```
R version 3.2.1 (2015-06-18) -- "World-Famous Astronaut"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.66 (6956) x86_64-apple-darwin13.4.0]

>
> rm(list=ls(all=TRUE))
>
> set.seed(123)
>
> library(MASS)
> library(matrixStats)
matrixStats v0.14.2 (2015-06-23) successfully loaded. See ?matrixStats for help.
>
> #setwd("replication_archive/Leadership/")
> setwd("~/Dropbox/beliefs_incomplete data/Paper/PSRM/final/replication_archive/Leadership")
>
>
> set.seed(123)
> NS = 10000
>
>
> data <- read.table("HMS_Data.raw.txt", header=T)
>
>
>
> ### FIGURE 7
>
> obs_Q3 <- ifelse(is.na(data$CQ3a_Clinic_or_Hospitals)==F & is.na(data$FQ3a_Clinic)==F, 1, 0)
> Q3_total <- subset(data, obs_Q3 == 1)
>
> Y = Q3_total$CQ3a_Clinic_or_Hospitals
> Z = Q3_total$FQ3a_Clinic
>
> num = 1
> rho = seq(0.01, 1, .01)
> rho2 = 0
> k  = rho2
>
> mu1 = c()
> mu0 = c()
> ci.h0 = c()
> ci.h1 = c()
> ci.l0 = c()
> ci.l1 = c()
>
> for(j in rho){
+     newdata = matrix(0, length(Z), NS)
+ for(i in 1:length(Z)){
+     newdata[i,] = rbinom(NS, 1, Z[i]*Y[i]*(1-j) + Z[i]*(1-Y[i])*(1 - k) + (1 - Z[i])*Y[i]*k + (1 - Z[i])*(1 - Y[i])*j)}
+     mu1 = c(mu1, mean(newdata[Y == 1,]))
+     mu0 = c(mu0, mean(newdata[Y == 0,]))
+     ci.h0 = c(ci.h0,quantile(colMeans(newdata[Y == 0,]), .975))
+     ci.l0 = c(ci.l0,quantile(colMeans(newdata[Y == 0,]), .025))
+     ci.h1 = c(ci.h1, quantile(colMeans(newdata[Y == 1,]), .975))
+     ci.l1 = c(ci.l1, quantile(colMeans(newdata[Y == 1,]), .025))
+ }
>
> quartz(type="pdf", width=5, height=5, file="output/leader_simulate_q3c1sca.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0))
> plot(mu1~rho, type = "l", ylim = c(0,1), xlab = expression(eta), ylab = ("P(T') = 1"), lwd=3)
> dev.off()
null device
          1
>
> quartz(type="pdf", width=5, height=5, file="output/leader_simulate_q3c0sca.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0))
> plot(mu0~rho, type = "l", ylim = c(0,1), xlab = expression(eta), ylab = ("P(T') = 1"), lwd=3)
> dev.off()
null device
          1
>
>
>
> ### FIGURE 8
>
> rho = 0
> rho2 = seq(0.01, 1, .01)
> j  = rho
>
> mu1 = c()
> mu0 = c()
> ci.h0 = c()
> ci.h1 = c()
> ci.l0 = c()
> ci.l1 = c()
>
> for(k in rho2){
+     newdata = matrix(0, length(Z), NS)
+ for(i in 1:length(Z)){
+     newdata[i,] = rbinom(NS, 1, Z[i]*Y[i]*(1-j) + Z[i]*(1-Y[i])*(1 - k) + (1 - Z[i])*Y[i]*k + (1 - Z[i])*(1 - Y[i])*j)}
+     mu1 = c(mu1, mean(newdata[Y == 1,]))
+     mu0 = c(mu0, mean(newdata[Y == 0,]))
+     ci.h0 = c(ci.h0,quantile(colMeans(newdata[Y == 0,]), .975))
+     ci.l0 = c(ci.l0,quantile(colMeans(newdata[Y == 0,]), .025))
+     ci.h1 = c(ci.h1, quantile(colMeans(newdata[Y == 1,]), .975))
+     ci.l1 = c(ci.l1, quantile(colMeans(newdata[Y == 1,]), .025))
+ }
>
> quartz(type="pdf", width=5, height=5, file="output/leader_simulate_q3c1scb.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0))
> plot(mu1~rho2, type = "l", ylim = c(0,1), xlab = expression(eta), ylab = ("P(T') = 1"), lwd=3)
> dev.off()
```

```
null device
          1
>
> quartz(type="pdf", width=5, height=5, file="output/leader_simulate_q3c0scb.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0))
> plot(mu0~rho2, type = "l", ylim = c(0,1), xlab = expression(eta), ylab = ("P(T') = 1"), lwd=3)
> dev.off()
null device
          1
>
>
>
>
> ### FIGURE 9
>
>
> obs_Q3 <- ifelse(is.na(data$CQ3a_Clinic_or_Hospitals)==F & is.na(data$FQ3a_Clinic)==F, 1, 0)
> q3_data <- subset(data, obs_Q3 == 1)
>
> etavec1 = seq(0, 0.99, length.out=100)
> etavec2 = seq(0, 0.99, length.out=100)
>
> results <-  NULL
> for(i in 1:length(etavec1)){
+     for(j in 1:length(etavec2)){
+         #print(i)
+         res <- NULL
+         count <- 1
+         while(count <= 100){
+             # create data given eta
+             q3_data$newtreat <- NA
+
+             # Scenatio (a)
+             q3_data$newtreat[q3_data$FQ3a_Clinic==0 & q3_data$CQ3a_Clinic_or_Hospitals==0] <- rbinom(length(q3_data$FQ3a_Clinic[q3_data$FQ3a_Clinic==0 & q3_data
$CQ3a_Clinic_or_Hospitals==0]), 1, etavec1[i])
+             q3_data$newtreat[q3_data$FQ3a_Clinic==1 & q3_data$CQ3a_Clinic_or_Hospitals==1] <- rbinom(length(q3_data$FQ3a_Clinic[q3_data$FQ3a_Clinic==1 & q3_data
$CQ3a_Clinic_or_Hospitals==1]), 1, 1-etavec1[i])
+
+             # Scenario (b)
+             q3_data$newtreat[q3_data$FQ3a_Clinic==0 & q3_data$CQ3a_Clinic_or_Hospitals==1] <- rbinom(length(q3_data$FQ3a_Clinic[q3_data$FQ3a_Clinic==0 & q3_data
