ECON 582 **Shabab Ahmed**

Due Date: Friday, April 9

Homework 1

PROBLEM 9.26:

In a paper in 1963, Marc Nerlove analyzed a cost function for 145 American electric companies. Nerlove was interested in estimating a *cost function*: $C = f(Q, PL, PF, PK)$.
a) First, estimate an unrestricted Cobb-Douglas specification

$$\log C = \beta_1 + \beta_2 \log Q + \beta_3 \log PL + \beta_4 \log PK + \beta_5 \log PF + e \qquad (9.23)$$

Report parameter estimates and standard errors.

b) What is the economic meaning of the restriction $H_0 : \beta_3 + \beta_4 + \beta_5 = 1$?

c) Estimate (9.23) by constrained least squares imposing $\beta_3 + \beta_4 + \beta_5 = 1$. Report your parameter estimates and standard errors.

d) Estimate (9.23) by efficient minimum distance imposing $\beta_3 + \beta_4 + \beta_5 = 1$. Report your parameter estimates and standard errors.

e) Test $H_0 : \beta_3 + \beta 4 + \beta 5 = 1$ using Wald statistic.

f) Test $H_0 : \beta_3 + \beta 4 + \beta 5 = 1$ using a minimum distance statistic.

SOLUTION:

---

a) Code:

```
data <- read.table("/Users/student/Desktop/Spring21/582/HW1/Nerlove1963.txt",
header = TRUE)
log_C <- matrix(log(data$Cost), ncol=1)
log_Q <- matrix(log(data$output), ncol=1)
log_PL <- matrix(log(data$Plabor), ncol=1)
log_PK <- matrix(log(data$Pcapital), ncol=1)
log_PF <- matrix(log(data$Pfuel), ncol=1)
x <- as.matrix(cbind(matrix(1,nrow(log_C),1),log_Q, log_PL, log_PK, log_PF))
y<- log_C
n<- nrow(x)
k<- ncol(x)
```

```
# a) Unrestricted regression

invx <- solve(t(x)%*%x)
b_ols <- solve(t(x)%*%x)%*%(t(x)%*%y)
e_ols <- rep((y-x%*%b_ols), times=k)
xe_ols <-x*e_ols
V_ols <- (n/(n-k))*invx%*%(t(xe_ols)%*%xe_ols)%*%invx
se_ols <- sqrt(diag(V_ols))
print(b_ols)
print(se_ols)

#--------------------------------------------------------------
```

	Estimate	**Standard Error**
$\beta 1$	-3.5265028	1.71860065
$\beta 2$	0.7203941	0.03259753
$\beta 3$	0.4363412	0.24563580
$\beta 4$	-0.2198884	0.32381213
$\beta 5$	0.4265170	0.07548271

b) The restriction $H_0 : \beta_3 + \beta_4 + \beta_5 = 1$ is testing whether we have a constant-return-to-scale production function with respect to the input (labor, capital and fuel). When we impose the restriction we are imposing that we have a constant-return-to-scale production function because we have a Cobb douglas specification. This means that the cost function exhibits constant-return-to-scale with respect to the input prices. If all the input prices increase by a certain amount x, then the cost will also increase by the same amount x.

c) Code:

```
#c) Constrained regression

R <- c(0,0,1,1,1)
c <- 1
iR <- invx%*%R%*%solve(t(R)%*%invx%*%R)
b_cls <- b_ols -iR%*%t(R)%*%b_ols +iR%*%c
e_cls <- rep((y-x%*%b_cls),times=k)
xe_cls <- x*e_cls
V_tilde <- (n/(n-k+1))*invx%*%(t(xe_cls)%*%xe_cls)%*%invx
V_cls <- V_tilde-iR%*%t(R)%*%V_tilde-V_tilde%*%t(iR%*%t(R))+
iR%*%t(R)%*%V_tilde%*%t(iR%*%t(R))
se_cls <- sqrt(diag(V_cls))
```

```
print(b_cls)
print(se_cls)

#-------------------------------------------------------------
```

	Estimate	Standard Error
$\beta 1$	-4.690789123	0.81485793
$\beta 2$	0.720687524	0.03245926
$\beta 3$	0.592909608	0.16906852
$\beta 4$	-0.007381064	0.15579133
$\beta 5$	0.414471455	0.07286728

d) Code:

```
# d) Efficient minimum distance
iV <- solve(t(R)%*%V_ols%*%R)
V<-V_ols%*%R%*%iV
b_emd <- b_ols-V%*%t(R)%*%b_ols+V%*%c
e_emd<- rep((y-x%*%b_emd), times=k)
xe_emd <-x*e_emd
V2 <- (n/(n-k+1))*invx%*%(t(xe_emd)%*%xe_emd)%*%invx
V_emd <-V2 - V2%*%R%*%solve(t(R)%*%V2%*%R)%*%t(R)%*%V2
se_emd<- sqrt(diag(V_emd))
print(b_emd)
print(se_emd)

#-------------------------------------------------------------
```

	Estimate	Standard Error
$\beta 1$	-4.744646018	0.81541660
$\beta 2$	0.720190849	0.03230573
$\beta 3$	0.580519645	0.16946463
$\beta 4$	0.009219041	0.15524763
$\beta 5$	0.410261314	0.07244074

e) Let $\theta = \beta_3 + \beta_4 + \beta_5$. Then, $\theta = r(\beta) = R'\beta$ is a linear function of β with $R' = [0\,0\,1\,1\,1]$. In this case, $\theta_0 = 1$. Then the Wald statistic is:

$$W = (R'\hat{\beta} - 1)'(R'\hat{V}_{\hat{\beta}}R)^{-1}(R'\hat{\beta} - 1)$$

Since, $q = 1$ the Wald statistic follows an asymptotic distribution of chi-square with 1 degree of freedom. We will conduct a test of asymptotic size $\alpha = 0.05$. Then we find a critical value c such that $0.05 = 1 - G_1(c)$ and reject H_0 if $W > c$. The following code is used for this problem:

```
# e) Wald statistic
q <- length(c)
V_r <- solve(t(R)%*%V_ols%*%R)
W <- t(t(R)%*%b_ols-c)%*%V_r%*%t(t(R)%*%b_ols-c)
alpha <- 0.05
C <- qchisq(1-alpha, q)
print(W)
print(C)

if (W>C){
print("Reject H0")
} else {
print("Accept H0")
}

#--------------------------------------------------------------
```

