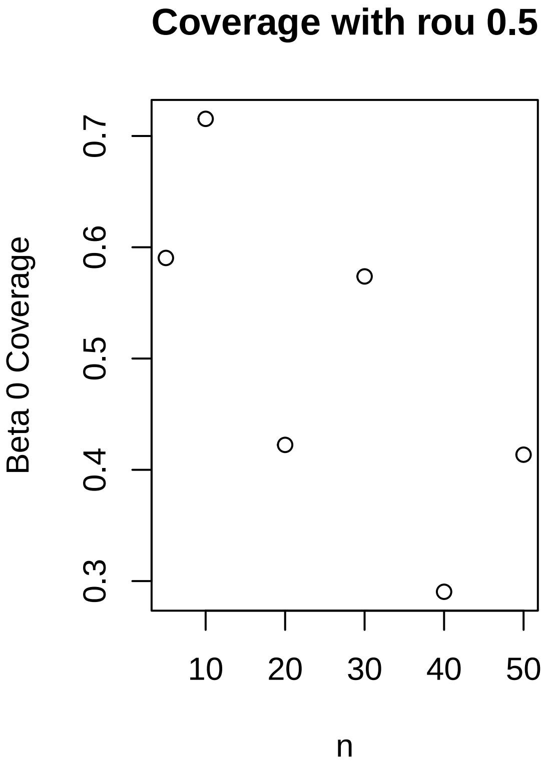


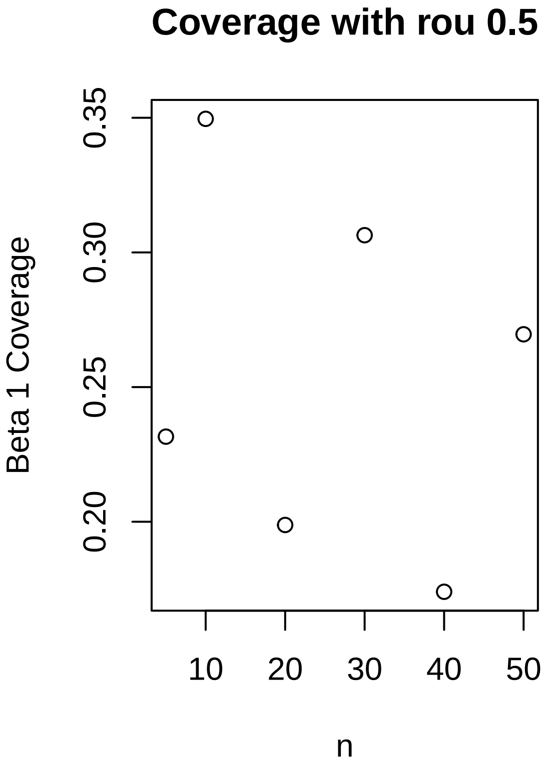
Coverage proportional to miu^2

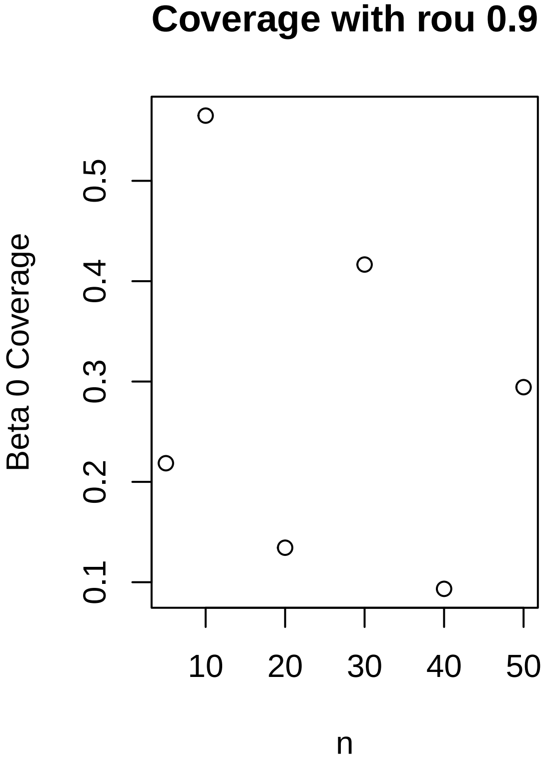
Coverage proportional to miu^2

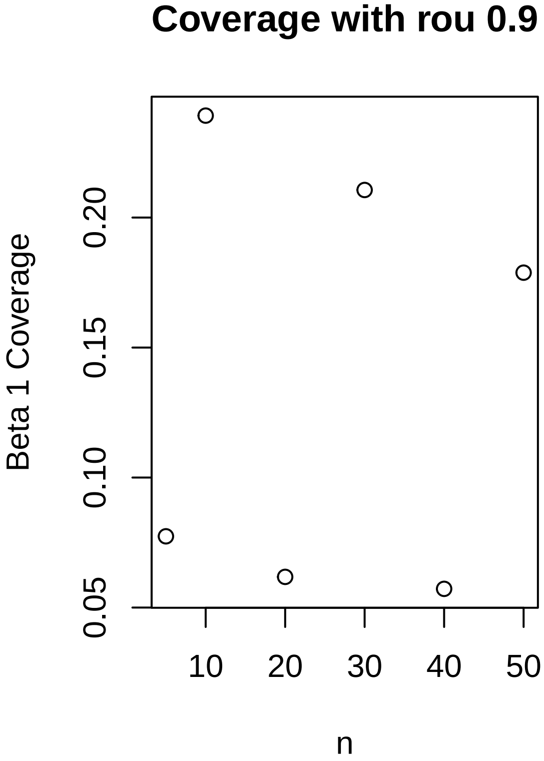
Coverage with rou 0.1

Coverage with rou 0.1









b

Based on the results above, we can observe that when the variance is correctly specified, i.e., no correlated errors, then the assumption for linear regression is met. In this way, as we increase the number of samples, we obtain better prediction results in terms of confidence interval coverage. It seems that there is more variance in the second model than in the first model, but the coverages are similar based on the simulations.

By contrast, the correlated errors case (model 3-5) all display poor coverage. And the coverage is actually worse as ρ gets larger, such that the errors are more correlated. As number of samples increase the coverage is also worse since the models are modeled based on incorrect assumption of homoskedasticity.

Q2

a