$CQ3a_Clinic_or_Hospitals==1]), 1, etavec2[j])
+             q3_data$newtreat[q3_data$FQ3a_Clinic==1 & q3_data$CQ3a_Clinic_or_Hospitals==0] <- rbinom(length(q3_data$FQ3a_Clinic[q3_data$FQ3a_Clinic==1 & q3_data
$CQ3a_Clinic_or_Hospitals==0]), 1, 1-etavec2[j])
+
+
+             if(0 %in% c(table(q3_data$newtreat, q3_data$CQ3a_Clinic_or_Hospitals))) next
+
+             # run model and get ATE
+             m <- glm(CQ3a_Clinic_or_Hospitals ~ newtreat, family=binomial(link="logit"), data=q3_data)
+             summary(m)
+
+             yes <- 1/(1+exp(-(c(1,1) %*% coef(m))))
+             no <- 1/(1+exp(-(c(1,0) %*% coef(m))))
+             atepoint <- yes-no
+
+             draw <- mvrnorm(1000, coef(m), vcov(m))
+             yesvec <- 1/(1+exp(-(draw %*% c(1,1))))
+             novec <- 1/(1+exp(-(draw %*% c(1,0))))
+             atepointvec <- yesvec-novec
+
+             #res <- rbind(res, c(atepoint, sd(atepointvec)))
+             res <- rbind(res, c(atepoint, sd(atepointvec), quantile(atepointvec, 0.025), quantile(atepointvec, 0.975)))
+             count <- count+1
+         }
+         results <- rbind(results, c(etavec1[i], etavec2[j], colMedians(res)))
+     }
+ }
>
> colnames(results) <- c("eta1", "eta2", "pointest", "stderr", "lowerci95", "upperci95")
> results <- as.data.frame(results)
>
> results$pval <- 2*pnorm(-abs(results$pointest/results$stderr))
>
>
> # p-values
> results$colors <- NA
> results$colors <- ifelse(results$pval <= .01, "black", results$colors)
> results$colors <- ifelse(results$pval > .01 & results$pval <= .05, "grey40", results$colors)
> results$colors <- ifelse(results$pval > .05 & results$pval <= .1, "grey80", results$colors)
> results$colors <- ifelse(results$pval > .1, "white", results$colors)
>
>
> quartz(type="pdf", width=5, height=5, file="output/leadership_ext.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0))
> plot(results$eta1, results$eta2, col = results$colors, xlab = expression(eta[1]), ylab = expression(eta[2]), pch=15, cex=0.8)
> text(0.05, 0.93, "+", col="white", cex=4, font=2)
> text(0.95, 0.03, "-", col="white", cex=4, font=2)
> dev.off()
null device
          1
>
>
>
>
>
> ### FIGURE 10
>
> obs_Q4a <- ifelse(is.na(data$CQ4a_Ed_Prim_or_Secondary)==F & is.na(data$FQ4a_Ed_Prim)==F, 1, 0)
> Q4_total <- subset(data, obs_Q4a == 1)
>
>
> Y = Q4_total$CQ4a_Ed_Prim_or_Secondary
> Z = Q4_total$FQ4a_Ed_Prim
>
>
> num = 1
> rho = seq(0.01, 1, .01)
> rho2 = 0
> k  = rho2
>
> mu1 = c()
> mu0 = c()
> ci.h0 = c()
> ci.h1 = c()
> ci.l0 = c()
> ci.l1 = c()
>
> for(j in rho){
+     newdata = matrix(0, length(Z), NS)
```

```
+ for(i in 1:length(Z)){
+     newdata[i,] = rbinom(NS, 1, Z[i]*Y[i]*(1-j) + Z[i]*(1-Y[i])*(1 - k) + (1 - Z[i])*Y[i]*k + (1 - Z[i])*(1 - Y[i])*j)}
+     mu1 = c(mu1, mean(newdata[Y == 1,]))
+     mu0 = c(mu0, mean(newdata[Y == 0,]))
+     ci.h0 = c(ci.h0,quantile(colMeans(newdata[Y == 0,]), .975))
+     ci.l0 = c(ci.l0,quantile(colMeans(newdata[Y == 0,]), .025))
+     ci.h1 = c(ci.h1, quantile(colMeans(newdata[Y == 1,]), .975))
+     ci.l1 = c(ci.l1, quantile(colMeans(newdata[Y == 1,]), .025))
+ }
>
> quartz(type="pdf", width=5, height=5, file="output/leader_simulate_q4ac1sca.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0))
> plot(mu1~rho, type = "l", ylim = c(0,1), xlab = expression(eta), ylab = ("P(T') = 1"), lwd=3)
> dev.off()
null device
          1
>
> quartz(type="pdf", width=5, height=5, file="output/leader_simulate_q4ac0sca.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0))
> plot(mu0~rho, type = "l", ylim = c(0,1), xlab = expression(eta), ylab = ("P(T') = 1"), lwd=3)
> dev.off()
null device
          1
>
>
>
> ### FIGURE 11
>
> rho = 0
> rho2 = seq(0.01, 1, .01)
> j  = rho
>
> mu1 = c()
> mu0 = c()
> ci.h0 = c()
> ci.h1 = c()
> ci.l0 = c()
> ci.l1 = c()
>
> for(k in rho2){
+     newdata = matrix(0, length(Z), NS)
+ for(i in 1:length(Z)){
+     newdata[i,] = rbinom(NS, 1, Z[i]*Y[i]*(1-j) + Z[i]*(1-Y[i])*(1 - k) + (1 - Z[i])*Y[i]*k + (1 - Z[i])*(1 - Y[i])*j)}
+     mu1 = c(mu1, mean(newdata[Y == 1,]))
+     mu0 = c(mu0, mean(newdata[Y == 0,]))
+     ci.h0 = c(ci.h0,quantile(colMeans(newdata[Y == 0,]), .975))
+     ci.l0 = c(ci.l0,quantile(colMeans(newdata[Y == 0,]), .025))
+     ci.h1 = c(ci.h1, quantile(colMeans(newdata[Y == 1,]), .975))
+     ci.l1 = c(ci.l1, quantile(colMeans(newdata[Y == 1,]), .025))
