Problem Set 4

Tate Mason

Question 1 - Hansen 7.17

Part A

Part B

Part C

Question 2 - Hansen 7.28

```r
library(haven)
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```r
rm(list = ls())

dat <- read_dta('~/SchoolWork/Sem2/Metrics/PSets/PS3/cps09mar.dta')
sample <- (dat[,11]==1)&(dat[,2]==0)&(dat[,3]==1)
df <- dat[sample,]

y <- as.matrix(log(df[,5]/(df[,6]*df[,7])))
exp <- df[,1]-df[,4]-6
exp2 <- (exp^2)/100

x_df <- data.frame(
  education = df[,4],
  experience = exp,
  exp_squared = exp2,
  intercept = 1
)
x <- as.matrix(x_df)

xx <- t(x)%*%x
xy <- t(x)%*%y
beta <- solve(xx,xy)

fitted <- x %*% beta
resid <- y - fitted

n <- nrow(x)
k <- ncol(x)
xx_inv <- solve(t(x)%*%x)

df <- n-k

hc0 <- matrix(0, nrow=k, ncol=k)
for (i in 1:n) {
  xi <- matrix(x[i,], nrow=k)
  hc0 <- hc0 + resid[i]^2 * (xi %*% t(xi))
}

V_robust <- xx_inv %*% hc0 %*% xx_inv
robust_se <- sqrt(diag(V_robust))
```

Part B

```
theta_hat <- beta[1]/(beta[2]+beta[3]/5)
cat("\nPart B: Estimated Theta:", theta_hat, "\n")
```

```
Part B: Estimated Theta: 3.468335
```

Part C

```
gradient_g <- c(
  1/(beta[2] + beta[3]/5),
  -beta[1]/(beta[2]+beta[3]/5)^2,
  -beta[1]/(5*(beta[2]+beta[3]/5)^2),
  0
)


var_theta <- t(gradient_g) %*% V_robust %*% gradient_g
theta_se <- sqrt(var_theta)
cat("Standard Error of Theta hat: ", theta_se, "\n")
```

```
Standard Error of Theta hat:  0.2267341
```

Part D

```
z_90 <- qnorm(0.95)
ci_90_low <- theta_hat - z_90*theta_se
ci_90_hi <- theta_hat + z_90*theta_se
cat("\nPart D: 90% Confidence Interval for theta:\n")
```

```
Part D: 90% Confidence Interval for theta:
```

```
cat("[", ci_90_low, ", ", ci_90_hi, "]\n")
```

```
[ 3.095391 ,  3.84128 ]
```

Part E

```
x_0 <- c(12,20,(20^2/100),1)
y_hat_0 <- sum(x_0*beta)

var_y_hat_0 <- t(x_0) %*% V_robust %*% x_0
se_y_hat_0 <- sqrt(var_y_hat_0)

z_95 <- qnorm(0.975)
ci_95_low <- y_hat_0 - z_95*se_y_hat_0
ci_95_hi <- y_hat_0 + z_95*se_y_hat_0

cat("\nPart E: Regression at educ = 12, exper = 20\n")
```

```
Part E: Regression at educ = 12, exper = 20
```

```
cat("Predicted log(wage): ", y_hat_0, "\n")
```

```
Predicted log(wage):  2.792167
```

```
cat("95% CI for regression function: [", ci_95_low, ", ", ci_95_hi, "]\n")
```

```
95% CI for regression function: [ 2.7693 ,  2.815034 ]
```

