# Confidence intervals for linear unbiased estimators under constrained dependence

*Peter M. Aronow, Forrest W. Crawford, and Jose R. Zubizarreta*

*March 21, 2017*

## Code for Section 5.1

```
########################
# Read packages

library("slam")
library("Rglpk")
```

```
## Using the GLPK callable library version 4.55
```

```
library("gurobi")
library("xtable")
library("sandwich")
library("lmtest")
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library("depinf")

########################
# Load data

load("Russia_sathcap.Rsave")

########################
# Initialize basic variables

Y = cycle1$hiv
X = cycle1[,c("hcv","age","sex")]

n = length(Y)

# GR is the known elements of the adjacency matrix
dr = apply(GR, 1, sum)
cycle1$degree_corrected = pmax(cycle1$degree, dr)
d = cycle1$degree_corrected

########################
# Add an intercept
```

```r
X = as.matrix(cbind(1, X))

########################
# Calculate the Theta matrix

Theta = solve(t(X)%*%X, t(X))

betahat = Theta %*% Y

########################
# Get the lm confidence intervals

linfit = lm(Y ~ hcv + age + sex, data = as.data.frame(X))

print(summary(linfit))
```

```
##
## Call:
## lm(formula = Y ~ hcv + age + sex, data = as.data.frame(X))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5911 -0.4932 -0.1106  0.4936  0.9010
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.139048   0.091657   1.517    0.130
## hcv          0.380922   0.045138   8.439 3.22e-16 ***
## age          0.001655   0.002828   0.585    0.559
## sex         -0.069802   0.045779  -1.525    0.128
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4621 on 518 degrees of freedom
## Multiple R-squared:  0.1239, Adjusted R-squared:  0.1188
## F-statistic: 24.42 on 3 and 518 DF,  p-value: 8.599e-15
```

```r
########################
# Get the naive and robust confidence intervals

# Naive
lmfit = lm(Y ~ X)
summary(lmfit)
```

```
##
## Call:
## lm(formula = Y ~ X)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5911 -0.4932 -0.1106  0.4936  0.9010
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  0.139048   0.091657   1.517    0.130
## X1                 NA         NA      NA       NA
## Xhcv         0.380922   0.045138   8.439 3.22e-16 ***
## Xage         0.001655   0.002828   0.585    0.559
## Xsex        -0.069802   0.045779  -1.525    0.128
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4621 on 518 degrees of freedom
## Multiple R-squared:  0.1239, Adjusted R-squared:  0.1188
## F-statistic: 24.42 on 3 and 518 DF,  p-value: 8.599e-15
```

```
# Robust (Stata default)
coeftest(lmfit, vcov = vcovHC(lmfit, "HC1"))[, 2]
```

```
## (Intercept)        Xhcv        Xage        Xsex
## 0.082084807 0.038993441 0.002809911 0.044394264
```

```
########################
# Build table

# Regression coefficients
col1 = betahat
# Naive standard errors
col2 = coef(summary(lmfit))[, 2]
# Robust standard errors
col3 = coeftest(lmfit, vcov = vcovHC(lmfit, "HC1"))[, 2]

dr = apply(GR, 1, sum)
cycle1$degree_corrected = pmax(cycle1$degree, dr)
d = cycle1$degree_corrected