+ }
>
> quartz(type="pdf", width=5, height=5, file="output/leader_simulate_q4ac1scb.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0))
> plot(mu1~rho2, type = "l", ylim = c(0,1), xlab = expression(eta), ylab = ("P(T') = 1"), lwd=3)
> dev.off()
null device
          1
>
> quartz(type="pdf", width=5, height=5, file="output/leader_simulate_q4ac0scb.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0))
> plot(mu0~rho2, type = "l", ylim = c(0,1), xlab = expression(eta), ylab = ("P(T') = 1"), lwd=3)
> dev.off()
null device
          1
>
>
>
> ### FIGURE 12
>
> obs_Q4a <- ifelse(is.na(data$CQ4a_Ed_Prim_or_Secondary)==F & is.na(data$FQ4a_Ed_Prim)==F, 1, 0)
> q4_data <- subset(data, obs_Q4a == 1)
>
> m <- glm(CQ4a_Ed_Prim_or_Secondary ~ FQ4a_Ed_Prim, family=binomial(link="logit"), data=q4_data)
> summary(m)

Call:
glm(formula = CQ4a_Ed_Prim_or_Secondary ~ FQ4a_Ed_Prim, family = binomial(link = "logit"),
    data = q4_data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7751  -1.0842   0.6809   1.2735   1.2735

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -0.2231     0.3000  -0.744  0.45699
FQ4a_Ed_Prim   1.5669     0.5479   2.860  0.00424 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 100.631  on 73  degrees of freedom
Residual deviance:  91.396  on 72  degrees of freedom
AIC: 95.396

Number of Fisher Scoring iterations: 4

>
>
> yes <- 1/(1+exp(-(c(1,1) %*% coef(m))))
> no <- 1/(1+exp(-(c(1,0) %*% coef(m))))
> atepoint <- yes-no
> atepoint
          [,1]
[1,] 0.348659
>
> draw <- mvrnorm(1000, coef(m), vcov(m))
> yesvec <- 1/(1+exp(-(draw %*% c(1,1))))
> novec <- 1/(1+exp(-(draw %*% c(1,0))))
> atepointvec <- yesvec-novec
> quantile(atepointvec, c(0.025, 0.975))
     2.5%     97.5%
0.1184683 0.5280012
>
>
>
> ### Run the sensitivity analysis: Scenario (a)
>
```

```
> etavec = seq(0, 0.99, length.out=100)
>
> results <-  NULL
> for(i in 1:length(etavec)){
+     res <- NULL
+     count <- 1
+     while(count <= 500){
+         # create data given eta
+         q4_data$newtreat <- NA
+         q4_data$newtreat[q4_data$FQ4a_Ed_Prim==0 & q4_data$CQ4a_Ed_Prim_or_Secondary==0] <- rbinom(length(q4_data$FQ4a_Ed_Prim[q4_data$FQ4a_Ed_Prim==0 & q4_data
$CQ4a_Ed_Prim_or_Secondary==0]), 1, etavec[i])
+         q4_data$newtreat[q4_data$FQ4a_Ed_Prim==1 & q4_data$CQ4a_Ed_Prim_or_Secondary==1] <- rbinom(length(q4_data$FQ4a_Ed_Prim[q4_data$FQ4a_Ed_Prim==1 & q4_data
$CQ4a_Ed_Prim_or_Secondary==1]), 1, 1-etavec[i])
+         q4_data$newtreat[q4_data$FQ4a_Ed_Prim==0 & q4_data$CQ4a_Ed_Prim_or_Secondary==1] <- q4_data$FQ4a_Ed_Prim[q4_data$FQ4a_Ed_Prim==0 & q4_data$CQ4a_Ed_Prim_or_Secondary==1]
+         q4_data$newtreat[q4_data$FQ4a_Ed_Prim==1 & q4_data$CQ4a_Ed_Prim_or_Secondary==0] <- q4_data$FQ4a_Ed_Prim[q4_data$FQ4a_Ed_Prim==1 & q4_data$CQ4a_Ed_Prim_or_Secondary==0]
+         if(0 %in% c(table(q4_data$newtreat, q4_data$CQ4a_Ed_Prim_or_Secondary))) next
+
+         # run model and get ATE
+         m <- glm(CQ4a_Ed_Prim_or_Secondary ~ newtreat, family=binomial(link="logit"), data=q4_data)
+         summary(m)
+
+         yes <- 1/(1+exp(-(c(1,1) %*% coef(m))))
+         no <- 1/(1+exp(-(c(1,0) %*% coef(m))))
+         atepoint <- yes-no
+
+         draw <- mvrnorm(1000, coef(m), vcov(m))
+         yesvec <- 1/(1+exp(-(draw %*% c(1,1))))
+         novec <- 1/(1+exp(-(draw %*% c(1,0))))
+         atepointvec <- yesvec-novec
+
+         #res <- rbind(res, c(atepoint, sd(atepointvec)))
+         res <- rbind(res, c(atepoint, quantile(atepointvec, 0.025), quantile(atepointvec, 0.975)))
+         count <- count+1
+     }
+     results <- rbind(results, c(etavec[i], colMedians(res)))
+ }
>
> colnames(results) <- c("eta", "pointest", "lowerci95", "upperci95")
> results <- as.data.frame(results)
>
>
> quartz(type="pdf", width=5, height=5, file="output/leadership_q4_sc1.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0), family="CMU Serif")
> plot(1:length(results$pointest), results$pointest, type="n", ylim=c(-1, 1), xlab = expression(eta), ylab="Average Treatment Effect", xaxt="n")
> polygon(c(1:length(results$pointest), rev(1:length(results$pointest))), c(results$lowerci95,rev(results$upperci95)), col="grey", border=NA)
> points(1:length(results$pointest), results$pointest, type="l", lwd=3)
> axis(1, at=seq(1, 100, length.out=6), labels = seq(0, 1, length.out=6), las=2)
> abline(h=0, col = "black", lwd=2)
