Lecture 6 Exercises

Isabel Fulcher

8/14/2018

Install packages

```
library(matrixStats)
library(knitr)
library(tidyverse)
library(reshape2)
```

Population Mean Example

Suppose you are interested in comparing the properties of the following 3 estimators for the maen μ for n iid draws $X_1, ..., X_n$ with $X_i \sim f(x)$

- Sample mean, T^1
- Sample 15% trimmed mean mean, T^2
- Sample median, T^3

How would you expect the estimators to compare if the distribution of X_i is N(1,16)?

Step 1: Conduct the simulation

```
#Set the seed
set.seed(123456)

#Set up simulation parameters and truth
B = 500 #number of replicates
true.mu = 1 #true population mean
samp.size = 100 #sample size

#Create a function to loop or apply over
simulate <- function(n,mu,b){
  #generate data
  samp <- rnorm(n, mean=true.mu, sd=4)

  #calculate relevant quantities
  mean <- mean(samp)
  mean_trim <- mean(samp,trim=0.15)
  med <- median(samp)

  #return results
  return(c(mean,mean_trim,med))
}

#Simulate 500 times
# Option 1: use sapply
```

```r
out.sapply <- sapply(1:B,simulate,n=samp.size,mu=true.mu) #this returns a 3xB matrix

# Option 2: use a for loop
out.for <- matrix(NA,B,3) #this will store results in a Bx3 matrix
for (b in 1:B){
  out.for[b,] <- simulate(n=samp.size,mu=true.mu,b)
}
```

Step 2: Calculate the simulation quantities for each estimator

```r
#OPTION 1: BASE R
# Mean
sim.mean <- colMeans(out.for)

# Bias
sim.bias <- colMeans(out.for-true.mu)

# Relative bias
sim.rel.bias <- colMeans(out.for-true.mu)/true.mu

# Standard deviation
sim.sd <- colSds(out.for)

# Mean squared error
sim.mse <- sim.bias^2 + sim.sd^2 #bias^2 + variance

# Combine all together
df.results <- data.frame(rbind(sim.mean,sim.bias,sim.rel.bias,sim.sd,sim.mse))

#OPTION 2: TIDYVERSE
df.out <- data.frame(out.for)
df.out %<>% rename(mean=X1,`trimmed mean`=X2,median=X3) %>%
  melt() %>% group_by(variable) %>%
  summarise(sim.mean=mean(value),sim.bias=mean(value)-true.mu,
            sim.rel.bias=(mean(value)-true.mu)/true.mu,
            sim.sd = sd(value),
            sim.mse = (mean(value)-true.mu)^2 + sd(value)^2) #one command!
```

```
## No id variables; using all as measure variables
```

Step 3: Present your results

```r
kable(df.results,digits=3,align=rep('c', 2),
      col.names=c("mean","trimmed mean","median"))
```

	mean	trimmed mean	median
sim.mean	1.016	1.022	1.034
sim.bias	0.016	0.022	0.034
sim.rel.bias	0.016	0.022	0.034
sim.sd	0.404	0.422	0.508

	mean	trimmed mean	median
sim.mse	0.164	0.178	0.260

Simulation exercise

```
library(MASS)
```

What happens if I exclude a covariate from my model? This follows from the first question on slide 6,

$$E[Y|X_1, X_2] = \beta_0 + \beta_1 X_1 + \beta_2 X_2$$

$$E[Y|X_1] = \alpha_0 + \alpha_1 X_1$$

The goal of this exercise is to say when $\hat{\alpha_1}$ is unbiased for β_1.