We have that $W = 0.6454737$ and the critical value $c = 3.841459$. Since, $W < c$ we accept the H_0 in this Wald test of asymptotic size 0.05.

f) The efficient minimum distance statistic is given by:

$$J = n(\hat{\beta} - \tilde{\beta}_{emd})'\hat{V}_{\beta}^{-1}(\hat{\beta} - \tilde{\beta}_{emd})$$

Code:

```
# f) Minimun distance statistic
q <- length(c)
J <- t((b_ols-b_emd))%*%solve(V_ols)%*%(b_ols-b_emd)
alpha <- 0.05
C <- qchisq(1-alpha, q)
print(J)
print(C)
if (J>C){
    print("Reject H0")
} else {
    print("Accept H0")
```

```
}
#-------------------------------------------------------------
```

We get that $J = 0.6454737$ and the critical value =3.841459. Therefore, we accept H_0 since $J < c$ in this minimum distance test. This makes sense as we know that in the class of linear hypotheses, the efficient minimum distance statistic is simply the Wald statistic. Since, we have a linear hypothesis we have found that $J = W$ and for a given asymptotic size α our conclusions from the two tests are going to be the same.

PROBLEM 10.28:

In Exercise 9.26 you estimated a cost function for 15 electric companies and tested the restriction $\theta = \beta_3 + \beta_4 + \beta_5 = 1$.

a) Estimate the regression by unrestricted least squares and report standard errors calculated by asymptotic, jackknife and the bootstrap.

b) Estimate $\theta = \beta_3 + \beta_4 + \beta_5$ and report standard errors calculated by asymptotic, jackknife and the bootstrap.

c) Report confidence intervals for θ using the percentile and BC_a methods.

SOLUTION:

---

a) Code:

```
set.seed(1)
# Creating a function to calculate Beta hat

estimate.beta <- function(z,y){
    n<- nrow(z)
    k<- ncol(z)


    invz <- solve(t(z)%*%z)
    b_ols <- solve(t(z)%*%z)%*%(t(z)%*%y)
     e_ols <- rep((y-z%*%b_ols), times=k)
    ze_ols <-z*e_ols
    V_ols <- (n/(n-k))*invz%*%(t(ze_ols)%*%ze_ols)%*%invz
    se_ols <- sqrt(diag(V_ols))
    return(b_ols)
    return(se_ols)
}

# Asymptotic

beta.hat <- estimate.beta(z=x, y=log_C)
print(beta.hat)
print(se_ols)
```

```
# Jackknife

beta.hat.jack <- matrix(data=0, nrow = k, ncol = n)

for (i in 1:n) {
    df.loo.i <- x[-i,]
    p<- log_C[-i,]
    beta.hat.jack[,i] <- estimate.beta(z=df.loo.i,y=p)
}

beta.bar.jack <- rowMeans(beta.hat.jack)
diff.jack <- (beta.hat.jack-beta.bar.jack)
var.jack <- ((n-1)/n)*(diff.jack%*%t(diff.jack))
se.jack <- sqrt(diag(var.jack))
se.jack

# Bootstrap

B<- 10000

beta.hat.boot <- matrix(0, nrow=k, ncol =B)

for (b in 1:B){
# Construct a b-th bootstrap sample
    idx.b <- sample(n, replace = TRUE)
    df.b <- x[idx.b,]
    y.b <- log_C[idx.b, ]
    beta.hat.boot[,b] <- estimate.beta(z=df.b, y=y.b)
}

beta.bar.boot <- rowMeans(beta.hat.boot)
diff.boot <- (beta.hat.boot-beta.bar.boot)
var.boot <- (1/B)*(diff.boot %*% t(diff.boot))
se.boot <- sqrt(diag(var.boot))
se.boot

#--------------------------------------------------------------
```

	Estimate	Asymptotic SE	Jackknife SE	Bootstrap SE
β_1	-3.5265028	1.71860065	1.78802845	1.7513769
β_2	0.7203941	0.03259753	0.03393373	0.0328449
β_3	0.4363412	0.24563580	0.25316596	0.2501165
β_4	-0.2198884	0.32381213	0.33634244	0.3303347
β_5	0.4265170	0.07548271	0.07775186	0.0777011

b) Code:

```
# b)

theta.hat <- b_ols[3]+b_ols[4]+b_ols[5]
theta.hat

# Asymptotic (Using the Delta Method)

var.theta <- t(R)%*%V_ols%*%R
se.theta <- sqrt(var.theta)
print(se.theta)

# Jackknife

theta.hat.jack <-rep(0,n)

for (i in 1:n) {
    theta.hat.jack[i] <- beta.hat.jack[3,i]+beta.hat.jack[4,i]+beta.hat.jack[5,i]
}

theta.bar.jack <- mean(theta.hat.jack)
var.jack <- (n-1)*mean((theta.hat.jack-theta.bar.jack)^2)
se.jack <- sqrt(var.jack)
se.jack



# Bootstrap

B<- 10000
```

```
theta.hat.boot <- rep(0,B)
for (b in 1:B){
    theta.hat.boot[b] <- beta.hat.boot[3,b]+beta.hat.boot[4,b]+beta.hat.boot[5,b]
}

theta.bar.boot <- mean(theta.hat.boot)
var.boot <- (B/(B-1))*mean((theta.hat.boot-theta.bar.boot)^2)
se.boot <- sqrt(var.boot)
se.boot

#--------------------------------------------------------------------------------
```

	Estimate	Asymptotic SE	Jackknife SE	Bootstrap SE
θ	0.6429698	0.4443914	0.4626814	0.4515477

c) The percentile bootstrap $100(1-\alpha)\%$ confidence interval:

$$C = [q^*_{\frac{\alpha}{2}}, q^*_{1-\frac{\alpha}{2}}]$$

where $q^*_{\frac{\alpha}{2}}$ and $q^*_{1-\frac{\alpha}{2}}$ are the $(\frac{\alpha}{2})$ and $(1-\frac{\alpha}{2})$ quantiles of bootstrap sample $\{\hat{\theta}^*_b\}^B_{b=1}$.

