Homework 1

Hugo Fluhr

Spring 2024

Note: In all distribution plots the black curve is the correct estimate and the red one the incorrect estimate (hopefully).

Problem 1

DAG

```
dag1 <- dagitty('dag {
    E [pos="0,1"]
    WL [pos="1,1"]
    CI [pos="1,0"]
    M [pos="0,0"]
    E -> WL
    E->CI->WL
    E<-M->CI
}')
plot(dag1)
```

Identifying the variables to control for to estimate the direct effect of E on WL

```
adjustmentSets(dag1, exposure='E', outcome='WL', effect = 'direct')
```

```
## { CI }
```

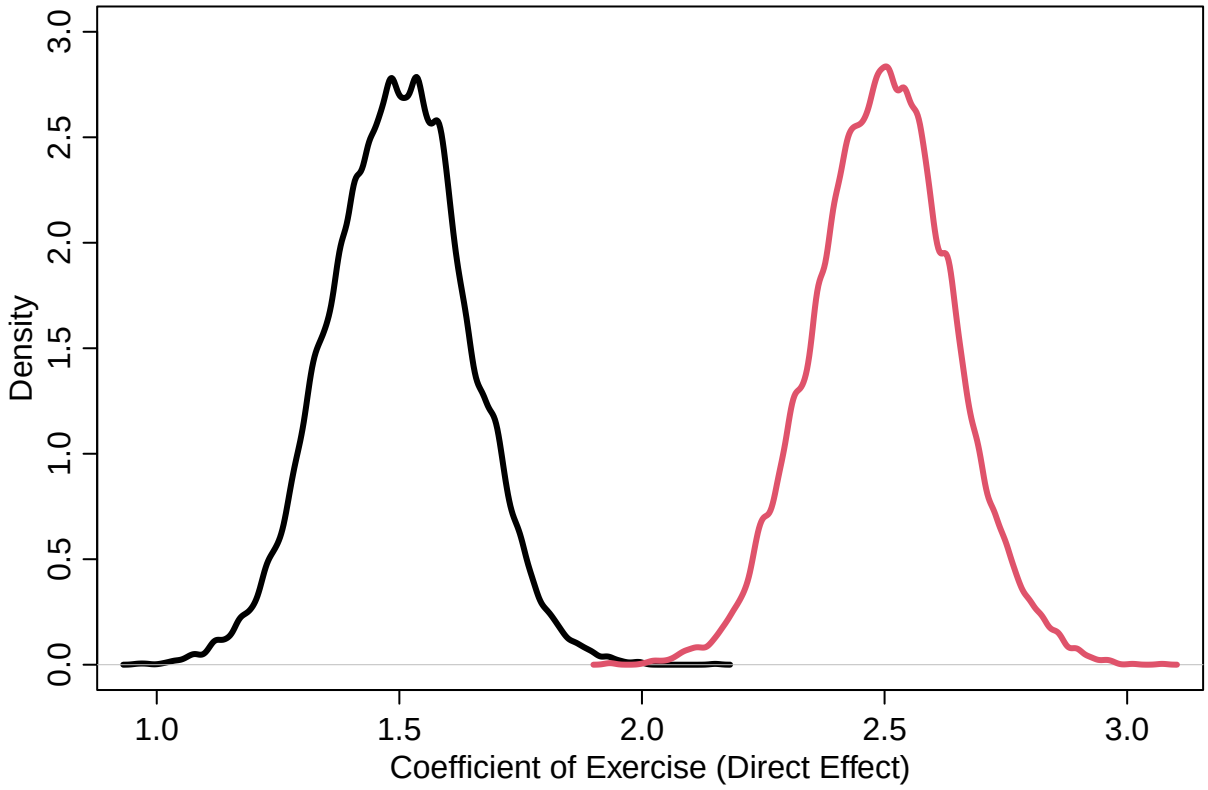
We need to adjust for CI to estimate the direct effect of E on WL, this close the path E -> CI -> WL. An incorrect model would be to adjust for M instead, this would give the total effect of E on WL.