EQ 1

```
library(haven)
library(dplyr)
```

```
rm(list = ls())

dat <- read_dta('~/SchoolWork/Sem2/Metrics/PSets/PS3/cps09mar.dta')
sample <- (dat[,11]==1)&(dat[,2]==0)&(dat[,3]==1)
df <- dat[sample,]
```

```r
y <- as.matrix(log(df[,5]/(df[,6]*df[,7])))
exp <- df[,1]-df[,4]-6
exp2 <- (exp^2)/100

x_df <- data.frame(
  education = df[,4],
  experience = exp,
  exp_squared = exp2,
  intercept = 1
)
x <- as.matrix(x_df)

xx <- t(x)%*%x
xy <- t(x)%*%y
beta <- solve(xx,xy)
```

```r
set.seed(0528)
bootstrap_ols <- function(indices) {
  x_boot <- x[indices, ]
  y_boot <- y[indices]

  xx_boot <- t(x_boot)%*%x_boot
  xy_boot <- t(x_boot)%*%y_boot
  beta_boot <- solve(xx_boot,xy_boot)

  return(beta_boot)
}
```

```r
B <- 1000
boot_res <- matrix(0, nrow=B, ncol=length(beta))

for (b in 1:B) {
  indices <- sample(1:nrow(x), nrow(x), replace=TRUE)
  boot_res[b, ] <- bootstrap_ols(indices)
}
```

```r
boot_se <- apply(boot_res, 2, sd)

alpha <- 0.05
bootstrap_ci <- matrix(0, nrow=length(beta), ncol=2)
for (j in 1:length(beta)) {
```

```
  bootstrap_ci[j, ] <- quantile(boot_res[, j], c(alpha/2, 1-alpha/2))
}
```

```
cat("Part A: Coefficient Estimates with Bootstrap Standard Errors: \n")
```

```
Part A: Coefficient Estimates with Bootstrap Standard Errors:
```

```
coef_tab <- cbind(beta, boot_se)
colnames(coef_tab) <- c("Coefficieent", "Bootstrap SE")
rownames(coef_tab) <- c("Education", "Experience", "Experience^2", "Intercept")
print(coef_tab)
```

```
             Coefficieent Bootstrap SE
Education      0.09044896  0.002830480
Experience     0.03537968  0.002615053
Experience^2  -0.04650594  0.005306071
Intercept      1.18520948  0.044950260
```

The results show the same coefficients as when calculated with robust standard errors. However, the standard errors differ very slightly. For all but experience, the bootstrap standard error is larger than the robust.

EQ 2

```
b0 <- 0
b1 <- 1
n <- 100
sim <- function() {
  X1 <- rexp(n)
  e <- mixtools::rnormmix(n,lambda=c(0.5,0.5),mu=c(-1,2),sigma=c(1,1))
  Y <- b0 + b1*X1 + e
  x <- cbind(1,X1)
  xx <- t(x)%*%x
  xy <- t(x)%*%Y
  bhat <- solve(xx,xy)
  return(c(bhat[2]))
}
```

These results show that the average of $\hat{\beta}_1 \to \beta_1$ as n grows. The variance also approaches 0. This is consistent with what was derived in class, that as $n \to \infty$, we see the predicted approach the actual, and variance should be 0 as with sufficiently large n, there will be no variance in observations.

```r
run_mc <- function(n_sims = 1000) {
  mc_res <- sapply(1:n_sims, function(s) {
    sim()
  })
  cat("Mean b1:", mean(mc_res), "\n")
  cat("Variance of b1:", var(mc_res), "\n")
}
```

```r
run_mc()
```

```
Mean b1: 1.002798
Variance of b1: 0.03317455
```

```r
n <- 2
run_mc()
```

```
Mean b1: -13.39236
Variance of b1: 1134519
```

```r
n <- 10
run_mc()
```

```
Mean b1: 1.009071
Variance of b1: 0.6886915
```

```r
n <- 50
run_mc()
```

```
Mean b1: 0.995345
Variance of b1: 0.07296549
```

```r
n <- 500
run_mc()
```

```
Mean b1: 0.9953144
Variance of b1: 0.006904567
```

As $n \to \infty$, the mean and variance get closer to the true values. This is a showcase of the WLLN.