1. Write a function that takes in b, n, Σ, λ, β_0, β_1, and β_2 and performs the following analysis:

- Generate an $n \times 2$ matrix containing the predictors X_1 and X_2 from a $MVN(0_{2\times 1}, \Sigma_{2\times 2})$ where $\Sigma_{2\times 2}$ is the covariance matrix (the MASS package has a function called mvrnorm)
- Generate an outcome vector with n observations $\mathbf{y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \boldsymbol{\epsilon}$ where $\epsilon_i \sim N(0, \lambda^2)$ and X_1 and X_2 come from above
- Fit the unadjusted model $E[Y \mid X_1] = \alpha_0 + \alpha_1 X_1$
- Return the coefficient estimates from the unadjusted model, i.e. $\hat{\alpha}_0$ and $\hat{\alpha}_1$

```
gen.func <- function(b,n,Sigma,lambda,b0,b1,b2){
  #only need b if you are going to use sapply in part 2

  beta <- c(b0,b1,b2)
  X <- cbind(1,MASS::mvrnorm(n,rep(0,2),Sigma))
  Y <- X%*%beta + rnorm(n,0,lambda)

  fit <- lm(Y ~ X[,2])
  beta.hat <- coefficients(fit)

  return(beta.hat)
}
```

2. Use your function to repeat the above analysis $B = 1000$ times with $n = 500$, $\lambda = 1$, $\beta_0 = 2$, $\beta_1 = 4$, for the four scenarios:

- Scenario 1: $\beta_2 = 2$ and
$$\Sigma = \begin{bmatrix} 2 & 0.3 \\ 0.3 & 2 \end{bmatrix}$$
- Scenario 2: $\beta_2 = 0$ and
$$\Sigma = \begin{bmatrix} 2 & 0.3 \\ 0.3 & 2 \end{bmatrix}$$
- Scenario 3: $\beta_2 = 2$ and
$$\Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

- Scenario 4: $\beta_2 = 0$ and
$$\Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

```
#Parameter values
B=1000
n=500
lambda=1
beta0=2
beta1=4

#Scenario 1
Sigma1 = matrix(c(2,.3,.3,2),2,2)
beta2= 2
scenario1 <- sapply(1:B,gen.func,n=n,Sigma=Sigma1,lambda=lambda,
                    b0=beta0,b1=beta1,b2=beta2)

#Scenario 2
beta2= 0
scenario2 <- sapply(1:B,gen.func,n=n,Sigma=Sigma1,lambda=lambda,
                    b0=beta0,b1=beta1,b2=beta2)

#Scenario 3
Sigma2 = matrix(c(2,0,0,2),2,2)
beta2= 2
scenario3 <- sapply(1:B,gen.func,n=n,Sigma=Sigma2,lambda=lambda,
                    b0=beta0,b1=beta1,b2=beta2)

#Scenario 4
beta2= 0
scenario4 <- sapply(1:B,gen.func,n=n,Sigma=Sigma2,lambda=lambda,
                    b0=beta0,b1=beta1,b2=beta2)
```