Simulation

```
# Function to simulate data and fit models
f1 <- function(n=100, bM_E=1, bM_CI=1, bE_CI=1, bCI_WL=1, bE_WL=1.5) {
  # Simulate data
  M <- rnorm(n)  # Motivation
  E <- rnorm(n, bM_E*M) # Regular Exercise
  CI <- rnorm(n, bM_CI*M + bE_CI*E) # Caloric Intake
  WL <- rnorm(n, bE_WL*E + bCI_WL*CI ) # Weight loss

  # Fit models
  b_direct <- coef(lm(WL ~ E + CI))['E']  # Model controlling for CI, direct effect
  b_incorrect <- coef(lm(WL ~ E + M))['E']  # Model controlling for M, total effect

  return(c(b_direct, b_incorrect))  # Return coefficients
}
# Perform Monte Carlo simulation
sim1 <- mcreplicate(1e4, f1(), mc.cores = 8)
```

Plotting the correct and incorrect versions of the estimate

```
# Plot posterior distributions
range1 <- range(sim1[1,])
range2 <- range(sim1[2,])
xlim <- range(c(range1, range2))
ylim <- c(0,3.)

dens(sim1[1,], lwd=3, xlab='Coefficient of Exercise (Direct Effect)', xlim=xlim, ylim=ylim)
dens(sim1[2,], lwd=3, col=2, add=TRUE, xlim=xlim, ylim=ylim)
```



Problem 2: Education Level and Job Satisfaction

DAG

```
dag2 <- dagitty('dag {
    EL [pos="0,1"]
    JS [pos="1,1"]
    F [pos="1,0"]
    PE [pos="0,0"]
    EL -> JS
    EL -> F -> JS
    EL <- PE -> JS
}')
plot(dag2)
```

Identifying the variables to control for to estimate the total effect of EL on JS

```
adjustmentSets(dag2, exposure='EL', outcome='JS', effect = 'total')
```

```
## { PE }
```

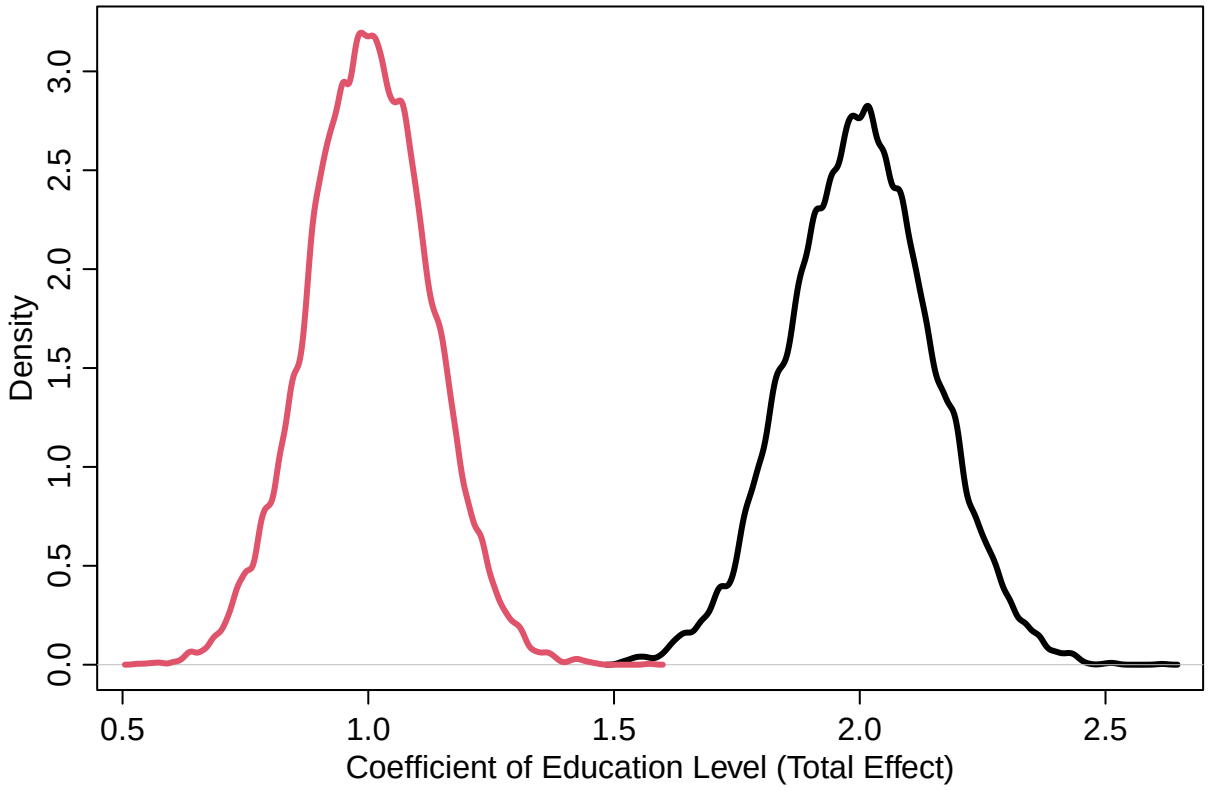
We need to adjust for both PE to estimate the total effect of EL on JS, this closes a backdoor that would bias our estimate. The path through F is part of the total effect of EL on JS.

