Multiple Mediators

James P. Long

Simulations with multiple treatments (x) and multiple mediators (m) are demonstrated. In this work the mediators are assumed independent conditioned on treatments. There are no confounders, i.e. $c = \emptyset$

```
library(graph)
```

```
## Loading required package: BiocGenerics

## Loading required package: parallel

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind,
##     colMeans, colnames, colSums, dirname, do.call, duplicated,
##     eval, evalq, Filter, Find, get, grep, grepl, intersect,
##     is.unsorted, lapply, lengths, Map, mapply, match, mget, order,
##     paste, pmax, pmax.int, pmin, pmin.int, Position, rank, rbind,
##     Reduce, rowMeans, rownames, rowSums, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which, which.max,
##     which.min
```

```
library(kableExtra)
library(Rgraphviz)
```

```
## Loading required package: grid
```

```
library(mediateR)
set.seed(280920181)
```

Simulation 1: Linear

Simulate `n` observations from the "True Graph" using independent, binary SNPs drawn from Bernoulli(1/2) and path coefficients given below.

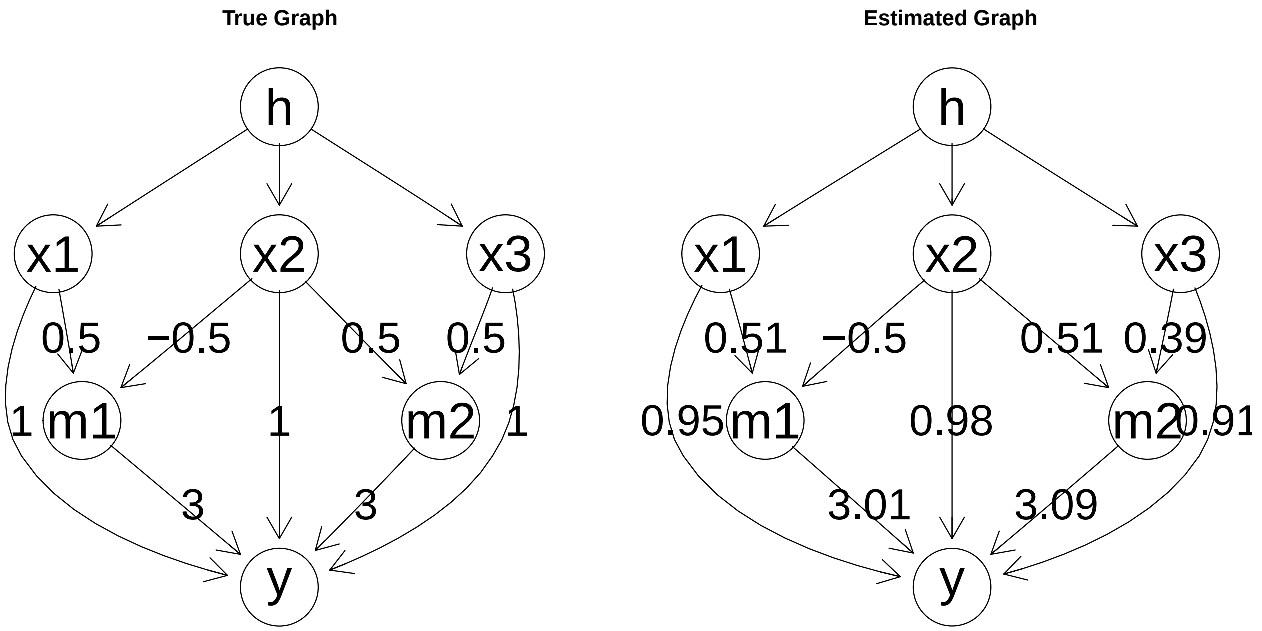
```
n <- 500
sim_params <- mediateR:::QuickSim(n,3,2,"gaussian")
dat <- SimulateData(sim_params)
```

```
gR <- mediateR:::MakeGraphNELObject(sim_params$path,sim_params$xx_direct,sim_params$mm_direct)
attrs <- list()
attrs$edge <- list()
```

```
attrs$edge$fontsize <- 12
edgeAttrs <- mediateR:::MakeedgeAttrs(sim_params$path_model,sim_params$xx_direct,sim_params$mm_direct)
```

```
fit <- ComputePath(dat)
eff_est <- ComputeEffectsLinear(fit)
```

```
gR2 <- mediateR:::MakeGraphNELObject(fit$path_model,fit$xx_direct,fit$mm_direct)
attrs2 <- list()
attrs2$edge <- list()
attrs2$edge$fontsize <- 12
edgeAttrs2 <- mediateR:::MakeedgeAttrs(fit$path_model,fit$xx_direct,fit$mm_direct)
par(mfcol=c(1,2))
plot(gR,edgeAttrs=edgeAttrs,attrs=attrs,main="True Graph")
plot(gR2,edgeAttrs=edgeAttrs2,attrs=attrs2,main="Estimated Graph")
```



