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
```

```
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

```r
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

```r
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

```r
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:
```

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

```
[ 3.095391 ,  3.84128 ]
```

Part E

```r
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
```

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

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

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

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

EQ 1

EQ 2

```r
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
```

```r
  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: 1.005084
Variance of b1: 0.0354155
```

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

```
Mean b1: 0.768838
Variance of b1: 2020.169
```

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

```
Mean b1: 1.005915
Variance of b1: 0.7500934
```

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

```
Mean b1: 0.9881707
Variance of b1: 0.07680855
```

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

```
Mean b1: 1.001566
Variance of b1: 0.006245428
```

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

```r
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_val <- 2*pt(abs(t_stat), df=df, lower.tail=FALSE)

  return(list(
    b1_hat = b1_hat,
    se_b1_hat = se_b1_hat,
    t_stat = t_stat,
    t_crit = t_crit,
    p_val = p_val,
    reject = reject
  ))
}
```

```
run_hypothesis_test <- function(n, b1_true, b1_null) {
  results <- data.frame(
    b1_hat = numeric(num_sims),
    se_b1_hat = numeric(num_sims),
    t_stat = numeric(num_sims),
    p_val = numeric(num_sims),
    reject = logical(num_sims)
  )

  for (i in 1:num_sims) {
    sim_result <- sim_test(n,b1_true,b1_null)
    results$b1_hat[i] <- sim_result$b1_hat
    results$se_b1_hat[i] <- sim_result$se_b1_hat
    results$t_stat[i] <- sim_result$t_stat
    results$p_val[i] <- sim_result$p_val
    results$reject[i] <- sim_result$reject
  }

  reject_rate <- mean(results$reject)
  mean_b1_hat <- mean(results$b1_hat)
  var_b1_hat <- var(results$b1_hat)
  mean_se_b1_hat <- mean(results$se_b1_hat)
```

```r
  theoretical_var <- mean(results$se_b1_hat^2)

  return(list(
    results = results,
    reject_rate = reject_rate,
    mean_b1_hat = mean_b1_hat,
    var_b1_hat = var_b1_hat,
    mean_se_b1_hat = mean_se_b1_hat,
    theoretical_var = theoretical_var
  ))
}
```

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

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

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

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

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

```
Observed rejection rate:
```

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

```
Mean ^ : 1.000734
```

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

```
Variance of ^ : 0.05760872
```

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

```
Mean standard error of ^ : 0.2301102
```

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

```
Theoretical variance (from SE): 0.05409141
```

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 :   = 1 (true value)\n")
  cat("Rejection rate:", results_varying_n[[paste0("n", n)]]$rejection_rate, "\n")
  cat("Mean ^ :", results_varying_n[[paste0("n", n)]]$mean_b1_hat, "\n")
  cat("Variance of ^ :", results_varying_n[[paste0("n", n)]]$var_b1_hat, "\n")
  cat("Mean standard error of ^ :", results_varying_n[[paste0("n", n)]]$mean_se_b1_hat, "\n")
  cat("Theoretical variance (from SE):", results_varying_n[[paste0("n", n)]]$theortical_var,
}
```

```
Results for n = 10 , H :   = 1 (true value)
Rejection rate:
Mean ^ : 1.019893
Variance of ^ : 1.068878
Mean standard error of ^ : 0.9230988
Theoretical variance (from SE):

Results for n = 50 , H :   = 1 (true value)
Rejection rate:
Mean ^ : 1.012697
Variance of ^ : 0.1091571
Mean standard error of ^ : 0.3362441
Theoretical variance (from SE):

Results for n = 500 , H :   = 1 (true value)
Rejection rate:
Mean ^ : 1.001802
Variance of ^ : 0.0102878
Mean standard error of ^ : 0.1007871
Theoretical variance (from SE):
```

```
Results for n = 1000 , H :   = 1 (true value)
Rejection rate:
Mean ^ : 0.9981505
Variance of ^ : 0.00507883
Mean standard error of ^ : 0.07095791
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$rejection_rate, "\n")
```

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

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

```
Mean ^ : 0.9960999
```

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

```
Variance of ^ : 0.0544113
```

```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 :   = 0 (false null)\n")
  cat("Rejection rate (power):", results_varying_n_false[[paste0("n", n)]]$rejection_rate, "
  cat("Mean ^ :", results_varying_n_false[[paste0("n", n)]]$mean_b1_hat, "\n")
  cat("Variance of ^ :", results_varying_n_false[[paste0("n", n)]]$var_b1_hat, "\n")
}
```

```
Results for n = 10 , H :   = 0 (false null)
Rejection rate (power):
Mean ˆ : 1.017303
Variance of ˆ : 1.029224
Results for n = 50 , H :   = 0 (false null)
Rejection rate (power):
Mean ˆ : 1.009133
Variance of ˆ : 0.1264878
Results for n = 500 , H :   = 0 (false null)
Rejection rate (power):
Mean ˆ : 1.004012
Variance of ˆ : 0.009162103
Results for n = 1000 , H :   = 0 (false null)
Rejection rate (power):
Mean ˆ : 0.997987
Variance of ˆ : 0.005327341
```