Simulation

```
# Function to simulate data and fit models
f2 <- function(n=100, bPE_EL=1, bPE_JS=1, bEL_F=1, bF_JS=1, bEL_JS=1) {
  # Simulate data
  PE <- rnorm(n)  # Prior Experience
  EL <- rnorm(n, bPE_EL*PE ) # Education level
  F <- rnorm(n, bEL_F*EL ) # Field
  JS <- rnorm(n, bF_JS*F + bEL_JS*EL ) # Job Satisfaction

  # Fit models
  b_total <- coef(lm(JS ~ EL + PE))['EL']  # Model controlling for PE (correct)
  b_incorrect <- coef(lm(JS ~ EL + F))['EL']  # Model controlling for F (incorrect)

  return(c(b_total, b_incorrect))  # Return coefficients
}
# Perform Monte Carlo simulation
sim2 <- mcreplicate(1e4, f2(), mc.cores = 8)
```

Plotting the correct and incorrect versions of the estimate

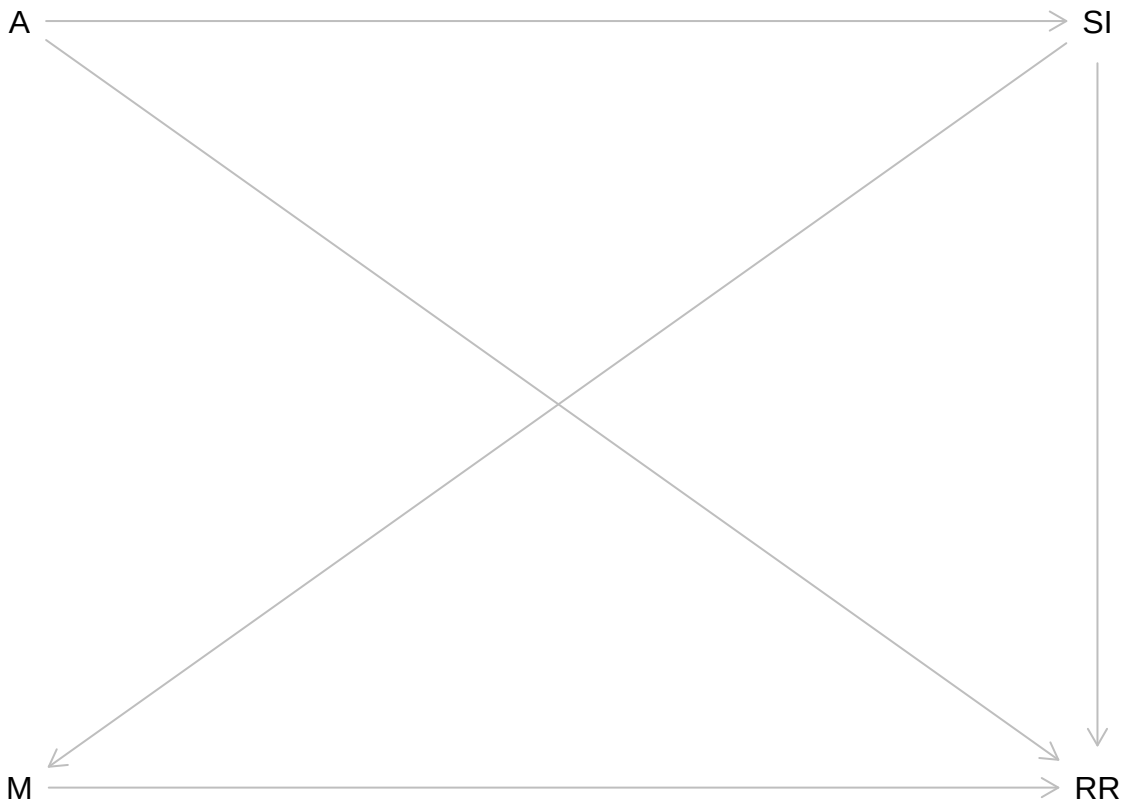
```
# Plot posterior distributions
range1 <- range(sim2[1,])
range2 <- range(sim2[2,])
xlim <- range(c(range1, range2))
ylim <- c(0,3.2)

dens(sim2[1,], lwd=3, xlab='Coefficient of Education Level (Total Effect)', xlim=xlim, ylim=ylim)
dens(sim2[2,], lwd=3, col=2, add=TRUE, xlim=xlim, ylim=ylim)
```