EQ 3

Part A

```
b0 <- 0
b1 <- 1
num_sims <- 1000
alpha <- 0.05

sim_test <- function(n,b1_true, b1_null) {
  X <- rexp(n)
  e <- mixtools::rnormmix(n,lambda=c(0.5,0.5),mu=c(-2,2),sigma=c(1,1))
  Y <- b0+b1*X + e
  x <- cbind(1,X)
  xx <- t(x)%*%x
  xy <- t(x)%*%Y
  bhat <- solve(xx,xy)
  b0_hat <- bhat[1]
  b1_hat <- bhat[2]

  yhat <- x %*% bhat
  ehat <- Y - yhat

  sigma_sq_hat <- sum(ehat^2)/(n-2)
  var_cov_matrix <- as.numeric(sigma_sq_hat)*solve(xx)
  se_b1_hat <- sqrt(var_cov_matrix[2,2])
  t_stat <- (b1_hat - b1_null)/se_b1_hat

  df <- n-2
  t_crit <- qt(1-alpha/2,df)
  reject <- abs(t_stat) > t_crit
  p_value <- 2*pt(abs(t_stat), df=df, lower.tail=FALSE)

  return(list(
```

```
    b0_hat = b0_hat,
    b1_hat = b1_hat,
    se_b1_hat = se_b1_hat,
    t_stat = t_stat,
    t_crit = t_crit,
    p_value = p_value,
    reject = reject
  ))
}
```

```
run_hypothesis_test <- function(n, b1_true, b1_null) {
  b1_hats <- numeric(num_sims)
  se_b1_hats <- numeric(num_sims)
  t_stats <- numeric(num_sims)
  p_values <- numeric(num_sims)
  rejects <- logical(num_sims)

  for (i in 1:num_sims) {
    sim_result <- sim_test(n,b1_true,b1_null)
    b1_hats[i] <- sim_result$b1_hat
    se_b1_hats[i] <- sim_result$se_b1_hat
    t_stats[i] <- sim_result$t_stat
    p_values[i] <- sim_result$p_value
    rejects[i] <- sim_result$reject
  }

  reject_rate <- mean(rejects)
  mean_b1_hat <- mean(b1_hats)
  var_b1_hat <- var(b1_hats)
  mean_se_b1_hat <- mean(se_b1_hats)

  return(list(
    b1_hats = b1_hats,
    se_b1_hats = se_b1_hats,
    t_stats = t_stats,
    p_values = p_values,
    rejects = rejects,
    mean_b1_hat = mean_b1_hat,
    var_b1_hat = var_b1_hat,
    mean_se_b1_hat = mean_se_b1_hat,
    reject_rate = reject_rate
  ))
```

```r
}
```

```r
set.seed(0528)
results_100_true <- run_hypothesis_test(n=100,b1_true=1,b1_null=1)
cat("Part a & b: Results for n = 100, H : b1 = 1 (true value)\n")
```

```
Part a & b: Results for n = 100, H : b1 = 1 (true value)
```

```r
cat("Theoretical rejection rate at alpha = 0.05 should be: 0.05\n")
```

```
Theoretical rejection rate at alpha = 0.05 should be: 0.05
```

```r
cat("Observed rejection rate:", results_100_true$reject_rate, "\n")
```

```
Observed rejection rate: 0.047
```

```r
cat("Mean b1_hat:", results_100_true$mean_b1_hat, "\n")
```

```
Mean b1_hat: 1.008178
```

```r
cat("Variance of b1_hat:", results_100_true$var_b1_hat, "\n")
```

```
Variance of b1_hat: 0.05378357
```

```r
cat("Mean standard error of b1_hat:", results_100_true$mean_se_b1_hat, "\n")
```

```
Mean standard error of b1_hat: 0.2301119
```

```r
cat("Theoretical variance (from SE):", results_100_true$theoretical_var, "\n\n")
```

```
Theoretical variance (from SE):
```

