Econ_4403_A3

Hanpeng Wang

February 26, 2019

Set up helper function for linear regression estimation

```
lm_calculation <- function(your_matrix, dependent_variables_col, independent_variables_col){
  k <- length(your_matrix[1,]) - 2
  n <- length(your_matrix[,1])
  #1. calculate betahats
  lm_betahats <- ( solve(t(your_matrix[,independent_variables_col]) %*% your_matrix[,independent_variables_col])
                   %*% t(your_matrix[,independent_variables_col]) %*% your_matrix[,dependent_variables_col])
  #2. calculate yhats
  lm_yhats <- your_matrix[,independent_variables_col] %*% lm_betahats
  #3. calculate uhats
  lm_uhats <-  your_matrix[,dependent_variables_col] - lm_yhats
  #3. calculate VCMs
  vcm_uhats <- lm_uhats %*% t(lm_uhats)
  var_uhat <- sum(t(lm_uhats) %*% lm_uhats) /  (n - k )
  vcm_betahats <- var_uhat * solve(t(your_matrix[,independent_variables_col])
                                   %*% your_matrix[,independent_variables_col])
  #4 calculate statistics
  mean_y <- mean(your_matrix[,dependent_variables_col])
  Ess <- sum((lm_yhats-mean_y)**2)
  Tss <- sum((your_matrix[,dependent_variables_col]-mean_y)**2)
  Rss <- Tss - Ess
  lm_rsqure <- Ess / Tss
  t_stats <- rbind(names(your_matrix[,independent_variables_col]),
                   lm_betahats / sqrt(diag(vcm_betahats)))
  f_stat <- (Ess / k ) / (Rss / (n - k -1))
    #return results
    return(list(betahat = lm_betahats, yhat = lm_yhats, uhat = lm_uhats, var = var_uhat,
                VCMuhat = vcm_uhats, VCMbetahat = vcm_betahats, r2 = lm_rsqure,
                tstat = t_stats, fstat = f_stat))
}
```

Part A

```
options(warn = -1)
#part A
path_1 <- '/Users/birdfly/Downloads/Andy.csv'
#q1
df_andy <- as.matrix(read.csv(path_1))
intercept <- c(rep(1,length(df_andy[,1])))
df_andy <- cbind(df_andy,intercept)
df_andy <- df_andy[,c(1,4,2,3)]
reg_andy1 <- lm_calculation(df_andy, c(1),c(2:4))
print('betahats are')
```

```
## [1] "betahats are"
```

```
reg_andy1$betahat
```

```
##                 [,1]
## intercept 118.913610
## price      -7.907854
## advert      1.862584
```

```
#q2
print('price negative affects sale and advertisement is positive, y intercept is 118')
```

```
## [1] "price negative affects sale and advertisement is positive, y intercept is 118"
```

```
#q3
log_df_andy <- log(df_andy)
log_df_andy[,2] <- intercept
reg_logandy <- lm_calculation(log_df_andy,c(1), c(2:4))
print('price and adv. elasticities are')
```

```
## [1] "price and adv. elasticities are"
```

```
reg_logandy$betahat[2:3]
```

```
## [1] -0.57493603  0.04544036
```

```
cat('when sales is 71 and price is 5, elasticity is ', reg_andy1$betahat[2] * (5/72))
```

```
## when sales is 71 and price is 5, elasticity is  -0.5491566
```

```
print('the reason of difference is because elasticity is not constant, and the one
      in log-log model is E(elasticity)')
```

```
## [1] "the reason of difference is because elasticity is not constant, and the one\n      in log-log model is E(elasticit
y)"
```

```
#q4
#f.revenue <- beta1*price + beta2 * p_sqr + beta3*adv*p
#dr/dp <- beta1 + beta2 * p + beta3*adv
print('optimal price is (beta1-beta3*adv) / beta2, and optimal sales is got from
      putting this number into estimated linear model, we have to know adv level anyways.
