Advanced Quantitative Methods
Heteroscedasticity

Johan A. Elkink
jos.elkink@ucd.ie

February 21, 2010

If you encounter any mistakes in this text, please inform the author.

Contents

1 White's HCCM in R

The following code implements the matrix algebra in R to calculate the various variations on White's **heteroscedasticity consistent covariance matrices** (HCCMs), as outlined in Long and Ervin (2000):

```
## Use sample data set from Faraway on French Presidential elections
library(faraway)

data(fpe)
m <- lm(A2 ~ A + B + C + D + E, data=fpe)

mf <- model.frame(m)
y <- model.response(mf)
X <- model.matrix(m)

n <- dim(X)[1]
k <- dim(X)[2]

## Estimate OLS and various covariance matrices
XXi <- solve(t(X) %*% X)
bhat <- XXi %*% t(X) %*% y
H <- X %*% XXi %*% t(X)
h <- diag(diag(H))  ## remove all off-diagonal elements
e <- as.numeric(y - X %*% bhat)
s2 <- t(e) %*% e / (n-k)
vcov <- s2 %x% XXi
```

```
omega0 <- diag(e^2)
hc0 <- XXi %*% t(X) %*% omega0 %*% X %*% XXi
hc1 <- n/(n-k) * hc0
omega2 <- diag(e^2) / (1 - h)
hc2 <- XXi %*% t(X) %*% omega2 %*% X %*% XXi
omega3 <- diag(e^2) / (1 - h)^2
hc3 <- XXi %*% t(X) %*% omega3 %*% X %*% XXi

tbl <- cbind(bhat, sqrt(diag(vcov)),
                   sqrt(diag(hc0)), sqrt(diag(hc1)),
                   sqrt(diag(hc2)), sqrt(diag(hc3)))
colnames(tbl) <- c("B-hat", "SE (OLS)", "SE (HC0)", "SE (HC1)",
                            "SE (HC2)", "SE (HC3)")
tbl

## Check using built-in methods
sqrt(diag(hccm(m, type="hc0")))
sqrt(diag(hccm(m, type="hc1")))
sqrt(diag(hccm(m, type="hc2")))
sqrt(diag(hccm(m, type="hc3")))
```

2 Harrison-McCabe statistics in R

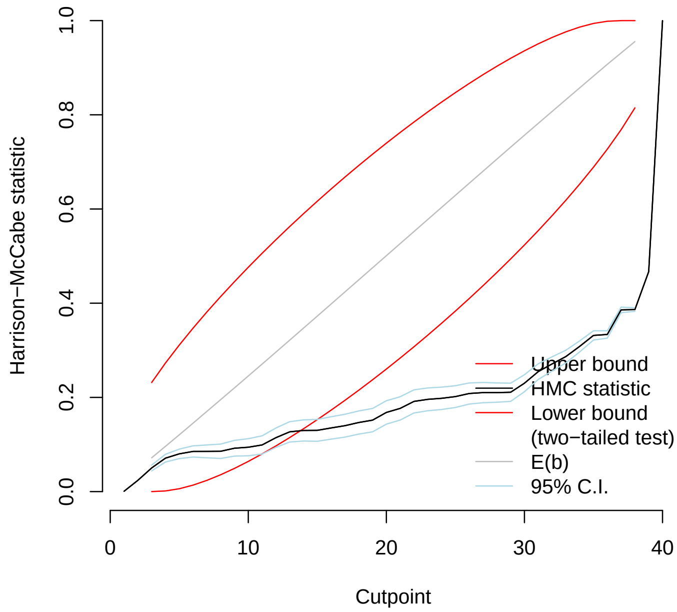
The following code provides a graphical visualisation of the test for heteroscedasticity as provided by

```
## Creating fake data set
n <- 40

x <- rnorm(n,0,1)
e <- exp(x) + rnorm(n)
y <- 1 + 3 * x + e

y <- y[order(y)]
x <- x[order(x)]

## Estimate model
summary(m <- lm(y~x))
k <- dim(model.matrix(m))[2]

## Harrison-McCabe test, manual (plot)
b <- cumsum(residuals(m)^2) / sum(residuals(m)^2)
r <- (k+1):(n-k)
bu <- qbeta(.975, r/2, (n-r-k)/2)
bl <- qbeta(.025, (r-k)/2, (n-r)/2)

plot(b, type="l", bty="n", ylim=c(0,1),
     ylab="Harrison-McCabe statistic",
     xlab="Cutpoint")
```

```
lines(bu ~ r, col="red")
lines(bl ~ r, col="red")
legend("bottomright", col=c("red","black","red","white","gray","lightblue"),
       lty="solid", legend=c("Upper bound", "HMC statistic", "Lower bound",
                             "(two-tailed test)","E(b)","95% C.I."), bty="n")

HMC.exact <- function(m, r) {

  X <- model.matrix(m)
  n <- dim(X)[1]
  k <- dim(X)[2]

  M <- diag(1, n) - X %*% solve(t(X) %*% X) %*% t(X)
  A <- diag(c(rep(1, r),
              rep(0, n-r)))
  lambda <- eigen(M %*% A, only.values=T)$values[(r-k/2):(r+k/2-1)]
  Eb <- (r-k+sum(lambda))/(n-k)
  Vb <- 2 * ((r-k)*(1-Eb)^2+(n-r-k)*Eb^2+sum((lambda-Eb)^2)) / ((n-k) * (n-k+2))

  list(Eb=Eb, Vb=Vb)
}

eb <- vb <- NULL
for (i in r) {
hmcx <- HMC.exact(m, i)
eb <- c(eb, hmcx$Eb)
vb <- c(vb, hmcx$Vb)
}

zscore <- qt(.975, n-k)
lines(eb ~ r, col="gray")
lines(b[r] - zscore * vb ~ r, col="lightblue")
lines(b[r] + zscore * vb ~ r, col="lightblue")
lines(b)
```

References

Long, J. Scott and Laurie H. Ervin. 2000. “Using heteroscedasticity consistent standard errors in the linear regression model.” *The American Statistician* 54(3):217–224.



Harrison−McCabe statistic
0.0
0.2
0.4
0.6
0.8
1.0
0
10
20
30
40
Cutpoint
Upper bound
HMC statistic
Lower bound
(two−tailed test)
E(b)
95% C.I.