> lines(c(1,1), c(results$lowerci95[1], results$upperci95[1]), lwd=3)
> points(1, results$pointest[1], pch=16)
> dev.off()
null device
          1
>
>
>
>
>
> ### Run the sensitivity analysis: Scenario (b)
>
> etavec = seq(0, 0.99, length.out=100)
>
> results <-  NULL
> for(i in 1:length(etavec)){
+     res <- NULL
+     count <- 1
+     while(count <= 500){
+         # create data given eta
+         q4_data$newtreat <- NA
+         q4_data$newtreat[q4_data$FQ4a_Ed_Prim==0 & q4_data$CQ4a_Ed_Prim_or_Secondary==1] <- rbinom(length(q4_data$FQ4a_Ed_Prim[q4_data$FQ4a_Ed_Prim==0 & q4_data
$CQ4a_Ed_Prim_or_Secondary==1]), 1, etavec[i])
+         q4_data$newtreat[q4_data$FQ4a_Ed_Prim==1 & q4_data$CQ4a_Ed_Prim_or_Secondary==0] <- rbinom(length(q4_data$FQ4a_Ed_Prim[q4_data$FQ4a_Ed_Prim==1 & q4_data
$CQ4a_Ed_Prim_or_Secondary==0]), 1, 1-etavec[i])
+         q4_data$newtreat[q4_data$FQ4a_Ed_Prim==1 & q4_data$CQ4a_Ed_Prim_or_Secondary==1] <- q4_data$FQ4a_Ed_Prim[q4_data$FQ4a_Ed_Prim==1 & q4_data$CQ4a_Ed_Prim_or_Secondary==1]
+         q4_data$newtreat[q4_data$FQ4a_Ed_Prim==0 & q4_data$CQ4a_Ed_Prim_or_Secondary==0] <- q4_data$FQ4a_Ed_Prim[q4_data$FQ4a_Ed_Prim==0 & q4_data$CQ4a_Ed_Prim_or_Secondary==0]
+         if(0 %in% c(table(q4_data$newtreat, q4_data$CQ4a_Ed_Prim_or_Secondary))) next
+
+         # run model and get ATE
+         m <- glm(CQ4a_Ed_Prim_or_Secondary ~ newtreat, family=binomial(link="logit"), data=q4_data)
+         summary(m)
+
+         yes <- 1/(1+exp(-(c(1,1) %*% coef(m))))
+         no <- 1/(1+exp(-(c(1,0) %*% coef(m))))
+         atepoint <- yes-no
+
+         draw <- mvrnorm(1000, coef(m), vcov(m))
+         yesvec <- 1/(1+exp(-(draw %*% c(1,1))))
+         novec <- 1/(1+exp(-(draw %*% c(1,0))))
+         atepointvec <- yesvec-novec
+
+         res <- rbind(res, c(atepoint, quantile(atepointvec, 0.025), quantile(atepointvec, 0.975)))
+         count <- count+1
+     }
+     results <- rbind(results, c(etavec[i], colMedians(res)))
+ }
>
> colnames(results) <- c("eta", "pointest", "lowerci95", "upperci95")
> results <- as.data.frame(results)
>
>
> quartz(type="pdf", width=5, height=5, file="output/leadership_q4_sc2.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0), family="CMU Serif")
> plot(1:length(results$pointest), results$pointest, type="n", ylim=c(-1, 1), xlab = expression(eta), ylab="Average Treatment Effect", xaxt="n")
> polygon(c(1:length(results$pointest), rev(1:length(results$pointest))), c(results$lowerci95,rev(results$upperci95)), col="grey", border=NA)
> points(1:length(results$pointest), results$pointest, type="l", lwd=3)
> axis(1, at=seq(1, 100, length.out=6), labels = seq(0, 1, length.out=6), las=2)
> abline(h=0, col = "black", lwd=2)
> lines(c(1,1), c(results$lowerci95[1], results$upperci95[1]), lwd=3)
> points(1, results$pointest[1], pch=16)
> dev.off()
null device
          1
>
>
>
>
>
> ### FIGURE 13
>
```

```
> obs_Q7b <- ifelse(is.na(data$CQ7b_Tr_Roads_or_Services)==F & is.na(data$FQ7b_Tr_Roads)==F, 1, 0)
> Q7_total <- subset(data, obs_Q7b == 1)
>
>
> Y = Q7_total$CQ7b_Tr_Roads_or_Services
> Z = Q7_total$FQ7b_Tr_Roads
>
>
> num = 1
> rho = seq(0.01, 1, .01)
> rho2 = 0
> k  = rho2
>
> mu1 = c()
> mu0 = c()
> ci.h0 = c()
> ci.h1 = c()
> ci.l0 = c()
> ci.l1 = c()
>
> for(j in rho){
+     newdata = matrix(0, length(Z), NS)
+ for(i in 1:length(Z)){
+     newdata[i,] = rbinom(NS, 1, Z[i]*Y[i]*(1-j) + Z[i]*(1-Y[i])*(1 - k) + (1 - Z[i])*Y[i]*k + (1 - Z[i])*(1 - Y[i])*j)}
+     mu1 = c(mu1, mean(newdata[Y == 1,]))
+     mu0 = c(mu0, mean(newdata[Y == 0,]))
+     ci.h0 = c(ci.h0,quantile(colMeans(newdata[Y == 0,]), .975))
+     ci.l0 = c(ci.l0,quantile(colMeans(newdata[Y == 0,]), .025))
+     ci.h1 = c(ci.h1, quantile(colMeans(newdata[Y == 1,]), .975))
+     ci.l1 = c(ci.l1, quantile(colMeans(newdata[Y == 1,]), .025))
+ }
>
> quartz(type="pdf", width=5, height=5, file="output/leader_simulate_q7bc1sca.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0))
> plot(mu1~rho, type = "l", ylim = c(0,1), xlab = expression(eta), ylab = ("P(T') = 1"), lwd=3)
> dev.off()
null device
          1
>
> quartz(type="pdf", width=5, height=5, file="output/leader_simulate_q7bc0sca.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0))
> plot(mu0~rho, type = "l", ylim = c(0,1), xlab = expression(eta), ylab = ("P(T') = 1"), lwd=3)
> dev.off()
null device
          1
>
>
>
> ### FIGURE 14
>
> rho = 0
> rho2 = seq(0.01, 1, .01)