Problem 3: Medication Use and Patient Recovery Rate

DAG

```
dag3 <- dagitty('dag {
    M [pos="0,1"]
    RR [pos="1,1"]
    SI [pos="1,0"]
    A [pos="0,0"]
    M -> RR
    M <- SI -> RR
    SI <- A -> RR
}')
plot(dag3)
```



Identifying the variables to control for to estimate the direct effect of M on RR

```
adjustmentSets(dag3, exposure='M', outcome='RR', effect = 'direct')
```

```
## { SI }
```

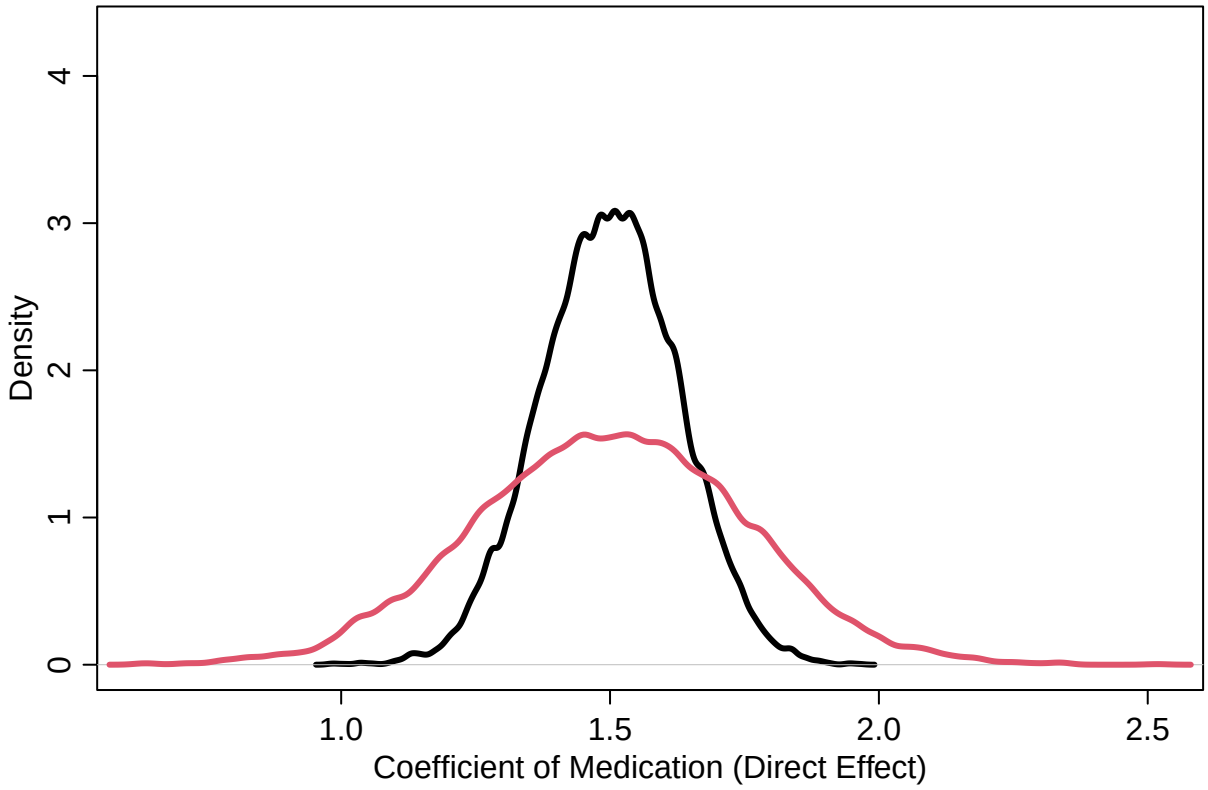
We need to adjust for SI to estimate the direct effect of M on RR as it closes a backdoor through SI. We do not need to control for A as it is independent of M.