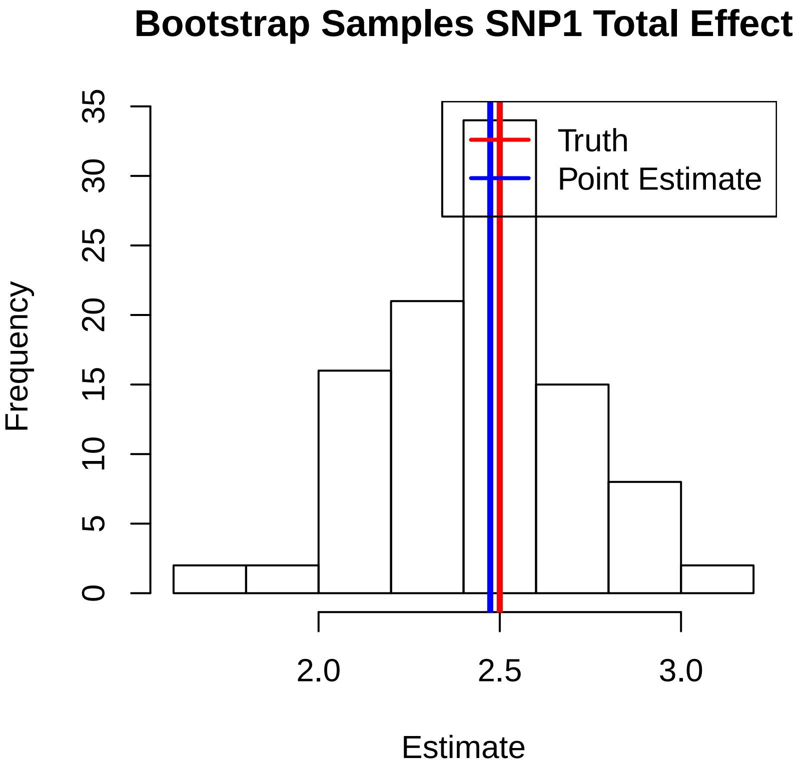
```
eff <- ComputeEffectsLinear(sim_params)
eff_comb <- matrix(0,nrow(eff),ncol=2*ncol(eff))
for(ii in 1:ncol(eff)){
  eff_comb[,2*ii-1] <- eff_est[,ii]
  eff_comb[,2*ii] <- eff[,ii]
}
for(ii in (ncol(dat$xx)+1):(ncol(dat$xx)+ncol(dat$mm))){
  eff_comb[ii,3:ncol(eff_comb)] <- NA
}
colnames(eff_comb) <- rep(c("est","true"),ncol(eff))
rownames(eff_comb) <- rownames(eff)
```

```
B <- 100
eff_est_boot <- vector("list",length=B)
for(ii in 1:B){
  ix <- sample(1:sim_params$n,replace=TRUE)
  dat_sub <- mediateR:::SubsetDat(dat,ix)
  fit <- ComputePath(dat_sub)
  eff_est_boot[[ii]] <- ComputeEffectsLinear(fit)
}
```

```
kable(eff_comb,digits=2) %>%
  kable_styling(bootstrap_options = "striped", full_width = F) %>%
  add_header_above(c(" ", "direct" = 2, "indirect" = 2, "total" = 2))
```

	direct		indirect		total	
	est	true	est	true	est	true
x1	0.95	1	1.52	1.5	2.47	2.5
x2	0.98	1	0.09	0.0	1.07	1.0
x3	0.91	1	1.19	1.5	2.10	2.5
m1	3.01	3				
m2	3.09	3				

```
total_xx1 <- vapply(eff_est_boot,function(x){x[1,3]},c(0))
hist(total_xx1,main="Bootstrap Samples SNP1 Total Effect",xlab="Estimate")
abline(v=eff[1,3],lwd=3,col='red')
abline(v=eff_est[1,3],lwd=3,col='blue')
legend("topright",c("Truth","Point Estimate"),col=c("red","blue"),lwd=2)
```