The MoM estimator is $\hat{x_k} = \frac{N}{n}Y_k$. The variance is $var(\hat{x_k}) = var(\frac{N}{n}Y_k) = \frac{N^2}{n^2}var(Y_k) = \frac{\hat{x_k}}{N}(1-\frac{\hat{x_k}}{N})\frac{N^2(N-n)}{n(N-1)} = \frac{Y}{n}(1-\frac{Y}{n})\frac{N^2(N-n)}{n(N-1)}$.

b

$p(p|X) \propto p(X|p) * \pi(p) = \frac{N!}{\Pi_{k=1}^K x_k!}\Pi_{k=1}^K p_k^{x_k} * \frac{\Gamma(\alpha_+)}{\Pi_{k=1}^K \Gamma(\alpha_k)}\Pi_{k=1}^K p_k^{\alpha_k - 1} \propto \Pi_{k=1}^K p_k^{x_k+\alpha_k-1}$ This is proportional to the Dirichlet distribution with parameter $x_k + \alpha_k$.

c

$\frac{\Gamma(\alpha_+ + \sum_{k=1}^K x_k)}{\Pi_{k=1}^K \Gamma(\alpha_k + x_k)}\Pi_{k=1}^K p_k^{x_k+\alpha_k-1} = p(p|X) = \frac{p(X|p)*\pi(p)}{p(X)} = \frac{\frac{N!}{\Pi_{k=1}^K x_k!}\Pi_{k=1}^K p_k^{x_k} * \frac{\Gamma(\alpha_+)}{\Pi_{k=1}^K \Gamma(\alpha_+)}\Pi_{k=1}^K p_k^{\alpha_k-1}}{p(X)}$

Solving this, $p(X) = \frac{N!\Gamma(\alpha_+)}{\Gamma(\alpha_+ + \sum_{k=1}^K x_k)}\Pi_{k=1}^K \frac{\Gamma(\alpha_k + x_k)}{x_k!\Gamma(\alpha_k)}$ which is compound multinomial distribution with $N = \sum_{k=1}^K x_k)$ and α is the α from the Dirichlet distribution.

d

$E(X_k) = E_p(E(X_k|p)) = E(Np_k) = N\frac{\alpha_k}{\alpha_+}$ since for the Dirichlet distribution $E(p_k) = \frac{\alpha_k}{\alpha_+}$.