      Elasticity is beta2 * (optimal price / optimal sales)')
```

```
## [1] "optimal price is (beta1-beta3*adv) / beta2, and optimal sales is got from\n      putting this number into estimated
linear model, we have to know adv level anyways.\n      Elasticity is beta2 * (optimal price / optimal sales)"
```

```
print('assume adv is 0, the optimal price is')
```

```
## [1] "assume adv is 0, the optimal price is"
```

```
opt_price <- - reg_andy1$betahat[1] / (2 * reg_andy1$betahat[2])
opt_price
```

```
## [1] 7.518703
```

```
print('optimal revenue is')
```

```
## [1] "optimal revenue is"
```

```
opt_rev <- reg_andy1$betahat[1] * (opt_price) + reg_andy1$betahat[2] *  (opt_price) ** 2
opt_rev
```

```
## [1] 447.038
```

```
print('price elasticity is')
```

```
## [1] "price elasticity is"
```

```
reg_andy1$betahat[2] * opt_price / opt_rev
```

```
## [1] -0.1330017
```

```
#q5
reg_logandy$tstat
```

```
##               [,1]
## intercept 38.830857
## price     -7.296069
## advert     3.349973
```

```
print('CI for all of them over 99%')
```

```
## [1] "CI for all of them over 99%"
```

```
print('Yes they are significant.')
```

```
## [1] "Yes they are significant."
```

```
(1.2 - reg_logandy$betahat[3]) / sqrt(reg_logandy$VCMbetahat[3,3])
```

```
## [1] 85.11693
```

```
print('Yes it is far from 1.2')
```

```
## [1] "Yes it is far from 1.2"
```

```
#q6
#a
df2_andy <- cbind(df_andy,df_andy[,3]**2, df_andy[,4]**2)
colnames(df2_andy) <- c('sales', 'intercept',
                        'price','advert',
                        'price_sq','advert_sq')
reg_andy2 <- lm_calculation(df2_andy, c(1), c(2:6))
print('old model and new model tstats are')
```

```
## [1] "old model and new model tstats are"
```

```
reg_andy1$tstat
```

```
##                [,1]
## intercept 18.851289
## price     -7.265175
## advert     2.745150
```

```
reg_andy2$tstat
```

```
##                [,1]
## intercept  3.072285
## price     -1.988634
## advert     3.725461
## price_sq   1.723480
## advert_sq -3.285964
```

```
print('I recommend the old model is better')
```

```
## [1] "I recommend the old model is better"
```

```
#b
df3_andy <- df2_andy[,-c(5)]
reg_andy3 <- lm_calculation(df3_andy, c(1), c(2:5))
print('price beta is stat significant')
```

```
## [1] "price beta is stat significant"
```

```
reg_andy3$tstat
```

```
##                [,1]
## intercept 16.250664
## price     -7.355702
## advert     3.440929
## advert_sq -2.963339
```

```
print("the third model's coefficients are more significant the second one, and F-test is better for third one as well")
```

```
## [1] "the third model's coefficients are more significant the second one, and F-test is better for third one as well"
```

```
#q7

#d(sales) / d(adv)
optimal_adv <- -reg_andy3$betahat[3] / (2 * reg_andy3$betahat[4])
cat('optimal adv spent should be',optimal_adv,'thousand dollars')
```

```
## optimal adv spent should be 2.194978 thousand dollars
```

```
#q8