Code:

```
# c) Confidence intervals for theta

# Percentile method

alpha = 0.05
q.star.alphas <- quantile(theta.hat.boot, probs = c(alpha/2, 1-alpha/2))

CI_percentile <- q.star.alphas
CI_percentile
#------------------------------------------------------------
```

2.5%	97.5%
-0.2423459	1.5288141

For the BC$_a$ method, we calculate $z_\alpha = \Phi^{-1}(\alpha)$ and $z^*_0 = \Phi^{-1}(p^*)$ where $p^* = \frac{1}{B}\sum_{i=b}^{B} 1(\hat{\theta}^*_b \leq \hat{\theta})$. Then we estimate skewness using the Jackknife estimator:

$$\hat{a}^{\text{jack}} = \frac{\sum_{j=1}^{n} (\bar{\theta} - \hat{\theta}_{-i})^3}{6(\sum_{i=1}^{n} (\bar{\theta} - \hat{\theta}_{-i})^2)^{\frac{3}{2}}}.$$

Then, for $\alpha \in (0, 1)$ we define the bias-corrected version of α:

$$x(\alpha) = \Phi\Big(z_0^* + \frac{z_\alpha + z_0^*}{1 - \hat{a}^{\text{jack}}(z_\alpha + z_0^*)}\Big).$$

The bias corrected version of $100(1-\alpha)\%$ CI for θ is

$$C^{\text{bc}} = [q^*_{x(\alpha/2)}, q^*_{x(1-\alpha/2)}].$$

Code:

```
# BC_a Percentile interval

alphas <- c(alpha/2, 1-alpha/2)

# Evaluate z_alpha for each alpha values

z.alphas <- qnorm(alphas)

# Evaluate z0.star

p.star <- mean(theta.hat.boot <= theta.hat)
z0.star <- qnorm(p.star)

diff.jack.t <- theta.bar.jack - theta.hat.jack
a.jack.num <- sum(diff.jack^3)
a.jack.den <- 6*sum(diff.jack^2)^(3/2)
a.jack <- a.jack.num/a.jack.den

correction <- (z.alphas+z0.star)/(1-a.jack*(z.alphas+z0.star))
x.alphas <-pnorm(z0.star+correction)
q.star.x.alphas <- quantile(theta.hat.boot, probs = x.alphas)

CI_BCa <- q.star.x.alphas
CI_BCa
```

#--

2.838307%	97.83019%
-0.2195445	1.5580979

PROBLEM 9.27:

In Section 8.12 we reported estimates from Mankiw, Romer and Weil (1992). We reported estimation both by unrestricted least squares and by constrained estimation, imposing the constraint that three coefficients (second, third and fourth coefficients) sum to zero as implied by the Solow growth theory. Using the same dataset MRW1992 estimate the unrestricted model and test the hypothesis that the three coefficients sum to zero.

SOLUTION:

Code:

```
mrw <- read.table("/Users/student/Desktop/Spring21/582/HW1/MRW1992.txt", header=TRUE)
N <- matrix(mrw$N, ncol =1)
lndY <- matrix(log(mrw$Y85)-log(mrw$Y60),ncol=1)
lnY60 <- matrix(log(mrw$Y60), ncol=1)
lnI <- matrix(log(mrw$invest/100), ncol =1)
lnG <- matrix(log(mrw$pop_growth/100+0.05), ncol=1)
lnS <- matrix(log(mrw$school/100), ncol =1)
X <- as.matrix(cbind(lnY60, lnI, lnG, lnS,matrix(1,nrow(lndY),1)))
x <- X[N==1,]
y <- lndY[N==1]

# Creating a function for estimating beta

estimate.beta <- function(z,y){
    n<- nrow(z)
    k<- ncol(z)
    invz <- solve(t(z)%*%z)
    b_ols <- solve(t(z)%*%z)%*%(t(z)%*%y)
    return(b_ols)

}


# Unrestricted regression

beta.hat <- estimate.beta(x,y)

# Standard error
```

```
n <-nrow(x)
k<- ncol(x)
invx <- solve(t(x)%*%x)
e_ols <- rep((y-x%*%beta.hat), times=k)
xe_ols <-x*e_ols
V_ols <- (n/(n-k))*invx%*%(t(xe_ols)%*%xe_ols)%*%invx
se_ols <- sqrt(diag(V_ols))
print(beta.hat)
print(se_ols)
#--------------------------------------------------------------
```

	Estimate	**Standard Error**
$\log(GDP_{1960})$	-0.2883737	0.05427556
$\log(I)$	0.5237367	0.10729137
$\log(G)$	-0.5056565	0.23603269
$\log(School)$	0.2311171	0.06640414
Intercept	3.0215222	0.73730944

Let $\theta = \beta_2 + \beta_3 + \beta_4$. Then, $\theta = r(\beta) = R'\beta$ is a linear function of β with $R' = [0\ 1\ 1\ 1\ 0]$. In this case, $\theta_0 = 0$. Then the Wald statistic is:

$$W = (R'\hat{\beta})'(R'\hat{V}_{\hat{\beta}}R)^{-1}(R'\hat{\beta})$$

Since, $q = 1$ the Wald statistic follows an asymptotic distribution of chi-square with 1 degree of freedom. We will conduct a test of asymptotic size $\alpha = 0.05$. Then we find a critical value c such that $0.05 = 1 - G_1(c)$ and reject H_0 if $W > c$. The following code is used for this problem:

```
# Test (Wald Statistic)

R <- c(0,1,1,1,0)
c<- 0
q <- length(c)
V_r <- solve(t(R)%*%V_ols%*%R)
W <- t(t(R)%*%beta.hat-c)%*%V_r%*%t(t(R)%*%beta.hat-c)
alpha <- 0.05
C <- qchisq(1-alpha, q)
print(W)
print(C)
```

```
if (W>C){
    print("Reject H0")
} else {
    print("Accept H0")
}

#--------------------------------------------------------------
```

We found that $W = 0.8362141$ and the critical value $c = 3.84159$. Therefore, we accept H_0 in this Wald test of asymptotic size 0.05 since $W < c$.