$var(X_k) = var(E(X_k|p)) + E(Var(X_k|p)) = var(Np_k) + E(Np_k(1-p_k))$ since for the multinomial distribution $E(X_k|p) = Np_k$ and $var(X_k|p) = Np_k(1-p_k)$. Therefore, $var(X_k) = N^2 var(p_k) + NE(p_k) - Nvar(p_k) - NE(p_k)^2 = N^2\frac{\alpha_k(\alpha_+ - \alpha_k)}{\alpha_+^2(\alpha_+ + 1)} + N\frac{\alpha_k}{\alpha_+} - N\frac{\alpha_k(\alpha_+ - \alpha_k)}{\alpha_+^2(\alpha_+ + 1)} - N(\frac{\alpha_k}{\alpha_+})^2 = \frac{\alpha_k(\alpha_+ - \alpha_k)N(\alpha_+ + N)}{\alpha_+^2(\alpha_+ + 1)}$ since $var(p_k) = \frac{\alpha_k(\alpha_+ - \alpha_k)}{\alpha_+^2(\alpha_+ + 1)}$.

e

$p(w|y) = \frac{p(y|x=w+y)\pi(x=w+y))}{p(y)}|\frac{dx}{dw}| \propto p(y|x = w+y)\pi(x = w+y) = \frac{\Pi_{k=1}^K (w_k+y_k)!n!}{N!y_k!}\frac{N!\Gamma(\alpha_+)}{\Gamma(\alpha_+ + N)}\Pi_{k=1}^K \frac{\Gamma(\alpha_k + w_k + y_k)}{(w_k+y_k)!\Gamma(\alpha_k)} \propto \Pi_{k=1}^K \frac{(w_k+y_k)!}{w_k!y_k!}\frac{\Gamma(\alpha_k + w_k + y_k)}{(w_k+y_k)!} \propto \Pi_{k=1}^K \frac{\Gamma(\alpha_k + w_k + y_k)}{w_k!}$

Therefore it laso follows compound multinomial, with parameters $N = \sum_{k=1}^K w_k = \sum_{k=1}^K (x_k - y_k) = N - n$ and $\alpha + y$ as α.

f

Based on results from d, $E(w_k|y) = (N-n)\frac{\alpha_k + y_k}{\alpha_+ + n}$ and $E(x_k|y) = E(w_k|y) + y_k = (N-n)\frac{\alpha_k + y_k}{\alpha_+ + n} + y_k$.

Also, $var(w_k|y) = (N-n)\frac{(\alpha_k + y_k)(\alpha_+ + n - \alpha_k - y_k)(\alpha_+ + N)}{(\alpha_+ + n)^2(\alpha_+ + n + 1)}$ and $var(x_k|y) = var(w_k|y)$.

When $\alpha = 0$, the results become the same as those for the MoM estimator of x, where $E(x|y) = \frac{N}{n}Y_k$ and $var(x|y) = \frac{Y}{n}(1-\frac{Y}{n})\frac{N^2(N-n)}{n(N-1)}$.

g

The estimators and se's are as follows:

```
## [1] 507.6923 242.3077   0.0000
```

```
## [1] 41.60367 41.60367  0.00000
```

h

The posterior means and se's are as follows:

```
## [1] 497.30882 242.61765  10.07353
```

```
## [1] 40.90529 40.44823 10.40751
```

```
##                        A         B        C
## mom_mean       507.69231 242.30769  0.00000
## mom_se          41.60367  41.60367  0.00000
## posterior_mean 497.30882 242.61765 10.07353
## posterior_se    40.90529  40.44823 10.40751
```

The posterior estimates are more reasonable because 0 seems not to make sense.

i

```
## Loading required package: boot
```

```
## Loading required package: spdep
```

```
## Loading required package: sp
```