delta_sales <- -0.4 * reg_andy3$betahat[2] + 0.8 * reg_andy3$betahat[3] + 0.8**2 * reg_andy3$betahat[4]
cat('price change is',delta_sales,'thousand dollars')
```

```
## price change is 11.00549 thousand dollars
```

```
#q9

VCM_betahats3 <-  reg_andy3$VCMbetahat
se_beta2_2beta4 <- sqrt(VCM_betahats3[2,2] + 4*VCM_betahats3[3,3] - 4*2*VCM_betahats3[3,2])
t_test <- (reg_andy3$betahat[2] + 2*reg_andy3$betahat[3]) / se_beta2_2beta4
print(t_test > qt(0.02, df = 72, lower.tail = F))
```

```
## [1] TRUE
```

```
print('reject null hypothesis with 98% confidence')
```

```
## [1] "reject null hypothesis with 98% confidence"
```

Part B

Models Used in The Question

$$Y_{unrestricted,testscr} = \beta_1 + \beta_2 X_{comp/stud} + \beta_3 X_{expn/stud} + \beta_4 X_{comp/str} + \beta_5 X_{elpct} + \beta_6 X_{meanpct} + \beta_7 X_{calwpct} + \beta_8 X_{avginc} + u$$

$$Y_{lin-log,testscr} = \beta_1 + \beta_2 \log(X_{comp/stud}) + \beta_3 \log(X_{expn/stud}) + \beta_4 log(X_{comp/str}) + \beta_5 log(X_{elpct}) + \beta_6 log(X_{meanpct}) + \beta_7 log(X_{calwpct}) + \beta_8 log(X$$

$$Y_{quodratic,testscr} = \beta_1 + \beta_2 X^2_{comp/stud} + \beta_3 X^2_{expn/stud} + \beta_4 X^2_{comp/str} + \beta_5 X^2_{elpct} + \beta_6 X^2_{meanpct} + \beta_7 X^2_{calwpct} + \beta_8 X^2_{avginc} + u$$

```
path_2 <- '/Users/birdfly/Downloads/caschool.csv'
df_score <- as.matrix(read.csv(path_2))
varianble_names <- colnames(df_score)
#q1

chosen_variables <- c('testscr', 'comp_stu', 'expn_stu', 'str', 'el_pct',
                      'meal_pct','calw_pct','avginc') # these are variables I choose, they cover most of influential factors
unrest_model <- df_score[,chosen_variables]
temp_matrix <- matrix(nrow = dim(unrest_model)[1], ncol = dim(unrest_model)[2])
colnames(temp_matrix) <- chosen_variables
for (i in 1:dim(unrest_model)[1]) {
  for (j in 1:dim(unrest_model)[2]){temp_matrix[i,j]=as.numeric(unrest_model[i,j])}

} # convert str to float
unrest_model <- cbind(temp_matrix[,1], c(rep(1,length(unrest_model[,1]))),
                      temp_matrix[,2:8]) # simple multiple linear regression 1st

colnames(unrest_model) <- c('testscr', 'intercept', 'comp_stu', 'expn_stu', 'str', 'el_pct',
                            'meal_pct','calw_pct','avginc')
rest_model1 <- cbind(unrest_model[,1:2],
                     log(unrest_model[,3:9])) # lin-log regression 2nd
rest_model1[!is.finite(rest_model1)] <- 0

rest_model2 <- cbind(unrest_model[,1:2],
                     (unrest_model[,3:9])**2) # quadratic regression 3rd
rest_model2[!is.finite(rest_model2)] <- 0

print('the reason for given 3 models is that there might be some non-linear relationship to dependent variables, then quadra
tic and lin-log functions are good choices,
      we will see whether non-linear assumption is true.')
```

```
## [1] "the reason for given 3 models is that there might be some non-linear relationship to dependent variables, then quadr
atic and lin-log functions are good choices,\n      we will see whether non-linear assumption is true."