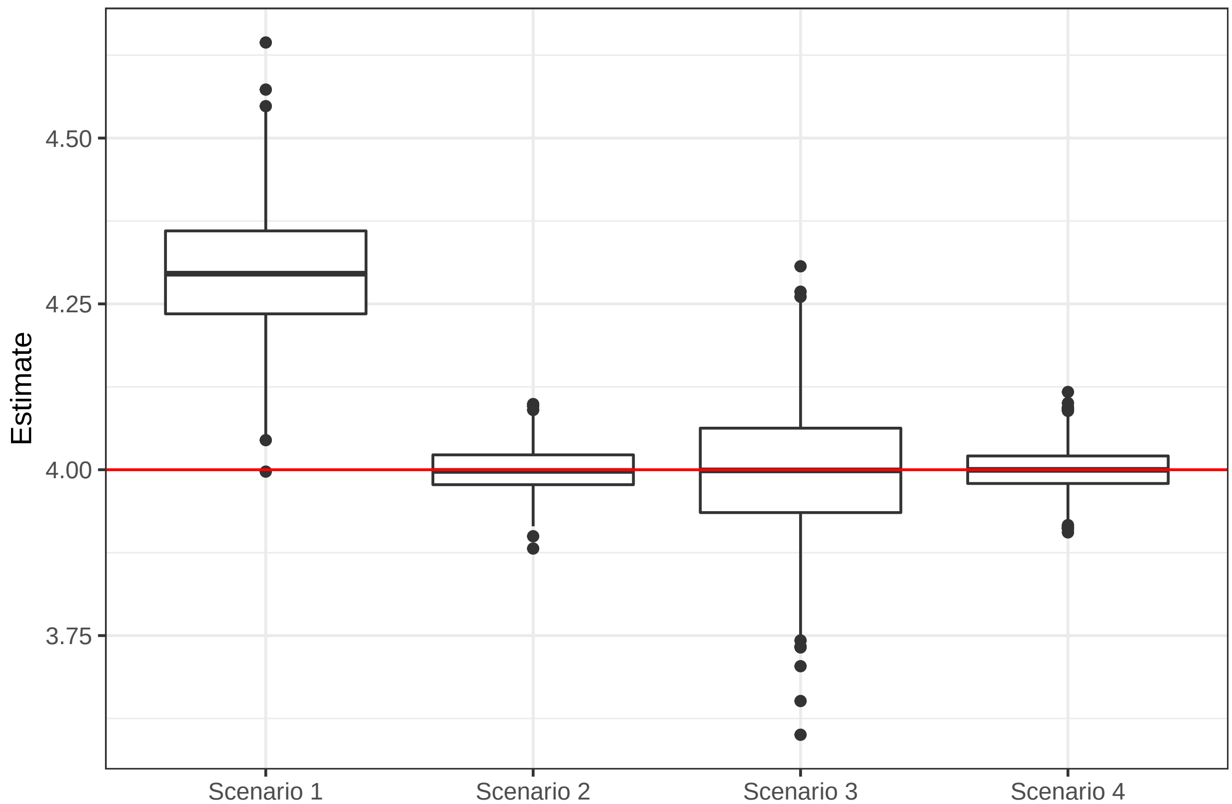
3. For all scenarios, compute the bias of the coefficient estimates of α_0 and α_1. Create a table with these results (columns should be scenarios).

```
#Calculate bias
truth = c(beta0,beta1)
bias1 <- rowMeans(scenario1)-truth
bias2 <- rowMeans(scenario2)-truth
bias3 <- rowMeans(scenario3)-truth
bias4 <- rowMeans(scenario4)-truth

#Construct dataframe for table
table <- cbind(t(t(bias1)),t(t(bias2)),t(t(bias3)),t(t(bias4)))
colnames(table) <- paste("Scenario",1:4)
row.names(table) <- c("alpha0","alpha1")

#Print table
kable(table,digits=3,align=rep('c', 4))
```

	Scenario 1	Scenario 2	Scenario 3	Scenario 4
alpha0	-0.004	0.000	0.001	0.002
alpha1	0.297	-0.001	-0.002	0.000

4. Using a boxplot, plot the coefficient estimates for α_1 for each scenario. Indicate the true value of β_2 on the plot.

```
#Reformat dataframe
alpha1 <- cbind(scenario1[2,],scenario2[2,],scenario3[2,],scenario4[2,])
colnames(alpha1) <- paste("Scenario",1:4)
alpha1 %>% melt() -> alpha1

#Boxplot
ggplot(alpha1,aes(x=Var2,y=value)) + geom_boxplot() +
  geom_hline(yintercept=truth[2],col="red") + theme_bw() +
  labs(y="Estimate",x="")
```



5. Under which scenarios is $\hat{\alpha}_1$ unbiased for β_1? Any other observations? $\hat{\alpha}_1$ seems to be unbiased for β_1 in Scenarios 2,3, and 4 as the bias is all close to zero. In other words, $\hat{\alpha}_1$ is biased for β_1 when X_2 is correlated with X_1 and X_2 has an effect on the outcome (i.e. $\beta_2 \neq 0$). In the boxplot, it seems that the standard deviation of $\hat{\alpha}_1$ in Scenario 3 is greater than the other unbiased scenarios (2 and 4).