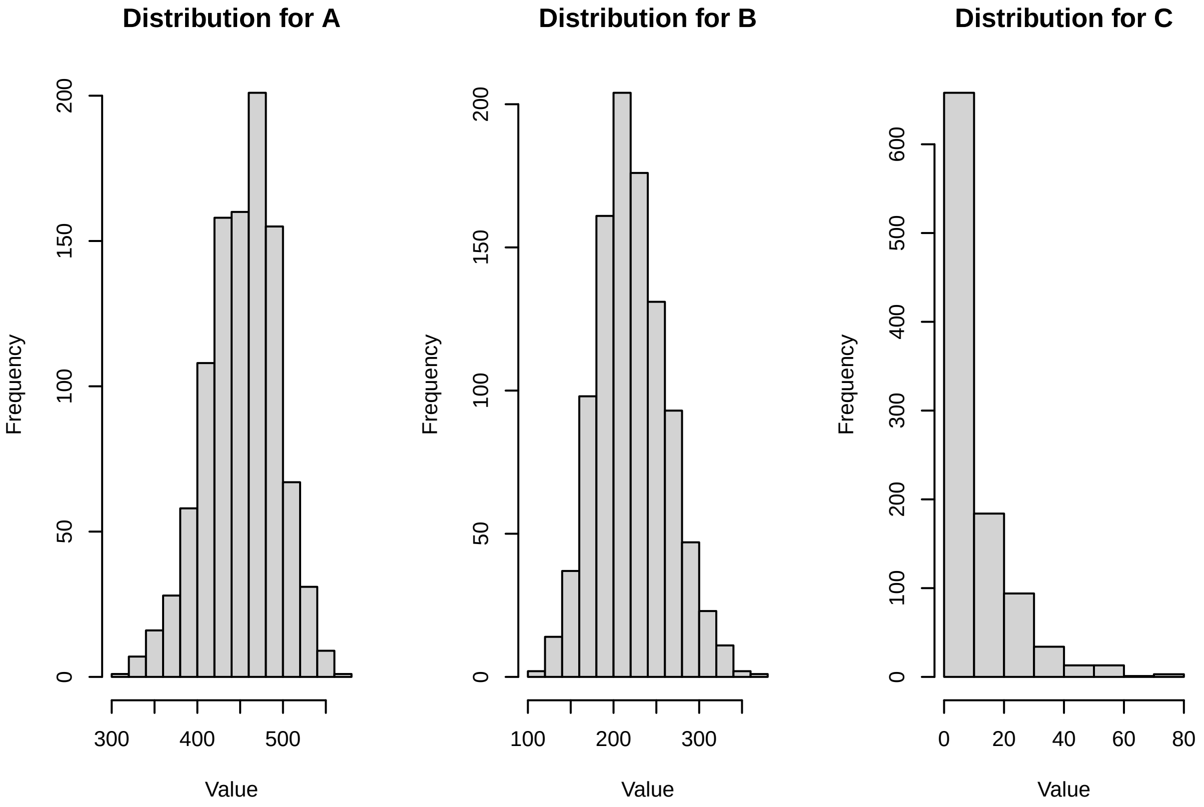
```
## Loading required package: spData
```

```
## To access larger datasets in this package, install the spDataLarge
## package with: `install.packages('spDataLarge',
## repos='https://nowosad.github.io/drat/', type='source')`
```

```
## Loading required package: sf
```

```
## Linking to GEOS 3.10.2, GDAL 3.4.2, PROJ 8.2.1; sf_use_s2() is TRUE
```

```
## Loading required package: MASS
```