Part C

```r
sample_size <- c(10,50,500,1000)
results_varying_n <- list()

for (n in sample_size) {
  results_varying_n[[paste0("n", n)]] <- run_hypothesis_test(n=n, b1_true=1, b1_null=1)
  cat("Results for n =", n, ", H : b1 = 1 (true value)\n")
  cat("Rejection rate:", results_varying_n[[paste0("n", n)]]$reject_rate, "\n")
  cat("Mean b1_hat:", results_varying_n[[paste0("n", n)]]$mean_b1_hat, "\n")
  cat("Variance of b1_hat:", results_varying_n[[paste0("n", n)]]$var_b1_hat, "\n")
  cat("Mean standard error of b1_hat:", results_varying_n[[paste0("n", n)]]$mean_se_b1_hat, "
  cat("Theoretical variance (from SE):", results_varying_n[[paste0("n", n)]]$theortical_var,
}
```

```
Results for n = 10 , H : b1 = 1 (true value)
Rejection rate: 0.045
Mean b1_hat: 1.006626
Variance of b1_hat: 1.021049
Mean standard error of b1_hat: 0.9657625
Theoretical variance (from SE):

Results for n = 50 , H : b1 = 1 (true value)
Rejection rate: 0.043
Mean b1_hat: 0.9931654
Variance of b1_hat: 0.1208395
Mean standard error of b1_hat: 0.3389943
Theoretical variance (from SE):

Results for n = 500 , H : b1 = 1 (true value)
Rejection rate: 0.054
Mean b1_hat: 0.9964828
Variance of b1_hat: 0.01071093
Mean standard error of b1_hat: 0.1009742
Theoretical variance (from SE):

Results for n = 1000 , H : b1 = 1 (true value)
Rejection rate: 0.06
Mean b1_hat: 1.002231
Variance of b1_hat: 0.005243213
Mean standard error of b1_hat: 0.07109302
Theoretical variance (from SE):
```

Part D

```r
results_100_false <- run_hypothesis_test(n=100,b1_true=1,b1_null=0)
cat("Part d: Results for n = 100, H :    = 0 (false null)\n")
```

```
Part d: Results for n = 100, H :    = 0 (false null)
```

```r
cat("Rejection rate (power):", results_100_false$reject_rate, "\n")
```

```
Rejection rate (power): 0.978
```

```r
cat("Mean ˆ :", results_100_false$mean_b1_hat, "\n")
```

```
Mean ˆ : 0.9998289
```

```r
cat("Variance of ˆ :", results_100_false$var_b1_hat, "\n\n")
```

```
Variance of ˆ : 0.0556058
```

```r
results_varying_n_false <- list()

for (n in sample_size) {
  set.seed(123)
  results_varying_n_false[[paste0("n", n)]] <- run_hypothesis_test(n = n, b1_true = 1, b1_nul

  cat("Results for n =", n, ", H : b1 = 0 (false null)\n")
  cat("Rejection rate (power):", results_varying_n_false[[paste0("n", n)]]$reject_rate, "\n"
  cat("Mean b1_hat:", results_varying_n_false[[paste0("n", n)]]$mean_b1_hat, "\n")
  cat("Variance of b1_hat:", results_varying_n_false[[paste0("n", n)]]$var_b1_hat, "\n")
}
```

```
Results for n = 10 , H : b1 = 0 (false null)
Rejection rate (power): 0.199
Mean b1_hat: 1.017303
Variance of b1_hat: 1.029224
Results for n = 50 , H : b1 = 0 (false null)
Rejection rate (power): 0.819
Mean b1_hat: 1.009133
```

```
Variance of b1_hat: 0.1264878
Results for n = 500 , H : b1 = 0 (false null)
Rejection rate (power): 1
Mean b1_hat: 1.004012
Variance of b1_hat: 0.009162103
Results for n = 1000 , H : b1 = 0 (false null)
Rejection rate (power): 1
Mean b1_hat: 0.997987
Variance of b1_hat: 0.005327341
```