PROBLEM 10.29:

In Excercise 9.27 you estimated the Mankiw, Romer, and Weil (1992) unrestricted regression. Let θ be the sm of the second, third, and fourth coefficients.

a) Estimate the regression by unrestricted least squares and report standard errors calculated by asymptotic, jackknife and the bootstrap.

b) Estimate θ and report standard errors calculated by asymptotic, jackknife and the bootstrap.

c) Report confidence intervals for θ using the percentile and BC methods.

SOLUTION:

---

a) Code:

```
# Asymptotic
n <-nrow(x)
k<- ncol(x)
beta.hat <- estimate.beta(x,y)
invx <- solve(t(x)%*%x)
e_ols <- rep((y-x%*%beta.hat), times=k)
xe_ols <-x*e_ols
V_ols <- (n/(n-k))*invx%*%(t(xe_ols)%*%xe_ols)%*%invx
se_ols <- sqrt(diag(V_ols))
print(beta.hat)
print(se_ols)

# Jackknife

beta.hat.jack <- matrix(data=0, nrow = k, ncol = n)

for (i in 1:n) {
    df.loo.i <- x[-i,]
    p<- y[-i]
    beta.hat.jack[,i] <- estimate.beta(z=df.loo.i,y=p)
}

beta.bar.jack <- rowMeans(beta.hat.jack)
diff.jack <- (beta.hat.jack-beta.bar.jack)
var.jack <- ((n-1)/n)*(diff.jack%*%t(diff.jack))
```

```
se.beta.jack <- sqrt(diag(var.jack))
se.beta.jack

# Bootstrap

B<- 10000
set.seed(1)
beta.hat.boot <- matrix(0, nrow=k, ncol =B)

for (b in 1:B){
# Construct a b-th bootstrap sample
    idx.b <- sample(n, replace = TRUE)
    df.b <- x[idx.b,]
    y.b <- y[idx.b]
    beta.hat.boot[,b] <- estimate.beta(z=df.b, y=y.b)
}

beta.bar.boot <- rowMeans(beta.hat.boot)
diff.boot <- (beta.hat.boot-beta.bar.boot)
var.boot <- (1/(B-1))*(diff.boot %*% t(diff.boot))
se.beta.boot <- sqrt(diag(var.boot))
se.beta.boot
#--------------------------------------------------------------
```

	Estimate	**Asymptotic SE**	**Jackknife SE**	**Bootstrap SE**
$\log(GDP_{1960})$	-0.2883737	0.05427556	0.05687096	0.05456973
$\log(I)$	0.5237367	0.10729137	0.11157415	0.10777204
$\log(G)$	-0.5056565	0.23603269	0.24473366	0.24000301
$\log(School)$	0.2311171	0.06640414	0.06900648	0.06731368
Intercept	3.0215222	0.73730944	0.75631932	0.74590081

b) Code:

```
# Asymptotic

theta.hat <- beta.hat[2]+beta.hat[3]+beta.hat[4]
theta.hat

# Standard error (Using Delta Method)
var.theta <- t(R)%*%V_ols%*%R
se.theta <- sqrt(var.theta)
```

```
print(se.theta)

# Jackknife

theta.hat.jack <-rep(0,n)

for (i in 1:n) {
    theta.hat.jack[i] <- beta.hat.jack[2,i] + beta.hat.jack[3,i]+beta.hat.jack[4,i]
}

theta.bar.jack <- mean(theta.hat.jack)
var.theta.jack <- (n-1)*mean((theta.hat.jack-theta.bar.jack)^2)
se.theta.jack <- sqrt(var.theta.jack)
se.theta.jack

B<- 10000

theta.hat.boot <- rep(0,B)
for (b in 1:B){
    theta.hat.boot[b] <- beta.hat.boot[2,b]+beta.hat.boot[3,b]+beta.hat.boot[4,b]
}

theta.bar.boot <- mean(theta.hat.boot)
var.theta.boot <- (B/(B-1))*mean((theta.hat.boot-theta.bar.boot)^2)
se.theta.boot <- sqrt(var.theta.boot)
se.theta.boot
#------------------------------------------------------------------------------
```

	Estimate	**Asymptotic SE**	**Jackknife SE**	**Bootstrap SE**
θ	0.2491973	0.2725114	0.2809195	0.2758996

c) The percentile bootstrap $100(1-\alpha)\%$ confidence interval:

$$C = [q^*_{\frac{\alpha}{2}}, q^*_{1-\frac{\alpha}{2}}]$$

where $q^*_{\frac{\alpha}{2}}$ and $q^*_{1-\frac{\alpha}{2}}$ are the $(\frac{\alpha}{2})$ and $(1-\frac{\alpha}{2})$ quantiles of bootstrap sample $\{\hat{\theta}^*_b\}_{b=1}^B$.

Code:

```
# c) Confidence intervals for theta
```

```
# Percentile method

alpha <- 0.05
q.star.alphas <- quantile(theta.hat.boot, probs = c(alpha/2, 1-alpha/2))

CI_percentile <- q.star.alphas
CI_percentile
#-----------------------------------------------------------------------
```

2.5%	97.5%
-0.2636315	0.8121308

For the BC$_a$ method, we calculate $z_\alpha = \Phi^{-1}(\alpha)$ and $z_0^* = \Phi^{-1}(p^*)$ where $p^* = \frac{1}{B}\sum_{i=b}^{B} 1(\hat{\theta}_b^* \leq \hat{\theta})$. Then we estimate skewness using the Jackknife estimator:

$$\hat{a}^{\text{jack}} = \frac{\sum_{j=1}^{n}(\bar{\theta} - \hat{\theta}_{-i})^3}{6(\sum_{i=1}^{n}(\bar{\theta} - \hat{\theta}_{-i})^2)^{\frac{3}{2}}}.$$

Then, for $\alpha \in (0,1)$ we define the bias-corrected version of α:

$$x(\alpha) = \Phi\Big(z_0^* + \frac{z_\alpha + z_0^*}{1 - \hat{a}^{\text{jack}}(z_\alpha + z_0^*)}\Big).$$