```

```
#q2
reg_score1 <- lm_calculation(unrest_model,c(1),c(2:9))
reg_score2 <- lm_calculation(rest_model1,c(1),c(2:9))
reg_score1_lm <- summary(lm(unrest_model[,1]~unrest_model[,3:9]))
reg_score2_lm <- summary(lm(rest_model1[,1]~rest_model1[,3:9]))
cat(round(reg_score1$fstat, digits = 4) == round(reg_score1_lm$fstatistic[1], digits = 4), round(reg_score2$fstat, digits =
4) == round(reg_score2_lm$fstatistic[1], digits = 4))
```

```
## TRUE TRUE
```

```
print('Results are the same')
```

```
## [1] "Results are the same"
```

```
reg_score1$betahat
```

```
##                    [,1]
## intercept 659.587074009
## comp_stu   11.890257644
## expn_stu    0.001526316
## str        -0.189910508
## el_pct     -0.198136763
## meal_pct   -0.375617949
## calw_pct   -0.077818273
## avginc      0.621672982
```

```
reg_score2$betahat
```

```
##                  [,1]
## intercept 688.6610506
## comp_stu    0.5791737
## expn_stu   -3.8497389
## str        -6.5028923
## el_pct     -3.9997197
## meal_pct   -2.0419974
## calw_pct   -4.9004349
## avginc     16.6542172
```

```
print('based on coefficients, we can say com_stu, expn_stud and avginc are positive effect, the rest of them is negative. Ho
wever, lin-log model give negative coef for expn_stud,
      there must be something wrong.')
```

```
## [1] "based on coefficients, we can say com_stu, expn_stud and avginc are positive effect, the rest of them is negative. H
owever, lin-log model give negative coef for expn_stud,\n      there must be something wrong."
```

```
print('I would go for first model/unresctricted one, because both r2 and f-test are better')
```

```
## [1] "I would go for first model/unresctricted one, because both r2 and f-test are better"
```

```
#q3
stds <- apply(unrest_model,2,sd)
means<- colMeans(unrest_model)
nor_variables <- t(apply(unrest_model,1,function(row_){(row_ - means) / stds}))
nor_variables <- nor_variables[,-2]
reg_score_normalized <- lm_calculation(nor_variables,c(1),c(2:8))
reg_score_normalized$betahat
```

```
##                 [,1]
## comp_stu  0.04053573
## expn_stu  0.05078309
## str      -0.01885626
## el_pct   -0.19015632
## meal_pct -0.53471069
## calw_pct -0.04678414
## avginc    0.23576647
```

```
print('based on beta estimation, avginc has most positive influence and meal_pct most negative')
```

```
## [1] "based on beta estimation, avginc has most positive influence and meal_pct most negative"
```

```
#q4
print('yes, R2 is high but some betas are not significant')
```

```
## [1] "yes, R2 is high but some betas are not significant"
```

```
#q5
#a

VCM_x <- matrix(0, 7, 7)
for (i in 1:7) {
    for (j in 1:7) {
        VCM_x[i,j] <- cov(nor_variables[,i+1] , nor_variables[,j+1])
    }
}
round(cov(nor_variables[,-1]), digits = 6) == round(VCM_x, digits = 6 )
```

```
##          comp_stu expn_stu  str el_pct meal_pct calw_pct avginc
## comp_stu     TRUE     TRUE TRUE   TRUE     TRUE     TRUE   TRUE
## expn_stu     TRUE     TRUE TRUE   TRUE     TRUE     TRUE   TRUE
## str          TRUE     TRUE TRUE   TRUE     TRUE     TRUE   TRUE
## el_pct       TRUE     TRUE TRUE   TRUE     TRUE     TRUE   TRUE
## meal_pct     TRUE     TRUE TRUE   TRUE     TRUE     TRUE   TRUE
## calw_pct     TRUE     TRUE TRUE   TRUE     TRUE     TRUE   TRUE
## avginc       TRUE     TRUE TRUE   TRUE     TRUE     TRUE   TRUE
```

```
#b
aux_matrix <- nor_variables[,-1]
aux_reg_x2 <- lm_calculation(aux_matrix, c(2), c(1,3:7))
aux_reg_x3 <- lm_calculation(aux_matrix, c(3), c(1:2,4:7))

#c
library(corpcor)
partial_cor <- cor2pcor(VCM_x)
simple_cor <- cov2cor(VCM_x)
colnames(partial_cor) <- colnames(simple_cor)
rownames(partial_cor) <- rownames(simple_cor)
print('partial and simple correlations between x2 and x3 are
      close and both are negative.') # x2 is student/teacher ratio and x3 is expense/student which makes sense that less tea
cher less expense
```