# V1, no known dependencies (no GR)
a1 = sqrt(depvar(Y, Theta[1, ]*n, d, NULL, case = "heteroskedastic", solver = "gurobi", approximate = 0)
a2 = sqrt(depvar(Y, Theta[2, ]*n, d, NULL, case = "heteroskedastic", solver = "gurobi", approximate = 0)
a3 = sqrt(depvar(Y, Theta[3, ]*n, d, NULL, case = "heteroskedastic", solver = "gurobi", approximate = 0)
a4 = sqrt(depvar(Y, Theta[4, ]*n, d, NULL, case = "heteroskedastic", solver = "gurobi", approximate = 0)
col4 = c(a1, a2, a3, a4)

# V1, some known dependencies
b1 = sqrt(depvar(Y, Theta[1, ]*n, d, GR, case = "heteroskedastic", solver = "gurobi", approximate = 0)$V
b2 = sqrt(depvar(Y, Theta[2, ]*n, d, GR, case = "heteroskedastic", solver = "gurobi", approximate = 0)$V
b3 = sqrt(depvar(Y, Theta[3, ]*n, d, GR, case = "heteroskedastic", solver = "gurobi", approximate = 0)$V
b4 = sqrt(depvar(Y, Theta[4, ]*n, d, GR, case = "heteroskedastic", solver = "gurobi", approximate = 0)$V
col5 = c(b1, b2, b3, b4)

# V2, some known dependencies
c1 = sqrt(depvar(Y, Theta[1, ]*n, d, GR, case = "homoskedastic", solver = "gurobi", approximate = 0)$V_
c2 = sqrt(depvar(Y, Theta[2, ]*n, d, GR, case = "homoskedastic", solver = "gurobi", approximate = 0)$V_
c3 = sqrt(depvar(Y, Theta[3, ]*n, d, GR, case = "homoskedastic", solver = "gurobi", approximate = 0)$V_
c4 = sqrt(depvar(Y, Theta[4, ]*n, d, GR, case = "homoskedastic", solver = "gurobi", approximate = 0)$V_
col6 = c(c1, c2, c3, c4)

# V2'
d1 = sqrt(depvar(Y, Theta[1, ]*n, d, NULL, case = "homoskedastic")$V_hat)
```

```r
d2 = sqrt(depvar(Y, Theta[2, ]*n, d, NULL, case = "homoskedastic")$V_hat)
d3 = sqrt(depvar(Y, Theta[3, ]*n, d, NULL, case = "homoskedastic")$V_hat)
d4 = sqrt(depvar(Y, Theta[4, ]*n, d, NULL, case = "homoskedastic")$V_hat)
col7 = c(d1, d2, d3, d4)

tab = cbind(col1, col2, col3, col4, col5, col6, col7)
tab
```

```
##                          col2        col3       col4       col5       col6
## 1    0.139047787 0.091657373 0.082084807 0.36621941 0.32578676 0.43536454
## hcv  0.380921703 0.045137749 0.038993441 0.17351122 0.15951663 0.19034247
## age  0.001654649 0.002827616 0.002809911 0.01210594 0.01064702 0.01461786
## sex -0.069801807 0.045779481 0.044394264 0.23132670 0.20556057 0.26939276
##           col7
## 1   0.43915251
## hcv 0.19199857
## age 0.01474504
## sex 0.27173666
```

```r
########################
# Sensitivity analysis: solve for V1, V2 and V2' for d = cycle1$degree_corrected + k

for (j in 1:4) {

    #for (k in 0:(n-1)) {
    for (k in 0:2) {

        cat("Coefficient ", j-1, ", iteration ", k, "\n", sep="")

        # V1
        out1 = depvar(y = Y, theta = Theta[j, ]*n, d = pmin(d+k, n-1), GR = GR, case = "heteroskedastic

        # V2
        out2 = depvar(y = Y, theta = Theta[j, ]*n, d = pmin(d+k, n-1), GR = GR, case = "homoskedastic",

        # V2'
        x = Y*Theta[j, ]*n
        out3 = var(x)/n * (1 + sum(pmin(d+k, n-1))/n)

        aux1 = c(sqrt(out1$V_hat), betahat[j]-1.96*sqrt(out1$V_hat), betahat[j]+1.96*sqrt(out1$V_hat))
        aux2 = c(sqrt(out2$V_hat), betahat[j]-1.96*sqrt(out2$V_hat), betahat[j]+1.96*sqrt(out2$V_hat))
        aux3 = c(sqrt(out3), betahat[j]-1.96*sqrt(out3), betahat[j]+1.96*sqrt(out3))