The bias corrected version of $100(1-\alpha)\%$ CI for θ is

$$C^{\text{bc}} = [q^*_{x(\alpha/2)}, q^*_{x(1-\alpha/2)}].$$

Code:

```
# BC_a Percentile interval

alphas <- c(alpha/2, 1-alpha/2)

# Evaluate z_alpha for each alpha values

z.alphas <- qnorm(alphas)

# Evaluate z0.star

p.star <- mean(theta.hat.boot <= theta.hat)
```

```
z0.star <- qnorm(p.star)

diff.jack.t <- theta.bar.jack - theta.hat.jack
a.jack.num <- sum(diff.jack^3)
a.jack.den <- 6*sum(diff.jack^2)^(3/2)
a.jack <- a.jack.num/a.jack.den

correction <- (z.alphas+z0.star)/(1-a.jack*(z.alphas+z0.star))
x.alphas <-pnorm(z0.star+correction)
q.star.x.alphas <- quantile(theta.hat.boot, probs = x.alphas)

CI_BCa <- q.star.x.alphas
CI_BCa
#-------------------------------------------------------------------
```

1.681433%	96.41663%
-0.3181800	0.7702345

PROBLEM 7.28:

As in Exercise 3.26, use the cps09mar dataset and the subsample of the white male Hispanics. Estimate the regression

$$\log(\hat{wage}) = \beta_1 education + \beta_2 experience + \beta_3 \frac{experience^2}{100} + \beta_4.$$

a) Report the coefficient estimates and robust standard errors.

b) Let θ be the ratio of the return to one year of education to the return of one year of experience for $experience = 10$. Write θ as a function of the regression coefficients and variables. Compute $\hat{\theta}$ from the estimated model.

c) Write out the formula for the asymptotic standard error for $\hat{\theta}$ as a function of the covariance matrix for $\hat{\beta}$. Compute $s(\hat{\theta})$ from the estimated model.

d) Construct a 90% asymptotic confidence interval for θ from the estimated model.

e) Compute the regression function at $education = 12$ and $experience = 20$. Compute a 95% confidence interval for the regression function at this point.

f) Consider an out-of-sample individual with 16 years of education and 5 years experience. Construct an 80% forecast interval for their log wage and wage.

SOLUTION:

---

a) Code:

```
library(tidyverse) # Packages for data manipulation
library(readxl) # Read xlsx file
library(sandwich) # Robust standard error
library(lmtest)

cps09mar <- read_excel("cps09mar.xlsx")

cps09mar.sub <- cps09mar %>%
     mutate(log_wage = log(earnings/(hours*week))) %>%
     mutate(experience = age-education-6) %>%
     mutate(experience2 = experience^2/100) %>%
     filter(female == 0, hisp ==1,race ==1)

OLS.out <- lm(log_wage ~ education+experience+experience2, data = cps09mar.sub)
```

```
coeftest(OLS.out, vcov = vcovHC(OLS.out, type="HC1"))
beta.hat <-coef(OLS.out)
beta.hat
#-------------------------------------------------------------
```

	Estimate	**Standard Error**
education	0.09044896	0.0029165
experience	0.03537968	0.0025854
$\frac{\text{experience}^2}{100}$	-0.04650594	0.0053069
Intercept	1.18520948	0.0461003

b) The return to one year of education is given by:

$$\frac{\partial \log(wage)}{\partial education} = \beta_1$$

The return to one year of experience is given by:

$$\frac{\partial \log(wage)}{\partial experience} = \beta_2 + \frac{2\beta_3}{100} experience$$

For $experience = 10$, the return to one year of experience is given by:

$$\beta_2 + \frac{\beta_3}{5}$$

Therefore:

$$\theta = \frac{\beta_1}{\beta_2 + \frac{\beta_3}{5}} = \frac{5\beta_1}{5\beta_2 + \beta_3}$$

Code:

```
theta.hat <- 5*beta.hat[2]/(5*beta.hat[3]+beta.hat[4])

theta.hat
#-------------------------------------------------------------
```

We find that $\hat{\theta} = 3.468335$.

c) We have that $\theta = r(\beta) = \frac{5\beta_1}{5\beta_2 + \beta_3}$. We will use the Delta Method. First, we calculate:

$$R' = \left(\frac{\partial r(\beta)}{\partial \beta_1} \quad \frac{\partial r(\beta)}{\partial \beta_2} \quad \frac{\partial r(\beta)}{\partial \beta_3} \quad \frac{\partial r(\beta)}{\partial \beta_4}\right)$$

$$R' = \left(\frac{5}{5\beta_2 + \beta_3} \quad -\frac{25\beta_1}{(5\beta_2 + \beta_3)^2} \quad -\frac{5\beta_1}{(5\beta_2 + \beta_3)^2} \quad 0\right)$$

Then:

$$V_\theta = R' V_\beta R$$

and the estimate is given by:

$$\hat{V}_\theta = \hat{R}' \hat{V}_{\hat{\beta}} \hat{R}.$$

Then, the standard error is given by:

$$s(\theta) = \sqrt{R' V_\beta R}$$

and the estimate is given by:

$$s(\hat{\theta}) = \sqrt{\hat{R}' \hat{V}_{\hat{\beta}} \hat{R}}.$$

Code:

```
# c)

V_ols <- vcovHC(OLS.out, type = "HC1")
R.hat<-c(0,5/(5*beta.hat[3]+beta.hat[4]),(-25*beta.hat[2])/(5*beta.hat[3]+beta.hat[4])^2
,(-5*beta.hat[2])/(5*beta.hat[3]+beta.hat[4])^2)
V.theta <- t(R.hat)%*%V_ols%*%R.hat
se.theta.hat <-sqrt(V.theta)
se.theta.hat
#---------------------------------------------------------------------------------
```

We found that $s(\hat{\theta}) = 0.2268414$.

d) We find a critical value c such that c is the $1 - \frac{\alpha}{2}$ quantile of the standard normal distribution. Then the confidence interval is given by:

$$\hat{C} = [\hat{\theta} - c \times s(\hat{\theta}), \hat{\theta} + c \times s(\hat{\theta})]$$

We used the following piece of code:

```
alpha <- 0.1
c<- qnorm(1-alpha/2)
u.bound <- theta.hat+c*se.theta.hat
l.bound <- theta.hat-c*se.theta.hat
text <- 'Confidence interval:[ ${l.bound} , ${u.bound}]'
cat(str_interp(text))

#-------------------------------------------------------------------------
```

Thus:

$$\hat{C} = [3.0952,\ 3.8415].$$

e) We plug in $education = 12$ and $experience = 20$ into the regression equation given in the question and using the values of β we found above.