> j  = rho
>
> mu1 = c()
> mu0 = c()
> ci.h0 = c()
> ci.h1 = c()
> ci.l0 = c()
> ci.l1 = c()
>
> for(k in rho2){
+     newdata = matrix(0, length(Z), NS)
+ for(i in 1:length(Z)){
+     newdata[i,] = rbinom(NS, 1, Z[i]*Y[i]*(1-j) + Z[i]*(1-Y[i])*(1 - k) + (1 - Z[i])*Y[i]*k + (1 - Z[i])*(1 - Y[i])*j)}
+     mu1 = c(mu1, mean(newdata[Y == 1,]))
+     mu0 = c(mu0, mean(newdata[Y == 0,]))
+     ci.h0 = c(ci.h0,quantile(colMeans(newdata[Y == 0,]), .975))
+     ci.l0 = c(ci.l0,quantile(colMeans(newdata[Y == 0,]), .025))
+     ci.h1 = c(ci.h1, quantile(colMeans(newdata[Y == 1,]), .975))
+     ci.l1 = c(ci.l1, quantile(colMeans(newdata[Y == 1,]), .025))
+ }
>
> quartz(type="pdf", width=5, height=5, file="output/leader_simulate_q7bc1scb.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0))
> plot(mu1~rho2, type = "l", ylim = c(0,1), xlab = expression(eta), ylab = ("P(T') = 1"), lwd=3)
> dev.off()
null device
          1
>
> quartz(type="pdf", width=5, height=5, file="output/leader_simulate_q7bc0scb.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0))
> plot(mu0~rho2, type = "l", ylim = c(0,1), xlab = expression(eta), ylab = ("P(T') = 1"), lwd=3)
> dev.off()
null device
          1
>
>
>
> ### FIGURE 15
>
> obs_Q7b <- ifelse(is.na(data$CQ7b_Tr_Roads_or_Services)==F & is.na(data$FQ7b_Tr_Roads)==F, 1, 0)
> q7_data <- subset(data, obs_Q7b == 1)
>
> m <- glm(CQ7b_Tr_Roads_or_Services ~ FQ7b_Tr_Roads, family=binomial(link="logit"), data=q7_data)
> summary(m)

Call:
glm(formula = CQ7b_Tr_Roads_or_Services ~ FQ7b_Tr_Roads, family = binomial(link = "logit"),
    data = q7_data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.5350  -0.5302  -0.5302  -0.5302   2.0156

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -1.89085    0.37930  -4.985 6.19e-07 ***
FQ7b_Tr_Roads  0.01905    0.84899   0.022    0.982
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 59.185  on 75  degrees of freedom
Residual deviance: 59.185  on 74  degrees of freedom
AIC: 63.185

Number of Fisher Scoring iterations: 4
```

```
>
>
> yes <- 1/(1+exp(-(c(1,1) %*% coef(m))))
> no <- 1/(1+exp(-(c(1,0) %*% coef(m))))
> atepoint <- yes-no
> atepoint
           [,1]
[1,] 0.002185792
>
> draw <- mvrnorm(1000, coef(m), vcov(m))
> yesvec <- 1/(1+exp(-(draw %*% c(1,1))))
> novec <- 1/(1+exp(-(draw %*% c(1,0))))
> atepointvec <- yesvec-novec
> quantile(atepointvec, c(0.025, 0.975))
      2.5%      97.5%
-0.1473607  0.2583290
>
>
>
> ### Run the sensitivity analysis: Scenario (a)
>
> etavec = seq(0, 0.99, length.out=100)
>
> results <-  NULL
> for(i in 1:length(etavec)){
+     res <- NULL
+     count <- 1
+     while(count <= 500){
+         # create data given eta
+         q7_data$newtreat <- NA
+         q7_data$newtreat[q7_data$FQ7b_Tr_Roads==0 & q7_data$CQ7b_Tr_Roads_or_Services==0] <- rbinom(length(q7_data$FQ7b_Tr_Roads[q7_data$FQ7b_Tr_Roads==0 & q7_data
$CQ7b_Tr_Roads_or_Services==0]), 1, etavec[i])
+         q7_data$newtreat[q7_data$FQ7b_Tr_Roads==1 & q7_data$CQ7b_Tr_Roads_or_Services==1] <- rbinom(length(q7_data$FQ7b_Tr_Roads[q7_data$FQ7b_Tr_Roads==1 & q7_data
$CQ7b_Tr_Roads_or_Services==1]), 1, 1-etavec[i])
+         q7_data$newtreat[q7_data$FQ7b_Tr_Roads==0 & q7_data$CQ7b_Tr_Roads_or_Services==1] <- q7_data$FQ7b_Tr_Roads[q7_data$FQ7b_Tr_Roads==0 & q7_data$CQ7b_Tr_Roads_or_Services==1]
+         q7_data$newtreat[q7_data$FQ7b_Tr_Roads==1 & q7_data$CQ7b_Tr_Roads_or_Services==0] <- q7_data$FQ7b_Tr_Roads[q7_data$FQ7b_Tr_Roads==1 & q7_data$CQ7b_Tr_Roads_or_Services==0]
+         if(0 %in% c(table(q7_data$newtreat, q7_data$CQ7b_Tr_Roads_or_Services))) next
+
+         # run model and get ATE
+         m <- glm(CQ7b_Tr_Roads_or_Services ~ newtreat, family=binomial(link="logit"), data=q7_data)
+         summary(m)
+
+         yes <- 1/(1+exp(-(c(1,1) %*% coef(m))))
+         no <- 1/(1+exp(-(c(1,0) %*% coef(m))))
+         atepoint <- yes-no
+
+         draw <- mvrnorm(1000, coef(m), vcov(m))
+         yesvec <- 1/(1+exp(-(draw %*% c(1,1))))
+         novec <- 1/(1+exp(-(draw %*% c(1,0))))
+         atepointvec <- yesvec-novec
+
+         #res <- rbind(res, c(atepoint, sd(atepointvec)))
+         res <- rbind(res, c(atepoint, quantile(atepointvec, 0.025), quantile(atepointvec, 0.975)))
+         count <- count+1
+     }
+     results <- rbind(results, c(etavec[i], colMedians(res)))
+ }
>
> colnames(results) <- c("eta", "pointest", "lowerci95", "upperci95")