Simulation

```
# Function to simulate data and fit models
f3 <- function(n=100, bA_SI=1, bA_RR=1, bSI_M=1, bSI_RR=1, bM_RR=1.5) {
  # Simulate data
  A <- rnorm(n)  # Age
  SI <- rnorm(n, bA_SI*A)
  M <- rnorm(n, bSI_M)
  RR <- rnorm(n, bA_RR*A + bSI_RR*SI + bM_RR*M )

  # Fit models
  b_direct <- coef(lm(RR ~ M + SI))['M']  # Model controlling for SI
  b_incorrect <- coef(lm(RR ~ M))['M']  # Model not controlling for SI

  return(c(b_direct, b_incorrect))  # Return coefficients
}
# Perform Monte Carlo simulation
sim3 <- mcreplicate(1e4, f3(), mc.cores = 8)
```

Plotting the correct and incorrect versions of the estimate

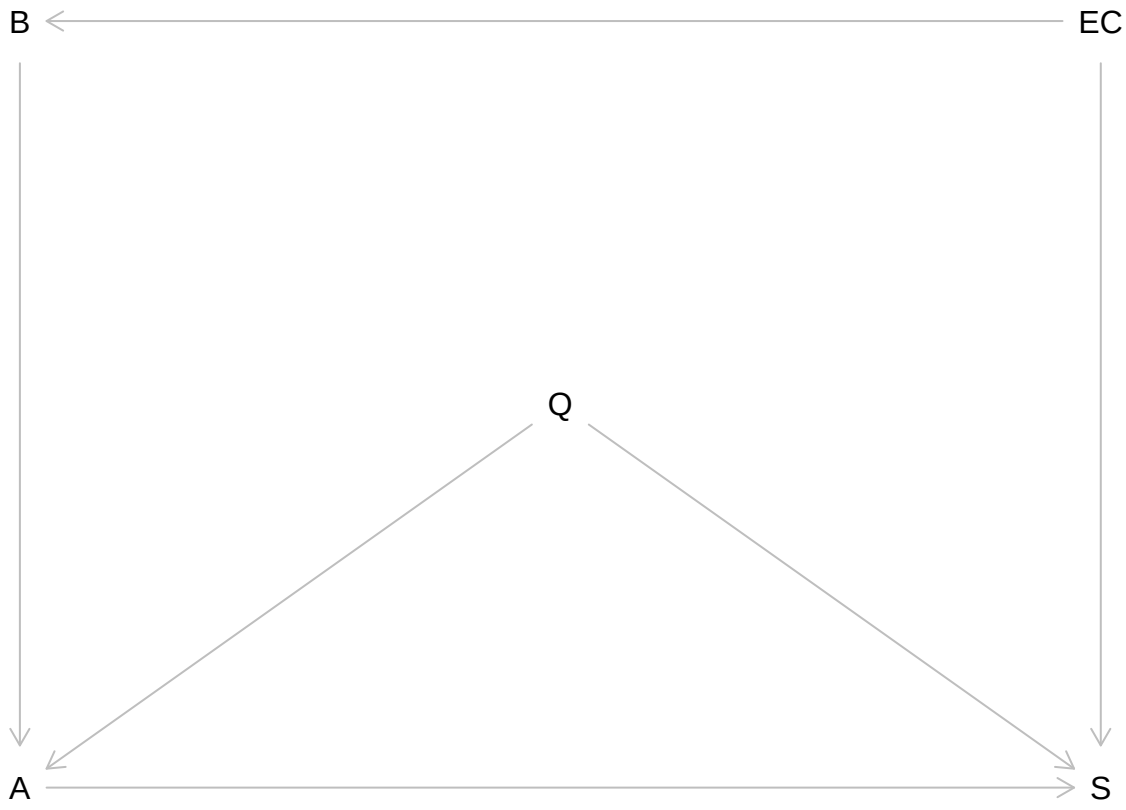
```
# Plot posterior distributions
range1 <- range(sim3[1,])
range2 <- range(sim3[2,])
xlim <- range(c(range1, range2))
ylim <- c(0,4.3)

dens(sim3[1,], lwd=3, xlab='Coefficient of Medication (Direct Effect)', xlim=xlim, ylim=ylim)
dens(sim3[2,], lwd=3, col=2, add=TRUE, xlim=xlim, ylim=ylim)
```