Code:

```
# e)
educ <- 12
exp <- 20

coef <- c(1,educ, exp, exp^2/100)
reg <- t(coef)%*%beta.hat
#-----------------------------------------------------------
```

We find that $\log(\hat{wage}) = 2.792167$. In order to construct the confidence interval, we need to find the standard error. The standard error is calculated by the Delta Method. We have $R = [12,\ 20,\ \frac{20^2}{100},\ 1]'$ and then by the Delta method, the standard error is given by: $\sqrt{R'\hat{V}_{\hat{\beta}}R}$. We again pick c such that it is the $1 - \frac{\alpha}{2}$ quantile of the standard normal distribution. The following piece of code is used to construct the confidence interval.

```
# SE (by the Delta method)

V.reg<- t(coef)%*%V_ols%*%coef
se.reg <- sqrt(V.reg)
```

```
# Confidence interval
alpha <- 0.05
c <- qnorm(1-alpha/2)
u.bound <- reg+c*se.reg
l.bound <- reg-c*se.reg
text <- 'Confidence interval:[ ${l.bound} , ${u.bound}]'
cat(str_interp(text))
```

The 95% confidence interval is given by:

$$[2.7693,\ 2.8150].$$

f) For the forecast interval, we require σ^2. The standard error of the forecast is $\hat{s}(x) = \sqrt{\sigma^2 + x'\hat{V}_{\hat{\beta}}x}$. Then the confidence interval is given by: $[x'\hat{\beta} - c * \hat{s}(x), x'\hat{\beta} + c * \hat{s}(x)]$ where $x = [16,\ 5,\ \frac{5^2}{100},\ 1]'$, that is, the out-of-sample information. Hence, the 80% forecast interval for $\log wage$ is given by:

$$\hat{C}_{\log(wage)} = [2.0621,\ 3.5332]$$

The 80% forecast interval for $wage$ is given by (we apply the exponential function to both endpoints of the confidence interval for $\log(wage)$:

$$\hat{C}_{wage} = [7.8625,\ 34.2343].$$

The following piece of code was used to calculate the above confidence intervals:

```
# f)
# Computing the regression function with the new values

out.educ <- 16
out.exp <- 5
x <- c(1, out.educ, out.exp, (out.exp)^2/100)
f_reg <- t(x)%*%beta.hat

# Standard error (we need sigma.hat^2)

e<- residuals(OLS.out)
sigma_2 <- mean(e^2)
V.x <- sigma_2+t(x)%*%V_ols%*%x
```

```
se.x <- sqrt(V.x)

# 80% confidence interval for log(wage)

alpha <- 0.2
c<- qnorm(1-alpha/2)
u.bound <- f_reg+c*se.x
l.bound <- f_reg-c*se.x
text <- 'Confidence interval:[ ${l.bound} , ${u.bound}]'
cat(str_interp(text))

# 80% confidence interval for wage

exp.u.bound<- exp(u.bound)
exp.l.bound<- exp(l.bound)
text <- 'Confidence interval:[ ${exp.l.bound} , ${exp.u.bound}]'
cat(str_interp(text))
#-------------------------------------------------------------------------------------
```

PROBLEM 10.30:

In Exercise 7.28, you estimated a wage regression with the cps09mar dataset and the sub-sample of white Male Hispanics. Further restrict teh sample to those never-married and live in the Midwest region. As in subquestion (b) let θ be the ratio of the return to one year of education to the return of one year of experience.

a) Estimate θ and report standard errors calculated by asymptotic, jackknife and the bootstrap.

b) Explain the discrepancy between the standard errors.

c) Report confidence intervals for θ using the BC percentile method.

SOLUTION:

---

a) We will let θ to be the ratio of the return to one year of education to the return of one year of experience. We will assume this is for $experience = 10$. From 7.28, we have that:

$$\theta = \frac{5\beta_1}{5\beta_2 + \beta_3}.$$

The asymptotic standard error for θ is given by:

$$s(\theta) = \sqrt{R' V_\beta R}$$

for $R' = \begin{pmatrix} \frac{5}{5\beta_2+\beta_3} & -\frac{25\beta_1}{(5\beta_2+\beta_3)^2} & -\frac{5\beta_1}{(5\beta_2+\beta_3)^2} & 0 \end{pmatrix}$.

The following table contains the estimate θ and the standard errors calculated by asymptotic, jackknife and the bootstrap:

	Estimate	Asymptotic SE	Jackknife SE	Bootstrap SE
θ	2.899323	0.7603923	0.8229674	1.376148

The following code was used to generate the values in the table above:

```
cps09mar.sub1 <- cps09mar.sub %>%
filter(region ==2, marital == 7)
```

```
# Creating a function to estimate theta

estimate.theta <-function(data){
    OLS1.out <- lm(log_wage ~ education+experience+experience2, data = data)
    b_ols <- coef(OLS1.out)
    theta.hat <- 5*b_ols[2]/(5*b_ols[3]+b_ols[4])
    return(theta.hat)
}

# Asymptotic

OLS2.out <- lm(log_wage ~ education+experience+experience2, data = cps09mar.sub1)
b_as <- coef(OLS2.out)
theta.as <- 5*b_as[2]/(5*b_as[3]+b_as[4])
theta.as
V_as <- vcovHC(OLS2.out, type = "HC1")
R.hat.as <- c(0, 5/(5*b_as[3]+b_as[4]), (-25*b_as[2])/(5*b_as[3]+b_as[4])^2
, (-5*b_as[2])/(5*b_as[3]+b_as[4])^2)
V.theta.as <- t(R.hat.as)%*%V_as%*%R.hat.as
se.theta.as <-sqrt(V.theta.as)
se.theta.as

# Jackknife

n <- nrow(cps09mar.sub1)
theta.hat.jack <-rep(0,n)

for (i in 1:n) {
    df.loo.i <- cps09mar.sub1[-i,]
    theta.hat.jack[i] <- estimate.theta(data = df.loo.i)
}

theta.bar.jack <- mean(theta.hat.jack)
var.jack <- (n-1)*mean((theta.hat.jack-theta.bar.jack)^2)
se.jack <- sqrt(var.jack)
se.jack


# Bootstrap

set.seed(1000)
```

```
B<- 10000
theta.hat.boot <- rep(0,B)
for (b in 1:B){
# Construct a b-th bootstrap sample
    idx.b <- sample(n, replace =TRUE)
    df.b <- cps09mar.sub1[idx.b,]
theta.hat.boot[b]<-estimate.theta(data=df.b)
}

theta.bar.boot <- mean(theta.hat.boot)
var.boot <- (B/(B-1))*mean((theta.hat.boot-theta.bar.boot)^2)
se.boot <- sqrt(var.boot)
se.boot


#------------------------------------------------------------------------------------
```