> results <- as.data.frame(results)
>
>
> quartz(type="pdf", width=5, height=5, file="output/leadership_q7_sc1.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0), family="CMU Serif")
> plot(1:length(results$pointest), results$pointest, type="n", ylim=c(-1, 1), xlab = expression(eta), ylab="Average Treatment Effect", xaxt="n")
> polygon(c(1:length(results$pointest), rev(1:length(results$pointest))), c(results$lowerci95,rev(results$upperci95)), col="grey", border=NA)
> points(1:length(results$pointest), results$pointest, type="l", lwd=3)
> axis(1, at=seq(1, 100, length.out=6), labels = seq(0, 1, length.out=6), las=2)
> abline(h=0, col = "black", lwd=2)
> lines(c(1,1), c(results$lowerci95[1], results$upperci95[1]), lwd=3)
> points(1, results$pointest[1], pch=16)
> dev.off()
null device
          1
>
>
>
>
>
> ### Run the sensitivity analysis: Scenario (b)
>
> etavec = seq(0, 0.99, length.out=100)
>
> results <-  NULL
> for(i in 1:length(etavec)){
+     res <- NULL
+     count <- 1
+     while(count <= 500){
+         # create data given eta
+         q7_data$newtreat <- NA
+         q7_data$newtreat[q7_data$FQ7b_Tr_Roads==0 & q7_data$CQ7b_Tr_Roads_or_Services==1] <- rbinom(length(q7_data$FQ7b_Tr_Roads[q7_data$FQ7b_Tr_Roads==0 & q7_data
$CQ7b_Tr_Roads_or_Services==1]), 1, etavec[i])
+         q7_data$newtreat[q7_data$FQ7b_Tr_Roads==1 & q7_data$CQ7b_Tr_Roads_or_Services==0] <- rbinom(length(q7_data$FQ7b_Tr_Roads[q7_data$FQ7b_Tr_Roads==1 & q7_data
$CQ7b_Tr_Roads_or_Services==0]), 1, 1-etavec[i])
+         q7_data$newtreat[q7_data$FQ7b_Tr_Roads==1 & q7_data$CQ7b_Tr_Roads_or_Services==1] <- q7_data$FQ7b_Tr_Roads[q7_data$FQ7b_Tr_Roads==1 & q7_data$CQ7b_Tr_Roads_or_Services==1]
+         q7_data$newtreat[q7_data$FQ7b_Tr_Roads==0 & q7_data$CQ7b_Tr_Roads_or_Services==0] <- q7_data$FQ7b_Tr_Roads[q7_data$FQ7b_Tr_Roads==0 & q7_data$CQ7b_Tr_Roads_or_Services==0]
+         if(0 %in% c(table(q7_data$newtreat, q7_data$CQ7b_Tr_Roads_or_Services))) next
+
+         # run model and get ATE
+         m <- glm(CQ7b_Tr_Roads_or_Services ~ newtreat, family=binomial(link="logit"), data=q7_data)
+         summary(m)
+
+         yes <- 1/(1+exp(-(c(1,1) %*% coef(m))))
+         no <- 1/(1+exp(-(c(1,0) %*% coef(m))))
+         atepoint <- yes-no
+
+         draw <- mvrnorm(1000, coef(m), vcov(m))
+         yesvec <- 1/(1+exp(-(draw %*% c(1,1))))
+         novec <- 1/(1+exp(-(draw %*% c(1,0))))
+         atepointvec <- yesvec-novec
+
+         res <- rbind(res, c(atepoint, quantile(atepointvec, 0.025), quantile(atepointvec, 0.975)))
+         count <- count+1
+     }
+     results <- rbind(results, c(etavec[i], colMedians(res)))
+ }
>
> colnames(results) <- c("eta", "pointest", "lowerci95", "upperci95")
```

```
> results <- as.data.frame(results)
>
>
> quartz(type="pdf", width=5, height=5, file="output/leadership_q7_sc2.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0), family="CMU Serif")
> plot(1:length(results$pointest), results$pointest, type="n", ylim=c(-1, 1), xlab = expression(eta), ylab="Average Treatment Effect", xaxt="n")
> polygon(c(1:length(results$pointest), rev(1:length(results$pointest))), c(results$lowerci95,rev(results$upperci95)), col="grey", border=NA)
> points(1:length(results$pointest), results$pointest, type="l", lwd=3)
> axis(1, at=seq(1, 100, length.out=6), labels = seq(0, 1, length.out=6), las=2)
> abline(h=0, col = "black", lwd=2)
> lines(c(1,1), c(results$lowerci95[1], results$upperci95[1]), lwd=3)
> points(1, results$pointest[1], pch=16)
> dev.off()
null device
          1
>
>
>
>
>
>
>
>
> ### FIGURE 16
>
> obs_Q11a <- ifelse(is.na(data$CQ11a_Discount_Binary)==F & is.na(data$FQ11a_Discount_Bin)==F, 1, 0)
> q11_data <- subset(data, obs_Q11a == 1)
>
>
> Y = q11_data$CQ11a_Discount_Binary
> Z = q11_data$FQ11a_Discount_Bin
>
>
> num = 1
> rho = seq(0.01, 1, .01)
> rho2 = 0
> k  = rho2
>
> mu1 = c()
> mu0 = c()
> ci.h0 = c()
> ci.h1 = c()
> ci.l0 = c()
> ci.l1 = c()
>
> for(j in rho){
+     newdata = matrix(0, length(Z), NS)
+ for(i in 1:length(Z)){
+     newdata[i,] = rbinom(NS, 1, Z[i]*Y[i]*(1-j) + Z[i]*(1-Y[i])*(1 - k) + (1 - Z[i])*Y[i]*k + (1 - Z[i])*(1 - Y[i])*j)}
+     mu1 = c(mu1, mean(newdata[Y == 1,]))
+     mu0 = c(mu0, mean(newdata[Y == 0,]))
+     ci.h0 = c(ci.h0,quantile(colMeans(newdata[Y == 0,]), .975))
+     ci.l0 = c(ci.l0,quantile(colMeans(newdata[Y == 0,]), .025))