Simulation 2: Logistic

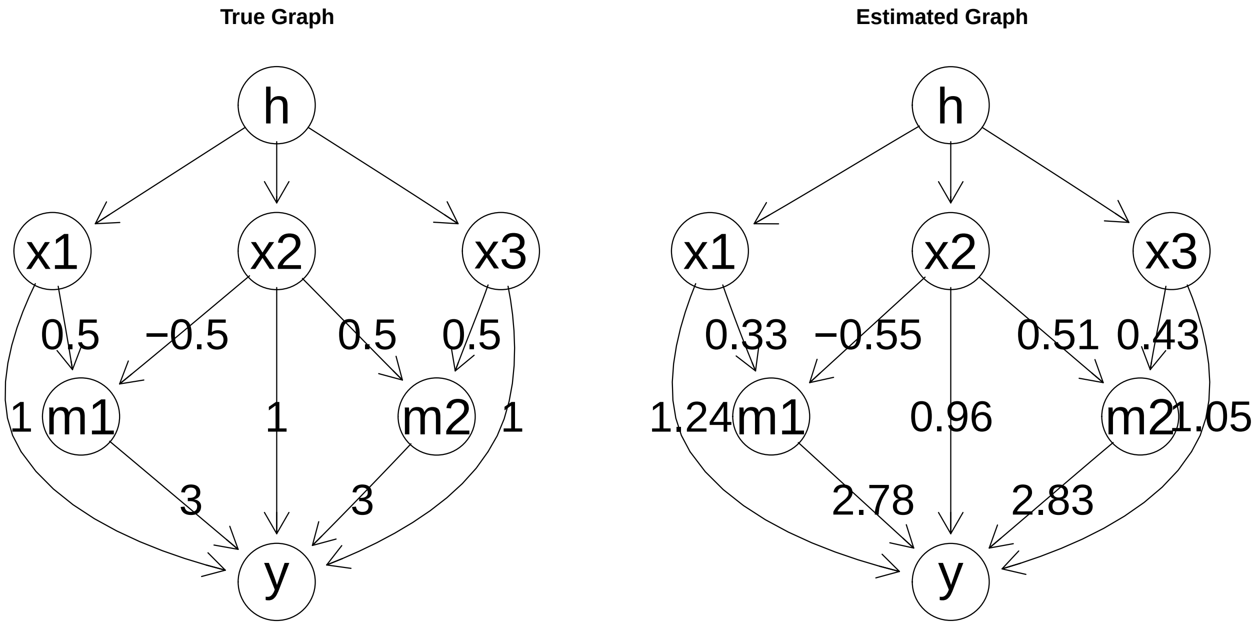
Same as simulation 1 but with binomial link function connecting y with snps and gene sets.

```
sim_params$family <- "binomial"
dat <- SimulateData(sim_params)
```

```r
gR <- mediateR:::MakeGraphNELObject(sim_params$path,sim_params$xx_direct,sim_params$mm_direct)
attrs <- list()
attrs$edge <- list()
attrs$edge$fontsize <- 12
edgeAttrs <- mediateR:::MakeedgeAttrs(sim_params$path_model,sim_params$xx_direct,sim_params$mm_direct)
```

```r
fit <- ComputePath(dat)
```

```r
gR2 <- mediateR:::MakeGraphNELObject(fit$path_model,fit$xx_direct,fit$mm_direct)
attrs2 <- list()
attrs2$edge <- list()
attrs2$edge$fontsize <- 12
edgeAttrs2 <- mediateR:::MakeedgeAttrs(fit$path_model,fit$xx_direct,fit$mm_direct)
par(mfcol=c(1,2))
plot(gR,edgeAttrs=edgeAttrs,attrs=attrs,main="True Graph")
plot(gR2,edgeAttrs=edgeAttrs2,attrs=attrs2,main="Estimated Graph")
```