b) I think the discrepancy between the standard errors are coming from the small sample size ($n = 99$). The asymptotic standard error is not a good estimate because of this small n. The bootstrap resamples will always have an extra observation compared to the jackknife samples. The bootstrap resample with small sample size might result in samples with a lot of outliers resulting in the difference with jackknife. If we suppose that the observations lie in the range [min,max]. The higher the number of observations, it is likely that there are higher number of observations that are not near the boundaries. So, when we sample with replacement the probability of having a lot of outliers will be less. However, when we have a smaller sample size we do not have a lot of observations that are away from the boundaries from the interval. Hence, when we sample with replacement the probability of getting an observation closer to the boundary is higher resulting in more outliers.

c) The following are the steps to calculate the confidence interval using the BC percentile method:

1. Calculate $z_\alpha = \Phi^{-1}(\alpha)$ and $z_0^* = \Phi^{-1}(p^*)$ where $p^* = \frac{1}{B}\sum_{i=b}^{B} 1(\hat{\theta}_b^* \leq \hat{\theta})$

2. For $\alpha \in (0,1)$, define the bias-corrected version of α,

$$x(\alpha) = \Phi(z_\alpha + 2z_0^*)$$

3. The bias-corrected version of $100(1-\alpha)\%$ CI for θ is

$$C^{bc} = [q^*_{x(\frac{\alpha}{2})}, q^*_{x(1-\frac{\alpha}{2})}].$$

The following code generates the confidence interval for θ using the BC percentile method:

```
# BC Percentile Interval

alpha <- 0.05
alphas <- c(alpha/2, 1-alpha/2)

# Evaluate z_alpha for each alpha values
z.alphas<-qnorm(alphas)

# Evaluate z0.star

p.star <- mean(theta.hat.boot <= theta.hat)
z0.star <- qnorm(p.star)

# Calculate x(alpha) for each alpha values
x.alphas <- pnorm(z.alphas+2*z0.star)

q.star.x.alphas <- quantile(theta.hat.boot, probs = x.alphas)
CI_BC<- q.star.x.alphas
CI_BC
#---------------------------------------------------
```

22.06992%	99.91839%
2.415283	12.252406

PROBLEM 4.26:

Extend the empirical analysis reported in Section 4.23 using the DDK2011 dataset on the website. Do a regression of standardized test score (*totalscore* normalized to have zero mean and variance 1) on tracking, age, gender, being assigned to the contract teacher, and student's percentile in the initial distribution. (The sample size will be smaller as some observations have missing variables.) Calculate standard errors using both the conventional robust formula, and clustering based on the school.

a) Compare the two sets of standard errors. Which standard error changes the most by clustering? Which changes the least?

b) How does the coefficient on *tracking* change by inclusion of the individual controls (in comparison to the results from (4.55))?

SOLUTION:

---

a) Code:

```
library(tidyverse) # Packages for data manipulation
library(readxl) # Read xlsx file
library(sandwich) # Robust standard error
library(lmtest)


DDK2011 <- read_excel("DDK2011.xlsx")
n.DDK2011 <- nrow(DDK2011)

DDK2011.sub <- DDK2011 %>%
    mutate(testscore = scale(totalscore)) %>%
# Removing observations with missing variables
    filter(testscore != ".", tracking != ".", agetest != ".", etpteacher != ".",
percentile != ".", girl != '.') %>%
    mutate_all(as.numeric)


OLS.out <- lm(testscore ~ tracking+agetest+girl+etpteacher+percentile,
data = DDK2011.sub)
b.hat <-coef(OLS.out)
b.hat

# Robust standard error
```

```
V.HC1 <- vcovHC(OLS.out, type = "HC1")
se.HC1 <- sqrt(diag(V.HC1))
se.HC1

# Clustered standard error

V.cluster <- vcovCL(OLS.out, cluster =~ schoolid)
se.cluster<- sqrt(diag(V.cluster))
se.cluster

# Calculating the absolute difference between the two standard errors

diff <- abs(se.HC1-se.cluster)
diff

# Calculating the percentage difference in the two standard errors

percent<- (se.cluster-se.HC1)/(se.HC1)*100
percent

#---------------------------------------------------------------------------------
```

	Estimate	**Robust SE**	**Clustered SE**	**Absolute Difference**	**% difference**
Intercept	-0.72905404	0.0809656002	0.1297340334	0.048768433	60.23352
tracking	0.17251170	0.0240222032	0.0761818728	0.052159670	217.13108
agetest	-0.04080292	0.0084928279	0.0133115688	0.004818741	56.73895
girl	0.08120349	0.0240886042	0.0284988400	0.004410236	18.30839
etpteacher	0.17987572	0.0237053545	0.0374764280	0.013771074	58.09267
percentile	0.01731724	0.0004245766	0.0007202686	0.000295692	69.64395

The standard error of the coefficient of tracking changes the most in terms of the absolute difference and also in terms of the percentage difference by clustering. The standard error of the coefficient of girl changes the least in percentage terms and the standard error of the coefficient of percentile changes the least in absolute terms by clustering.

b) The coefficient on *tracking* increases with the inclusion of the individual controls. The omitted variables in the result from (4.55) biased the coefficient on *tracking* downwards. This indicates that the variables with positive coefficients in the long regression is probably negatively correlated with *tracking* and *agetest* is probably positively correlated with *tracking*. We checked for the correlation and found that our guess was indeed true except for the correlation between *tracking* and *girl* which is slightly positive. The following code was used for this:

```
> cor(DDK2011.sub$tracking, DDK2011.sub$percentile)
[1] -0.01555854
> cor(DDK2011.sub$tracking, DDK2011.sub$agetest)
[1] 0.07443057
> cor(DDK2011.sub$tracking, DDK2011.sub$girl)
[1] 0.02607656
> cor(DDK2011.sub$tracking, DDK2011.sub$etpteacher)
[1] -0.02381453
>
```

PROBLEM 10.31:

In Exercise 4.26 you extended the work from Duflo, Dupas and Kremer(2011). Repeat that regression, now calculating the standard error by cluster bootstrap. Report a BC_a confidence interval for each coefficient.