```
## [1] "partial and simple correlations between x2 and x3 are\n      close and both are negative."
```

```
#6
library(car)
```

```
## Loading required package: carData
```

```
vifs <- vif(lm(nor_variables[,1]~nor_variables[,2]+nor_variables[,3]+nor_variables[,4]+nor_variables[,5]+nor_variables[,6]+n
or_variables[,7]+nor_variables[,8]))
round(1 / (1 - aux_reg_x2$r2), digits = 4) == round(vifs[2], digits = 4)
```

```
## nor_variables[, 3]
##               TRUE
```

```
round(1 / (1 - aux_reg_x3$r2), digits = 4) == round(vifs[3], digits = 4)
```

```
## nor_variables[, 4]
##               TRUE
```

```
print('yes package gives same values with 5(b)')
```

```
## [1] "yes package gives same values with 5(b)"
```

```
diag(reg_score_normalized$VCMbetahat)
```

```
##     comp_stu     expn_stu          str       el_pct     meal_pct
## 0.0005503836 0.0008758637 0.0007887404 0.0010124060 0.0025980647
##     calw_pct       avginc
## 0.0011774967 0.0011013529
```

```
print('yes mean_pct has the highest vif and variance')
```

```
## [1] "yes mean_pct has the highest vif and variance"
```

```
#7
print('based on partial correlation & simple correlation, first we remove avginc
      as it has higher vif and some correlations with other variables. Then,
      remove expn_stu, then remove meal_pct whose vif is over 5. We keep
      el_pct and calw_pct')
```

```
## [1] "based on partial correlation & simple correlation, first we remove avginc\n      as it has higher vif and some corre
lations with other variables. Then,\n      remove expn_stu, then remove meal_pct whose vif is over 5. We keep \n      el_pct
and calw_pct"
```

```
final_model <- nor_variables[,c(1,2,4,5,7)]
summary(lm(final_model[,1]~0+final_model[,2:5]))
```

```
##
## Call:
## lm(formula = final_model[, 1] ~ 0 + final_model[, 2:5])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5990 -0.3997  0.0412  0.3477  1.8304
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## final_model[, 2:5]comp_stu  0.04793    0.03215   1.491 0.136768
## final_model[, 2:5]str      -0.11676    0.03162  -3.693 0.000251 ***
## final_model[, 2:5]el_pct   -0.45983    0.03251 -14.144  < 2e-16 ***
## final_model[, 2:5]calw_pct -0.47049    0.03162 -14.878  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.61 on 416 degrees of freedom
## Multiple R-squared:  0.6305, Adjusted R-squared:  0.627
## F-statistic: 177.5 on 4 and 416 DF,  p-value: < 2.2e-16
```