Appendix

Q1

```
rm(list = ls())
library(spam)

n = c(5, 10, 20, 30, 40, 50)
range_t = c(-2,2)
beta_0 = 4
beta_1 = 1.75
sigma = 1
row = c(0.1, 0.5, 0.9)

gen_t <- function(N){
  seq(range_t[1], range_t[2], length.out =  N)
}

gen_miu <- function(t){
  beta_0 + beta_1 * (t - mean(t))
}

gen_cov <- function(N, case){
  t <- gen_t(N)
  miu_i <- gen_miu(t)

  if (case == 1){

    var_eps <- miu_i
    eps <- rnorm(N, 0, sqrt(var_eps))
    y_i <- miu_i + eps

  }

  else if (case == 2){

    var_eps <- miu_i^2
    eps <- rnorm(N, 0, sqrt(var_eps))
    y_i <- miu_i + eps

  }

  else if (case == 3){
    rho <- 0.1

    var <- c()
    for (i in 1:N){
      for (j in 1:N){
        var <- c(var, rho^(abs(t[i] - t[j])))
      }
    }

    var <- matrix(var, nrow = N, ncol = N)
    eps <- rmvnorm(1, mu=rep.int(0, N),  sigma=var)
    y_i <- miu_i + eps
    y_i <- as.vector(y_i)
  }

  else if (case == 4){
    rho <- 0.5

    var <- c()
    for (i in 1:N){
      for (j in 1:N){
        var <- c(var, rho^(abs(t[i] - t[j])))
      }
    }

    var <- matrix(var, nrow = N, ncol = N)
    eps <- rmvnorm(1, mu=rep.int(0, N),  sigma=var)
    y_i <- miu_i + eps
    y_i <- as.vector(y_i)
  }

  else if (case == 5){
    rho <- 0.9

    var <- c()
    for (i in 1:N){
      for (j in 1:N){
        var <- c(var, rho^(abs(t[i] - t[j])))
      }
    }

    var <- matrix(var, nrow = N, ncol = N)
    eps <- rmvnorm(1, mu=rep.int(0, N),  sigma=var)
    y_i <- miu_i + eps
    y_i <- as.vector(y_i)
  }

  model <- lm(y_i ~ t)
  cov1 <-  c(model$coef[1] + qnorm(0.025) * sqrt(diag(vcov(model)))[1],
             model$coef[1] + qnorm(0.975) * sqrt(diag(vcov(model)))[1])
  cov2 <-  c(model$coef[2] + qnorm(0.025) * sqrt(diag(vcov(model)))[2],
             model$coef[2] + qnorm(0.975) * sqrt(diag(vcov(model)))[2])

  return(c(cov1[1] <= beta_0 & beta_0 <= cov1[2], cov2[1] <= beta_1 & beta_1 <= cov2[2]))

}

results <- vector("list", length = 5)

for (i in 1:5){
  result <- c()

  for (j in n){
    res <- replicate(5000, gen_cov(N = j, case = i))
    mean_res <- rowMeans(res)
    result <- c(result, mean_res)
  }

  result <- matrix(result, ncol = 2)
  results[[i]] = result

}

names <- c("Coverage proportional to miu",
           "Coverage proportional to miu squared",
           "Coverage sigma squared but with rou 0.1",
           "Coverage sigma squared but  rou 0.5",
           "Coverage sigma squared but w rou 0.9")
rou <- c("", "", " 0.1", " 0.5", " 0.9")


for (i in 1:5){
  par(mfrow=c(1,2))
  plot(n,results[[i]][,1], ylab = "Beta 0 Coverage", main = names[i])
  plot(n,results[[i]][,2], ylab = "Beta 1 Coverage", main = names[i])
}
```

Q2

```
y <- c(44,21,0)
N <- 750
n <- 65
mom_x <- y*N/n
mom_x

sqrt(y/n * (1-y/n) * N^2 * (N-n)/(n*(N-1)))

alpha <- c(1,1,1)
posterior_mean <- (N-n)*(y+alpha)/(n+sum(alpha)) + y
posterior_se <- sqrt((N-n)*(alpha+y)*(sum(alpha)+n-alpha-y)*(sum(alpha)+N)/((sum(alpha)+n)^2*(sum(alpha

posterior_mean
posterior_se

table_compare <- rbind(mom_mean, mom_se, posterior_mean, posterior_se)
colnames(table_compare) <- c("A", "B", "C")
?rmultin
table_compare

#install.packages('MCMCprecision')
#install.packages('DCluster')
library(MCMCprecision)
library(DCluster)

repeat_n <- 1000
dir <- rdirichlet(repeat_n, alpha+y)
multi <- dir
```

```r
for (i in 1:repeat_n){
  multi[i,] <- rmultin(N-n,dir[i,])
}

par(mfrow = c(1,3))
hist(multi[,1], xlab = "Value", main = "Distribution for A")
hist(multi[,2], xlab = "Value", main = "Distribution for B")
hist(multi[,3], xlab = "Value", main = "Distribution for C")
```