+     ci.h1 = c(ci.h1, quantile(colMeans(newdata[Y == 1,]), .975))
+     ci.l1 = c(ci.l1, quantile(colMeans(newdata[Y == 1,]), .025))
+ }
>
> quartz(type="pdf", width=5, height=5, file="output/leader_simulate_q11ac1sca.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0))
> plot(mu1~rho, type = "l", ylim = c(0,1), xlab = expression(eta), ylab = ("P(T') = 1"), lwd=3)
> dev.off()
null device
          1
>
> quartz(type="pdf", width=5, height=5, file="output/leader_simulate_q11ac0sca.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0))
> plot(mu0~rho, type = "l", ylim = c(0,1), xlab = expression(eta), ylab = ("P(T') = 1"), lwd=3)
> dev.off()
null device
          1
>
>
>
> ### FIGURE 17
>
> rho = 0
> rho2 = seq(0.01, 1, .01)
> j  = rho
>
> mu1 = c()
> mu0 = c()
> ci.h0 = c()
> ci.h1 = c()
> ci.l0 = c()
> ci.l1 = c()
>
> for(k in rho2){
+     newdata = matrix(0, length(Z), NS)
+ for(i in 1:length(Z)){
+     newdata[i,] = rbinom(NS, 1, Z[i]*Y[i]*(1-j) + Z[i]*(1-Y[i])*(1 - k) + (1 - Z[i])*Y[i]*k + (1 - Z[i])*(1 - Y[i])*j)}
+     mu1 = c(mu1, mean(newdata[Y == 1,]))
+     mu0 = c(mu0, mean(newdata[Y == 0,]))
+     ci.h0 = c(ci.h0,quantile(colMeans(newdata[Y == 0,]), .975))
+     ci.l0 = c(ci.l0,quantile(colMeans(newdata[Y == 0,]), .025))
+     ci.h1 = c(ci.h1, quantile(colMeans(newdata[Y == 1,]), .975))
+     ci.l1 = c(ci.l1, quantile(colMeans(newdata[Y == 1,]), .025))
+ }
>
> quartz(type="pdf", width=5, height=5, file="output/leader_simulate_q11ac1scb.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0))
> plot(mu1~rho2, type = "l", ylim = c(0,1), xlab = expression(eta), ylab = ("P(T') = 1"), lwd=3)
> dev.off()
null device
          1
>
> quartz(type="pdf", width=5, height=5, file="output/leader_simulate_q11ac0scb.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0))
> plot(mu0~rho2, type = "l", ylim = c(0,1), xlab = expression(eta), ylab = ("P(T') = 1"), lwd=3)
> dev.off()
null device
          1
>
>
>
> ### FIGURE 18
>
> obs_Q11a <- ifelse(is.na(data$CQ11a_Discount_Binary)==F & is.na(data$FQ11a_Discount_Bin)==F, 1, 0)
> q11_data <- subset(data, obs_Q11a == 1)
>
```

```
> m <- glm(CQ11a_Discount_Binary ~ FQ11a_Discount_Bin, family=binomial(link="logit"), data=q11_data)
> summary(m)

Call:
glm(formula = CQ11a_Discount_Binary ~ FQ11a_Discount_Bin, family = binomial(link = "logit"),
    data = q11_data)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.177  -1.177  -1.023   1.177   1.340

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)        -0.3747     0.3917  -0.957    0.339
FQ11a_Discount_Bin  0.3747     0.4647   0.806    0.420

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 125.88  on 90  degrees of freedom
Residual deviance: 125.22  on 89  degrees of freedom
AIC: 129.22

Number of Fisher Scoring iterations: 4

>
>
> yes <- 1/(1+exp(-(c(1,1) %*% coef(m))))
> no <- 1/(1+exp(-(c(1,0) %*% coef(m))))
> atepoint <- yes-no
> atepoint
             [,1]
[1,] 0.09259259
>
> draw <- mvrnorm(1000, coef(m), vcov(m))
> yesvec <- 1/(1+exp(-(draw %*% c(1,1))))
> novec <- 1/(1+exp(-(draw %*% c(1,0))))
> atepointvec <- yesvec-novec
> quantile(atepointvec, c(0.025, 0.975))
      2.5%      97.5%
-0.1457325  0.2918926
>
>
>
> ### Run the sensitivity analysis: Scenario (a)
>
> etavec = seq(0, 0.99, length.out=100)
>
> results <-  NULL
> for(i in 1:length(etavec)){
+     res <- NULL
+     count <- 1
+     while(count <= 500){
+         # create data given eta
+         q11_data$newtreat <- NA
+         q11_data$newtreat[q11_data$FQ11a_Discount_Bin==0 & q11_data$CQ11a_Discount_Binary==0] <- rbinom(length(q11_data$FQ11a_Discount_Bin[q11_data$FQ11a_Discount_Bin==0 & q11_data
$CQ11a_Discount_Binary==0]), 1, etavec[i])
+         q11_data$newtreat[q11_data$FQ11a_Discount_Bin==1 & q11_data$CQ11a_Discount_Binary==1] <- rbinom(length(q11_data$FQ11a_Discount_Bin[q11_data$FQ11a_Discount_Bin==1 & q11_data
$CQ11a_Discount_Binary==1]), 1, 1-etavec[i])
+         q11_data$newtreat[q11_data$FQ11a_Discount_Bin==0 & q11_data$CQ11a_Discount_Binary==1] <- q11_data$FQ11a_Discount_Bin[q11_data$FQ11a_Discount_Bin==0 & q11_data
$CQ11a_Discount_Binary==1]
+         q11_data$newtreat[q11_data$FQ11a_Discount_Bin==1 & q11_data$CQ11a_Discount_Binary==0] <- q11_data$FQ11a_Discount_Bin[q11_data$FQ11a_Discount_Bin==1 & q11_data
$CQ11a_Discount_Binary==0]
+         if(0 %in% c(table(q11_data$newtreat, q11_data$CQ11a_Discount_Binary))) next
+