```
vif(lm(final_model[,1]~final_model[,2] + final_model[,3] + final_model[,4]
        +final_model[,5]))
```

```
## final_model[, 2] final_model[, 3] final_model[, 4] final_model[, 5]
##         1.163794         1.125607         1.189998         1.125966
```

```
print('this is much better and without lossing some relavant predictors')
```

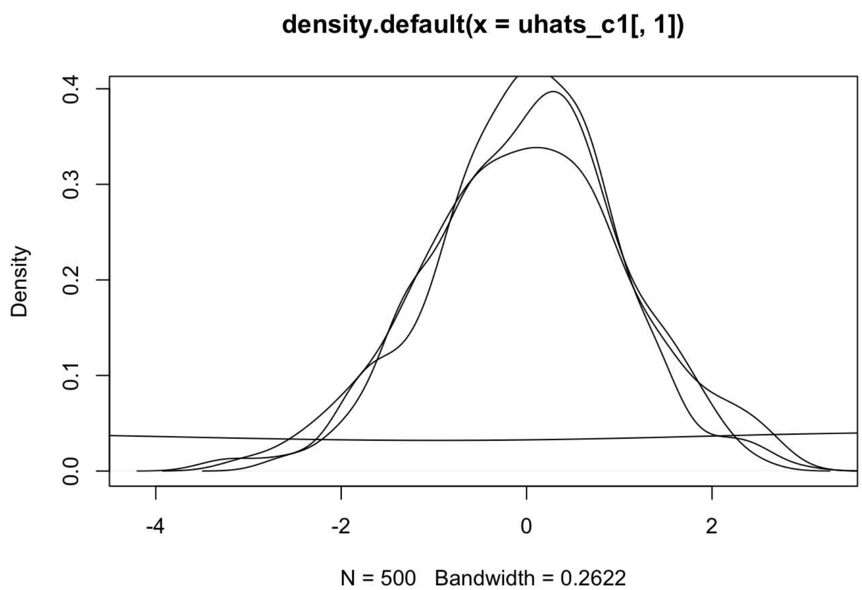
```
## [1] "this is much better and without lossing some relavant predictors"
```

Part C

```r
#q1
x1 <- rep(1, 1000)
x2 <- sample(c(0:100),1000, replace = T)
x3 <- sample(0:1, 1000, replace = T)
x4 <- floor(runif(1000, min=1, max=50))
x5 <-rnorm(1000, 5.2, 1.25)
beta <- c(12, -0.7, 34, -0.17, 5.4)
uhats_c1 <- c()
for (rho in c(-0.1,0.1,0.5,1) ) {
  u <- rnorm(1000, 0, 1)
  for(i in 1:999){
      u[i+1] <- u[i] * rho + rnorm(1,0,1)
  }

  xs <- cbind(x1,x2,x3,x4,x5)
  y <- xs%*%beta + u
  model <- cbind(y,xs,u)

  sample_1 <- model[sample(1:1000,size = 500,replace = T ),]
  betahat_c1 <- solve(t(sample_1[,2:6]) %*% sample_1[,2:6]) %*% t(sample_1[,2:6]) %*% sample_1[,1]
  uhats_c1 <- cbind(uhats_c1,sample_1[,1] - sample_1[,2:6] %*% betahat_c1)
}
plot(density(uhats_c1[,1]))
lines(density(uhats_c1[,2]))
lines(density(uhats_c1[,3]))
lines(density(uhats_c1[,4])) # this is the flattened line and represents rho = 1
```




```r
#chose rho = 0.1 & 0.5 for question b
mean(uhats_c1[,2])
```

```
## [1] -1.965325e-14
```

```r
var(uhats_c1[,2])
```

```
## [1] 0.9312752
```

```r
cor(uhats_c1[2:500,2], uhats_c1[1:499,2])
```

```
## [1] -0.006161314
```

```r
print('I dont think correlation here is close to rho')
```

```
## [1] "I dont think correlation here is close to rho"
```

```r
# rho = 0.5
mean(uhats_c1[,3])
```

```
## [1] -3.144716e-14
```

```r
var(uhats_c1[,3])
```

```
## [1] 1.28978
```

```r
print('yes, larger rho gives larger mean and variance. And larger sample size gives smaller SE as well')
```

```
## [1] "yes, larger rho gives larger mean and variance. And larger sample size gives smaller SE as well"
```

```r
#MC sim.
#1.