Problem 4: Advertising and Sales

DAG

```
dag4 <- dagitty('dag {
    A [pos="0,1"]
    S [pos="1,1"]
    EC [pos="1,0"]
    B [pos="0,0"]
    Q [pos="0.5,0.5"]
    A -> S
    B -> A
    B <- EC -> S
    S <- Q -> A
}')
plot(dag4)
```



Identifying the variables to control for to estimate the total effect of Advertising on Sales

```
adjustmentSets(dag4, exposure='A', outcome='S', effect = 'total')
```

```
## { EC, Q }
## { B, Q }
```

We need to either adjust for both EC and Q or adjust for EC and B to estimate the total effect of advertising on sales. These adjustments close the backdoors through the path A<-B<-EC->S and the one through Q.

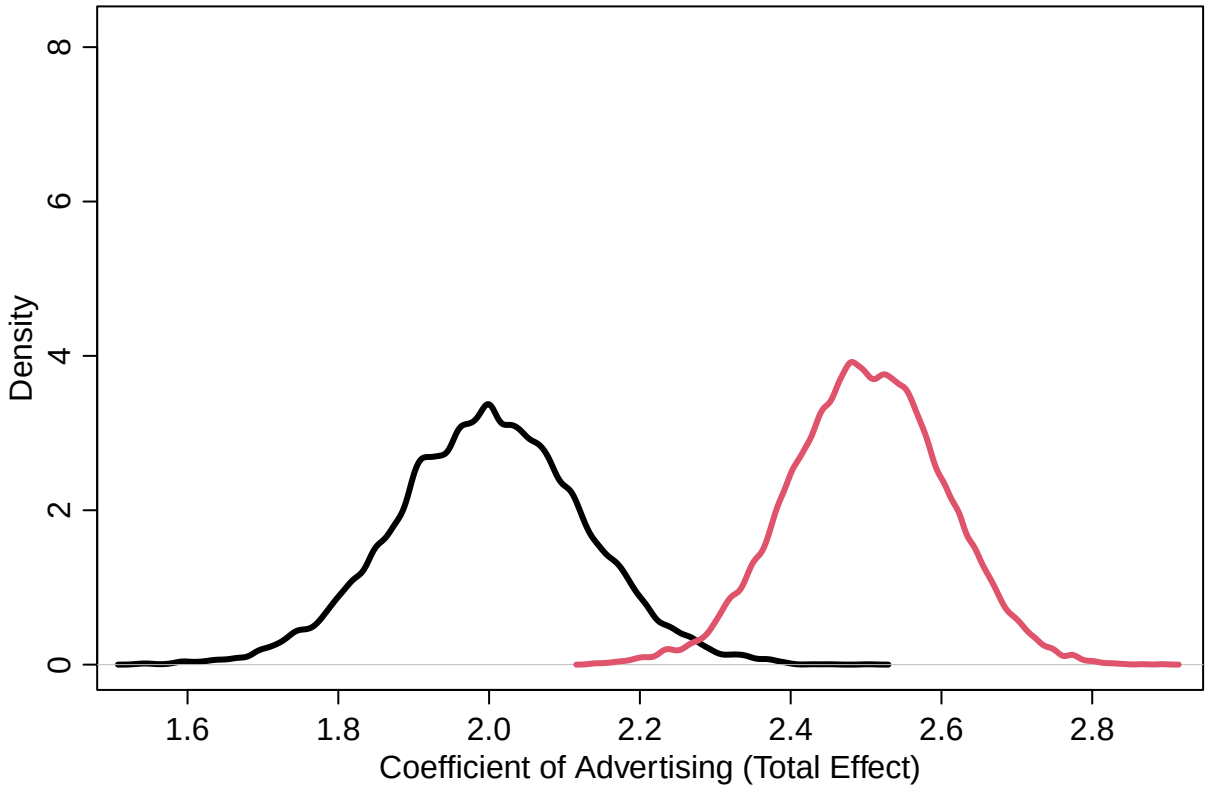
Simulation

```
# Function to simulate data and fit models
f4 <- function(n=100, bB_A=1., bQ_A=1., bQ_S=1., bEC_B=1., bEC_S=1., bA_S=2.) {
  # Simulate data
  Q <- rnorm(n)  # Quality of the product
  EC <- rnorm(n) # Economic climate
  B <- rnorm(n, bEC_B*EC )
  A <- rnorm(n, bB_A*B + bQ_A*Q )
  S <- rnorm(n, bQ_S*Q + bEC_S*EC + bA_S*A )

  # Fit models
  b_total <- coef(lm(S ~ A + B + Q))['A']  # Model controlling for B and Q
  b_incorrect <- coef(lm(S ~ A + B))['A']  # Model controlling for B only

  return(c(b_total, b_incorrect))  # Return coefficients
}
```

```r
# Perform Monte Carlo simulation
sim4 <- mcreplicate(1e4, f4(), mc.cores = 8)
```

