```
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))
>
> #setwd("replication_archive/MonteCarlo/output")
> setwd("~/Dropbox/beliefs_incomplete data/Paper/PSRM/final/replication_archive/MonteCarlo/output")
>
> library(MASS)
>
> set.seed(123)
>
>
> for(pr in c(.2, .4, .6, .8)){
+     Bmult = matrix(0, 100, 5)
+     eta = seq(1,3, length = 100)
+     Bmult[,5 ] = eta
+
+     for(i in 1:length(eta)){
+         t1 <- w1 <- NULL
+         for(k in 1:100){
+             bla <- mvrnorm(1000, mu=c(3, 1), Sigma=matrix(c(2, 0.3, 0.3, 1), nrow=2))
+             t1 <- cbind(t1, bla[,1])
+             w1 <- cbind(w1, bla[,2])
+         }
+         c1 <- matrix(rbinom(100000, 1, pr), 1000,100)
+
+         BM = matrix(0, 100, 4)
+         for(j in 1:100){
+             y <- 1 + 2*t1[,j] - 3*c1[,j] + 1*w1[,j] + rnorm(1000, 0, 1)
+             mtrue = lm(y ~ t1[,j] + c1[,j] + w1[,j])
+
+             x1 <- ((1-c1[,j]) * t1[,j]) + (c1[,j]*(t1[,j]*eta[i]))
+
+             m <- lm(y ~ x1 + c1[,j] + w1[,j])
+             summary(m)
+
+             BM[j,] = m$coefficients - mtrue$coefficients
+         }
+         Bmult[i,1:4] = colMeans(abs(BM))
+     }
+
+     quartz(type="pdf", width=10, height=4, file=paste0("mcbias", pr*10,".pdf"))
+     par(mfrow = c(1,3), mar=c(4,4.5,2,1))
+
+     plot(abs(Bmult[,2]) ~ Bmult[,5], type = "l", xlab = expression(eta), ylab = expression(abs(beta[1] - hat(delta[1]))), ylim = c(0, 1.3), lwd=3)
+     plot(abs(Bmult[,3]) ~ Bmult[,5], type = "l", xlab = expression(eta), ylab = expression(abs(beta[2] - hat(delta[2]))), ylim = c(0, 6.5), lwd=3)
+     plot(abs(Bmult[,4]) ~ Bmult[,5], type = "l", xlab = expression(eta), ylab = expression(abs(beta[3] - hat(delta[3]))), ylim = c(0, 0.17), lwd=3)
+     dev.off()
+ }
>
```