n <- 500
sample <- 5000
fixed_xs <- model[sample(1:1000,n, replace = T),2:6]
betahats_collector <- c()
uhats_collector <- c()
variance_collector <- c()
for (i in 1:sample) {
    u <- rnorm(500, 0, 1)
    for(j in 1:(n-1)){
      u[j+1] <- u[j] * 0.3  + rnorm(1,0,1)
    }
    y <- fixed_xs %*% beta + u
    reg_mc <- lm_calculation(cbind(y,fixed_xs),c(1),c(2:6))
    betahats_mc <- reg_mc$betahat
    yhats_mc <- reg_mc$yhat
    uhats_mc <- reg_mc$uhat
    variance_mc <- reg_mc$var
    betahats_collector <- rbind(betahats_collector, t(betahats_mc))
    uhats_collector <- rbind(uhats_collector, t(uhats_mc))
    variance_collector <- rbind(variance_collector, variance_mc)
}

betahats_means <- colMeans(betahats_collector)
betahats_means - beta
```

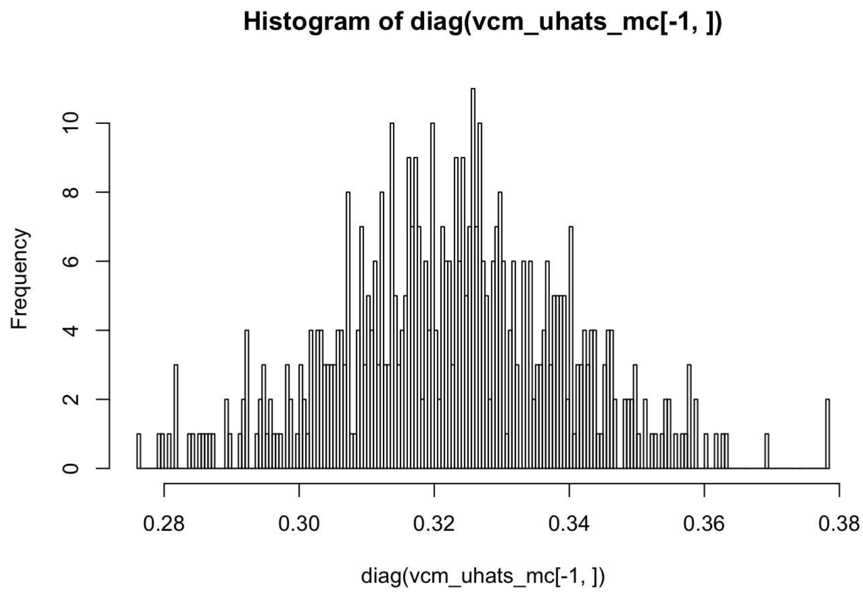
```
##            x1            x2            x3            x4            x5
##  1.743878e-03 -2.146981e-05  1.647838e-03 -1.166795e-05 -2.845572e-04
```

```r
print('yes they are very close')
```

```
## [1] "yes they are very close"
```

```r
#2.
vcm_uhats_mc <- matrix(0,n,n)
for (i in 1:n) {
    for (j in 1:n) {
      vcm_uhats_mc[i,j] <- cov(uhats_collector[,i], uhats_collector[,j])
    }
}

hist(diag(vcm_uhats_mc[-1,]), breaks = 150)
```




```
print('no it does not look like so, E(UtUt-1) seems like close to rho')
```

```
## [1] "no it does not look like so, E(UtUt-1) seems like close to rho"
```

```
#3
vcm_beta_AR1 <- solve(t(fixed_xs) %*% fixed_xs) %*% t(fixed_xs) %*% vcm_uhats_mc %*% fixed_xs %*% solve(t(fixed_xs) %*% fixe
d_xs)
vcm_beta_AR1
```

```
##               x1            x2            x3            x4            x5
## x1 -1.572461e-19  2.153047e-21 -1.263582e-19  7.626962e-21  4.216177e-21
## x2 -1.802987e-21  2.563237e-23 -1.217663e-21 -2.165015e-23  6.424702e-22
## x3 -4.490309e-20 -7.020984e-22 -1.031050e-19  1.840253e-21  2.533280e-20
## x4 -2.001882e-22  9.581820e-23  4.561655e-21 -1.792986e-22  2.744083e-22
## x5  3.133198e-19 -5.353232e-22  2.702554e-20 -2.124964e-21 -4.244377e-20
```

```
#4
sigma_sqr <- (t(uhats_collector[sample,]) %*% uhats_collector[sample,]) / (500 - 5)
VCM_betahat_OLS <- sum(sigma_sqr) * t(fixed_xs) %*% fixed_xs
print('yes exactly, their values are very large')
```

```
## [1] "yes exactly, their values are very large"
```