Plotting the correct and incorrect versions of the estimate

```r
# Plot posterior distributions
range1 <- range(sim4[1,])
range2 <- range(sim4[2,])
xlim <- range(c(range1, range2))
ylim <- c(0,8.2)

dens(sim4[1,], lwd=3, xlab='Coefficient of Advertising (Total Effect)', xlim=xlim, ylim=ylim)
dens(sim4[2,], lwd=3, col=2, add=TRUE, xlim=xlim, ylim=ylim)
```



Problem 5

DAG

```r
dag5 <- dagitty('dag {
    SM [pos="0,1"]
    MH [pos="1,1"]
    OSN [pos="1,0"]
    PI [pos="0,0"]
    SM -> MH
    SM <- OSN -> MH
    PI -> SM
}')
```

```r
plot(dag5)
```

Identifying the variables to control for to estimate the direct effect of Social Media use on Mental Health

```r
adjustmentSets(dag5, exposure='SM', outcome='MH', effect = 'direct')
```

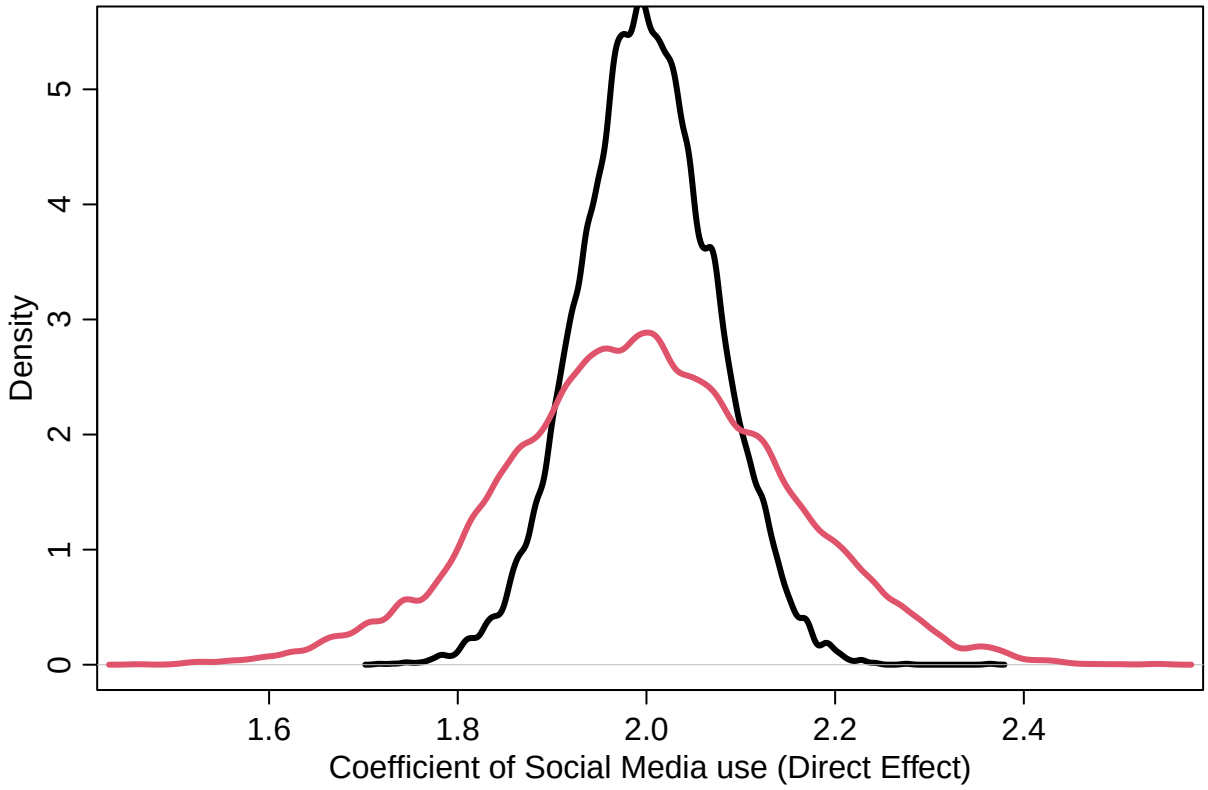
```
## { OSN }
```

We need to adjust for Offline Social Network to estimate the direct effect of Social Media use on Mental Health. This closes the backdoor through OSN. ## Simulation

```r
# Function to simulate data and fit models
f5 <- function(n=100, bPI_SM = 1, bOSN_SM = 1, bOSN_MH = 1, bSM_MH = 2) {
  # Simulate data
  PI <- rnorm(n)  # Personal Interest
  OSN <- rnorm(n) # Offline Social Network
  SM <- rnorm(n, bPI_SM*PI )
  MH <- rnorm(n, bOSN_MH*OSN + bSM_MH*SM )

  # Fit models
  b_direct <- coef(lm(MH ~ SM + OSN))['SM']  # Model controlling for OSN
  b_incorrect <- coef(lm(MH ~ SM + PI))['SM']  # Model not controlling for OSN but PI instead

  return(c(b_direct, b_incorrect))  # Return coefficients
}
# Perform Monte Carlo simulation
sim5 <- mcreplicate(1e4, f5(), mc.cores = 8)
```

Plotting the correct and incorrect versions of the estimate

```
# Plot posterior distributions
range1 <- range(sim5[1,])
range2 <- range(sim5[2,])