        temp = rbind(aux1, aux2, aux3)
        rownames(temp) = c(paste("V1, k =", k, sep = " "), paste("V2, k =", k, sep = " "), paste("V2', k
        colnames(temp) = c("s.e", "(", ")")

        if (k == 0) {
            tab = temp
        }
        else {
            tab = rbind(tab, temp)
        }
    }
```

```r
    assign(paste('tab_beta', j-1, sep=''), tab)
}
```

```
## Coefficient 0, iteration 0
## Coefficient 0, iteration 1
## Coefficient 0, iteration 2
## Coefficient 1, iteration 0
## Coefficient 1, iteration 1
## Coefficient 1, iteration 2
## Coefficient 2, iteration 0
## Coefficient 2, iteration 1
## Coefficient 2, iteration 2
## Coefficient 3, iteration 0
## Coefficient 3, iteration 1
## Coefficient 3, iteration 2
```

```r
tab_approx = tab

sink("tab_approx.txt")

cat("Intercept", "\n")
```

```
## Intercept
```

```r
tab_beta0
```

```
##                    s.e          (          )
## V1, k = 0  0.3257868 -0.4994943 0.7775898
## V2, k = 0  0.4353645 -0.7142667 0.9923623
## V2', k = 0 0.4391767 -0.7217385 0.9998341
## V1, k = 1  0.3506231 -0.5481736 0.8262691
## V2, k = 1  0.4515966 -0.7460816 1.0241772
## V2', k = 1 0.4515966 -0.7460816 1.0241772
## V1, k = 2  0.3707692 -0.5876598 0.8657554
## V2, k = 2  0.4636840 -0.7697729 1.0478685
## V2', k = 2 0.4636840 -0.7697729 1.0478685
```

```r
cat("\n", "Beta 1", "\n")
```

```
##
##  Beta 1
```

```r
tab_beta1
```

```
##                    s.e            (          )
## V1, k = 0  0.1595183  0.068265845 0.6935776
## V2, k = 0  0.1903425  0.007850468 0.7539929
## V2', k = 0 0.1920091  0.004583783 0.7572596
## V1, k = 1  0.1675352  0.052552725 0.7092907
## V2, k = 1  0.1974392 -0.006059080 0.7679025
## V2', k = 1 0.1974392 -0.006059080 0.7679025
## V1, k = 2  0.1749642  0.037991807 0.7238516
## V2, k = 2  0.2027238 -0.016416972 0.7782604
## V2', k = 2 0.2027238 -0.016416972 0.7782604
```

```r
cat("\n", "Beta 2", "\n")
```

```
##
```

```
##  Beta 2
```

```r
tab_beta2
```

```
##                    s.e           (           )
## V1, k = 0  0.01064702 -0.01921351 0.02252281
## V2, k = 0  0.01461786 -0.02699635 0.03030565
## V2', k = 0 0.01474585 -0.02724722 0.03055652
## V1, k = 1  0.01150136 -0.02088801 0.02419730
## V2, k = 1  0.01516287 -0.02806457 0.03137387
## V2', k = 1 0.01516287 -0.02806457 0.03137387
## V1, k = 2  0.01219763 -0.02225271 0.02556201
## V2, k = 2  0.01556871 -0.02886003 0.03216933
## V2', k = 2 0.01556871 -0.02886003 0.03216933
```

```r
cat("\n", "Beta 3", "\n")
```

```
## 
##  Beta 3
```

```r
tab_beta3
```

```
##                   s.e          (          )
## V1, k = 0  0.2056665 -0.4729082 0.3333045
## V2, k = 0  0.2693928 -0.5978116 0.4582080
## V2', k = 0 0.2717516 -0.6024350 0.4628314
## V1, k = 1  0.2205891 -0.5021564 0.3625528
## V2, k = 1  0.2794368 -0.6174979 0.4778943
## V2', k = 1 0.2794368 -0.6174979 0.4778943
## V1, k = 2  0.2340781 -0.5285950 0.3889914
## V2, k = 2  0.2869162 -0.6321575 0.4925538
## V2', k = 2 0.2869162 -0.6321575 0.4925538
```

```r
save(tab_beta0, tab_beta1, tab_beta2, tab_beta3, file = "tab_approx.RData")
```