+         # run model and get ATE
+         m <- glm(CQ11a_Discount_Binary ~ newtreat, family=binomial(link="logit"), data=q11_data)
+         summary(m)
+
+         yes <- 1/(1+exp(-(c(1,1) %*% coef(m))))
+         no <- 1/(1+exp(-(c(1,0) %*% coef(m))))
+         atepoint <- yes-no
+
+         draw <- mvrnorm(1000, coef(m), vcov(m))
+         yesvec <- 1/(1+exp(-(draw %*% c(1,1))))
+         novec <- 1/(1+exp(-(draw %*% c(1,0))))
+         atepointvec <- yesvec-novec
+
+         #res <- rbind(res, c(atepoint, sd(atepointvec)))
+         res <- rbind(res, c(atepoint, quantile(atepointvec, 0.025), quantile(atepointvec, 0.975)))
+         count <- count+1
+     }
+     results <- rbind(results, c(etavec[i], colMedians(res)))
+ }
>
> colnames(results) <- c("eta", "pointest", "lowerci95", "upperci95")
> results <- as.data.frame(results)
>
>
> quartz(type="pdf", width=5, height=5, file="output/leadership_q11_sc1.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0), family="CMU Serif")
> plot(1:length(results$pointest), results$pointest, type="n", ylim=c(-1, 1), xlab = expression(eta), ylab="Average Treatment Effect", xaxt="n")
> polygon(c(1:length(results$pointest), rev(1:length(results$pointest))), c(results$lowerci95,rev(results$upperci95)), col="grey", border=NA)
> points(1:length(results$pointest), results$pointest, type="l", lwd=3)
> axis(1, at=seq(1, 100, length.out=6), labels = seq(0, 1, length.out=6), las=2)
> abline(h=0, col = "black", lwd=2)
> lines(c(1,1), c(results$lowerci95[1], results$upperci95[1]), lwd=3)
> points(1, results$pointest[1], pch=16)
> dev.off()
null device
          1
>
>
>
>
>
> ### Run the sensitivity analysis: Scenario (b)
>
> etavec = seq(0, 0.99, length.out=100)
>
> results <-  NULL
> for(i in 1:length(etavec)){
+     res <- NULL
+     count <- 1
+     while(count <= 500){
+         # create data given eta
+         q11_data$newtreat <- NA
+         q11_data$newtreat[q11_data$FQ11a_Discount_Bin==0 & q11_data$CQ11a_Discount_Binary==1] <- rbinom(length(q11_data$FQ11a_Discount_Bin[q11_data$FQ11a_Discount_Bin==0 & q11_data
$CQ11a_Discount_Binary==1]), 1, etavec[i])
+         q11_data$newtreat[q11_data$FQ11a_Discount_Bin==1 & q11_data$CQ11a_Discount_Binary==0] <- rbinom(length(q11_data$FQ11a_Discount_Bin[q11_data$FQ11a_Discount_Bin==1 & q11_data
```

```
$CQ11a_Discount_Binary==0]), 1, 1-etavec[i])
+         q11_data$newtreat[q11_data$FQ11a_Discount_Bin==1 & q11_data$CQ11a_Discount_Binary==1] <- q11_data$FQ11a_Discount_Bin[q11_data$FQ11a_Discount_Bin==1 & q11_data
$CQ11a_Discount_Binary==1]
+         q11_data$newtreat[q11_data$FQ11a_Discount_Bin==0 & q11_data$CQ11a_Discount_Binary==0] <- q11_data$FQ11a_Discount_Bin[q11_data$FQ11a_Discount_Bin==0 & q11_data
$CQ11a_Discount_Binary==0]
+         if(0 %in% c(table(q11_data$newtreat, q11_data$CQ11a_Discount_Binary))) next
+
+         # run model and get ATE
+         m <- glm(CQ11a_Discount_Binary ~ newtreat, family=binomial(link="logit"), data=q11_data)
+         summary(m)
+
+         yes <- 1/(1+exp(-(c(1,1) %*% coef(m))))
+         no <- 1/(1+exp(-(c(1,0) %*% coef(m))))
+         atepoint <- yes-no
+
+         draw <- mvrnorm(1000, coef(m), vcov(m))
+         yesvec <- 1/(1+exp(-(draw %*% c(1,1))))
+         novec <- 1/(1+exp(-(draw %*% c(1,0))))
+         atepointvec <- yesvec-novec
+
+         res <- rbind(res, c(atepoint, quantile(atepointvec, 0.025), quantile(atepointvec, 0.975)))
+         count <- count+1
+     }
+     results <- rbind(results, c(etavec[i], colMedians(res)))
+ }
>
> colnames(results) <- c("eta", "pointest", "lowerci95", "upperci95")
> results <- as.data.frame(results)
>
>
> quartz(type="pdf", width=5, height=5, file="output/leadership_q11_sc2.pdf")
> par(mar = c(4,4,0.3,0.3), mgp=c(2.5,1,0), family="CMU Serif")
> plot(1:length(results$pointest), results$pointest, type="n", ylim=c(-1, 1), xlab = expression(eta), ylab="Average Treatment Effect", xaxt="n")
> polygon(c(1:length(results$pointest), rev(1:length(results$pointest))), c(results$lowerci95,rev(results$upperci95)), col="grey", border=NA)
> points(1:length(results$pointest), results$pointest, type="l", lwd=3)
> axis(1, at=seq(1, 100, length.out=6), labels = seq(0, 1, length.out=6), las=2)
> abline(h=0, col = "black", lwd=2)
> lines(c(1,1), c(results$lowerci95[1], results$upperci95[1]), lwd=3)
> points(1, results$pointest[1], pch=16)
> dev.off()
null device
          1
>
```