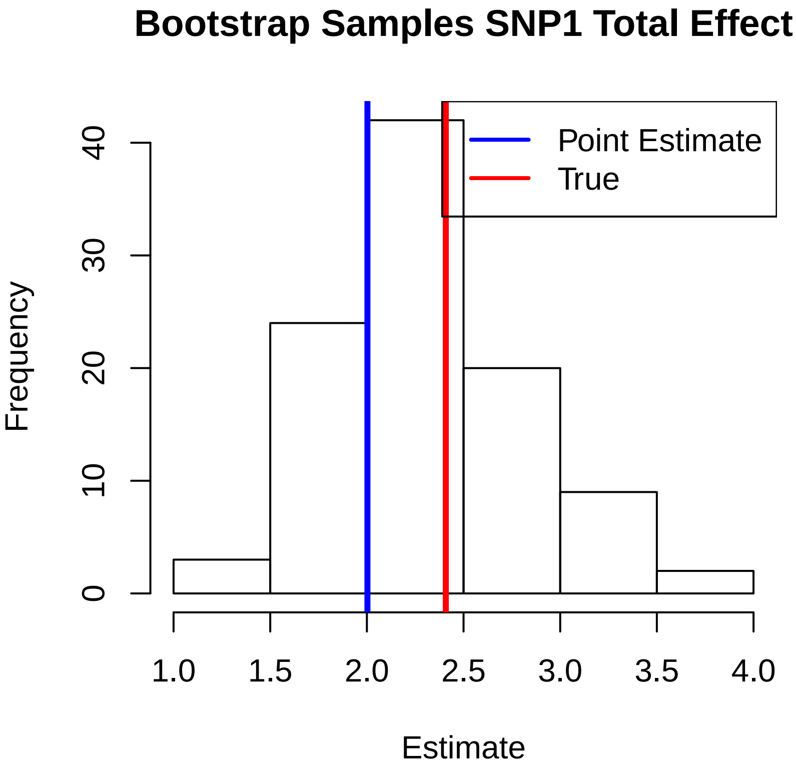
```r
## compute true effects by approximating integral with large sample size
sim_params2 <- sim_params
sim_params2$n <- 1e6 ## use million observations to approximate effects
dat2 <- SimulateData(sim_params2)
direct_t <- ComputeEffectxx(dat2,sim_params,"direct")
indirect_t <- ComputeEffectxx(dat2,sim_params,"indirect")
total_t <- ComputeEffectxx(dat2,sim_params,"total")

direct <- ComputeEffectxx(dat,fit,"direct")
indirect <- ComputeEffectxx(dat,fit,"indirect")
total <- ComputeEffectxx(dat,fit,"total")
eff_comb <- cbind(direct,direct_t,indirect,indirect_t,total,total_t)
rownames(eff_comb) <- names(fit$xx_direct)
colnames(eff_comb) <- c("est","true","est","true","est","true")
xx1total_sample <- eff_comb[1,5]
xx1total_true <- eff_comb[1,6]
```

```
B <- 100
total_xx1 <- rep(0,length=B)
for(ii in 1:B){
  ix <- sample(1:sim_params$n,replace=TRUE)
  dat_boot <- mediateR:::SubsetDat(dat,ix)
  fit_boot <- ComputePath(dat_boot)
  total_xx1[ii] <- ComputeEffectxx(dat_boot,fit_boot,"total")[1]
}
```

```
kable(eff_comb,digits=2) %>%
  kable_styling(bootstrap_options = "striped", full_width = F) %>%
  add_header_above(c(" ", "direct" = 2, "indirect" = 2, "total" = 2))
```

	direct		indirect		total	
	est	true	est	true	est	true
x1	1.61	1.40	1.33	1.7	2.00	2.41
x2	1.34	1.40	0.90	1.0	1.47	1.40
x3	1.73	1.41	1.86	1.7	2.40	2.41

```
hist(total_xx1,main="Bootstrap Samples SNP1 Total Effect",xlab="Estimate")
abline(v=xx1total_sample,lwd=3,col='blue')
abline(v=xx1total_true,lwd=3,col='red')
legend("topright",c("Point Estimate","True"),col=c("blue","red"),lwd=2)
```



Simulation 3: Survival

Same as simulation 1 and 3 but with exponentially distributed survival times. The rate is $e^{\beta^T x}$ where x are the covariates. This model can be fit with Cox Proportional Hazards.