xlim <- range(c(range1, range2))
ylim <- c(0,5.5)

dens(sim5[1,], lwd=3, xlab='Coefficient of Social Media use (Direct Effect)', xlim=xlim, ylim=ylim)
dens(sim5[2,], lwd=3, col=2, add=TRUE, xlim=xlim, ylim=ylim)
```



Problem 6

DAG

```
dag6 <- dagitty('dag {
    PE [pos="0,1"]
    HC [pos="1,1"]
    D [pos="0,0"]
    US [pos="1, 0"]
    PE -> HC
    PE <- D -> HC
    HC -> US
}')
plot(dag6)
```

Identifying the variables to control for to estimate the direct effect of Pesticide Exposure on Health Condition

```
adjustmentSets(dag6, exposure='PE', outcome='HC', effect = 'direct')
```

```
## { D }
```

We need to either adjust for Dietary Habits to estimate the direct effect of Pesticide Exposure on Health Condition, getting rid of the backdoor through D. US does not need to be included in the model.

Simulation

```
# Function to simulate data and fit models
f6 <- function(n=100, bD_PE = 1, bD_HC = 1, bPE_HC = 1.5) {
  # Simulate data
  D <- rnorm(n)  # Dietary Habits
  PE <- rnorm(n, bD_PE*D )
  HC <- rnorm(n, bD_HC*D + bPE_HC*PE )

  # Fit models
  b_direct <- coef(lm(HC ~ PE + D))['PE']  # Model controlling for D
  b_incorrect <- coef(lm(HC ~ PE))['PE']  # Model not controlling for D

  return(c(b_direct, b_incorrect))  # Return coefficients
}
# Perform Monte Carlo simulation
sim6 <- mcreplicate(1e4, f6(), mc.cores = 8)
```

Plotting the correct and incorrect versions of the estimate

```
# Plot posterior distributions
range1 <- range(sim6[1,])
range2 <- range(sim6[2,])
xlim <- range(c(range1, range2))
ylim <- c(0,5.6)

dens(sim6[1,], lwd=3, xlab='Coefficient of Pesticide Exposure (Direct Effect)', xlim=xlim, ylim=ylim)
dens(sim6[2,], lwd=3, col=2, add=TRUE, xlim=xlim, ylim=ylim)
```