SOLUTION:

---

a) Code:

```
DDK2011 <- read_excel("DDK2011.xlsx")
DDK2011 <- DDK2011 %>%
    mutate(testscore = scale(totalscore)) %>%
# Removing observations with missing variables
    filter(testscore != ".", tracking != ".", agetest != ".", etpteacher != ".",
    percentile != ".", girl != '.') %>%
    mutate_all(as.numeric)

n.DDK2011 <- nrow(DDK2011)
DDK2011_group <- DDK2011 %>% group_nest(schoolid)

cluster <- unique(DDK2011_group$schoolid)
n.cluster <- length(cluster)

# Jackknife

n.beta <- 6
beta.cluster.loo <- matrix(0, nrow =n.beta, ncol =n.cluster)

for (i in 1:n.cluster){
    df.cluster.loo.i <- DDK2011_group[-i,] %>% unnest(data)
    ols.out.i <- lm(testscore ~ tracking+agetest+girl+etpteacher+percentile, data = df.c
    beta.loo.i <- coef(ols.out.i)

    beta.cluster.loo[,i] <- beta.loo.i
}

beta.bar.cluster <- rowMeans(beta.cluster.loo)
diff.jack.cluster <- beta.cluster.loo-beta.bar.cluster
var.cluster.jack <- (n.cluster-1)/n.cluster*diff.jack.cluster%*%t(diff.jack.cluster)
se.cluster.jack <- sqrt(diag(var.cluster.jack))
```

```
se.cluster.jack

#---------------------------------------------------------------------------------------
# Bootstrap

set.seed(1)
DDK2011 <- read_excel("DDK2011.xlsx")
DDK2011 <- DDK2011 %>%
    mutate(testscore = scale(totalscore)) %>%
# Removing observations with missing variables
    filter(testscore != ".", tracking != ".", agetest != ".", etpteacher != ".",
    percentile != ".", girl != '.') %>%
    mutate_all(as.numeric)

n.DDK2011 <- nrow(DDK2011)
DDK2011_group <- DDK2011 %>% group_nest(schoolid)

cluster <- unique(DDK2011_group$schoolid)
n.cluster <- length(cluster)
n.beta <- 6

B <- 10000
beta.hat.boot.cluster <- matrix(0, nrow = n.beta, ncol = B)

for (b in 1:B){
    idx.boot <- sample(1:n.cluster, size = n.cluster, replace = TRUE)
    df.boot.b <- DDK2011_group[idx.boot, ] %>% unnest(data)
    ols.out.b <- lm(testscore ~ tracking+agetest+girl+etpteacher+percentile, data = df.b
    beta.boot.b <- coef(ols.out.b)
    beta.hat.boot.cluster[, b] <- beta.boot.b
}

beta.bar.boot.cluster <- rowMeans(beta.hat.boot.cluster)
diff.boot.cluster <- (beta.hat.boot.cluster - beta.bar.boot.cluster)
var.boot.cluster <- (1/(B-1))*(diff.boot.cluster%*%t(diff.boot.cluster))
se.boot.cluster <- sqrt(diag(var.boot.cluster))
se.boot.cluster
#---------------------------------------------------------------------------------------
```

	Estimate	Robust SE	Clustered SE	Jackknife Cluster SE	Bootstrap Cluster SE
Intercept	-0.72905404	0.0809656002	0.1297340334	0.1314840692	0.1288463968
tracking	0.17251170	0.0240222032	0.0761818728	0.0769932513	0.0768193932
agetest	-0.04080292	0.0084928279	0.0133115688	0.0134690635	0.0132705020
girl	0.08120349	0.0240886042	0.0284988400	0.0286870805	0.0285862791
etpteacher	0.17987572	0.0237053545	0.0374764280	0.0377870687	0.0374092540
percentile	0.01731724	0.0004245766	0.0007202686	0.0007252765	0.0007194658

We did not run the code for the robust standard error and the clustered standard error in this section because we solved that problem and calculated the values in the previous problem. Those values have been taken from the solution to Exercise 4.26.

BC$_a$ Confidence Intervals:

Intercept:

2.465498%	97.46532%
-0.9824817	-0.4741744

Tracking:

2.751852%	97.72917%
0.02393461	0.32769810

Agetest:

3.156466%	98.07644%
-0.06502531	-0.01272990

Girl:

2.680383%	97.66845%
0.02606974	0.13790803

Etpteacher:

2.497645%	97.49765%
0.1053798	0.2540949

Percentile:

2.204925%	97.18032%
0.01585208	0.01867208

The following code is used to calculate these confidence intervals:

```
# Constructing BC_a percentile intervals for each coefficient

alpha <- 0.05
alphas <- c(alpha/2, 1-alpha/2)

# evaluate z_alpha for each alpha values
z.alphas <- qnorm(alphas)


for (i in 1:n.beta){
    p.star.i <- mean(beta.hat.boot.cluster[i,] <= b.hat[i])
    z0.star.i <- qnorm(p.star.i)
    beta.bar.loo <- mean(beta.cluster.loo[i,])
    diff.jack <- beta.bar.loo-beta.cluster.loo[i,]
    a.jack.num <- sum(diff.jack^3)
    a.jack.den <- 6*sum(diff.jack^2)^(3/2)
    a.jack <- a.jack.num/a.jack.den
    correction <-     (z.alphas+z0.star.i)/(1-a.jack*(z.alphas+z0.star.i))
    x.alphas <-pnorm(z0.star.i+correction)
    q.star.x.alphas <- quantile(beta.hat.boot.cluster[i,], probs =x.alphas)
    CI_BCa <- q.star.x.alphas
    print(CI_BCa)
}
#---------------------------------------------------------------------------
```