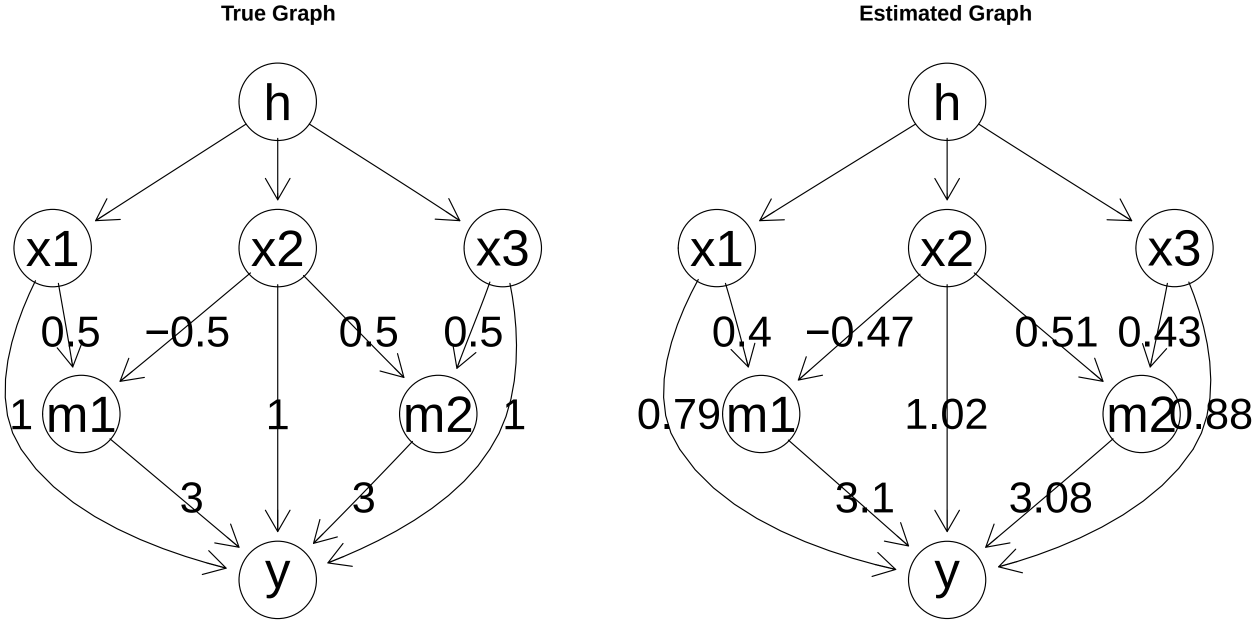
```
sim_params$family <- "cox"
dat <- SimulateData(sim_params)
rmean <- max(as.matrix(dat$y)[,1])
```

Same as simulation 1 and 3 but with exponentially distributed survival times. The rate is $e^{\beta^T x}$ where x are the covariates. This model can be fit with Cox Proportional Hazards. We compute the effects on mean survival restricted to 357.

```
gR <- mediateR:::MakeGraphNELObject(sim_params$path,sim_params$xx_direct,sim_params$mm_direct)
attrs <- list()
attrs$edge <- list()
attrs$edge$fontsize <- 12
edgeAttrs <- mediateR:::MakeedgeAttrs(sim_params$path_model,sim_params$xx_direct,sim_params$mm_direct)
```

```
fit <- ComputePath(dat)
```

```
gR2 <- mediateR:::MakeGraphNELObject(fit$path_model,fit$xx_direct,fit$mm_direct)
attrs2 <- list()
attrs2$edge <- list()
attrs2$edge$fontsize <- 12
edgeAttrs2 <- mediateR:::MakeedgeAttrs(fit$path_model,fit$xx_direct,fit$mm_direct)
par(mfcol=c(1,2))
plot(gR,edgeAttrs=edgeAttrs,attrs=attrs,main="True Graph")
plot(gR2,edgeAttrs=edgeAttrs2,attrs=attrs2,main="Estimated Graph")
```



```
## compute true effects by approximating integral with large sample size
## use the maximum observed time as the restricted mean time
sim_params2 <- sim_params
sim_params2$n <- 1e4 ## use many observations to approximate effects
dat2 <- SimulateData(sim_params2)
fit2 <- ComputePath(dat2) ## perhaps better to modify fit by entering new coefficients

direct_t <- ComputeEffectxx(dat2,fit2,"direct",rmean=rmean)
indirect_t <- ComputeEffectxx(dat2,fit2,"indirect",rmean=rmean)
total_t <- ComputeEffectxx(dat2,fit2,"total",rmean=rmean)
```

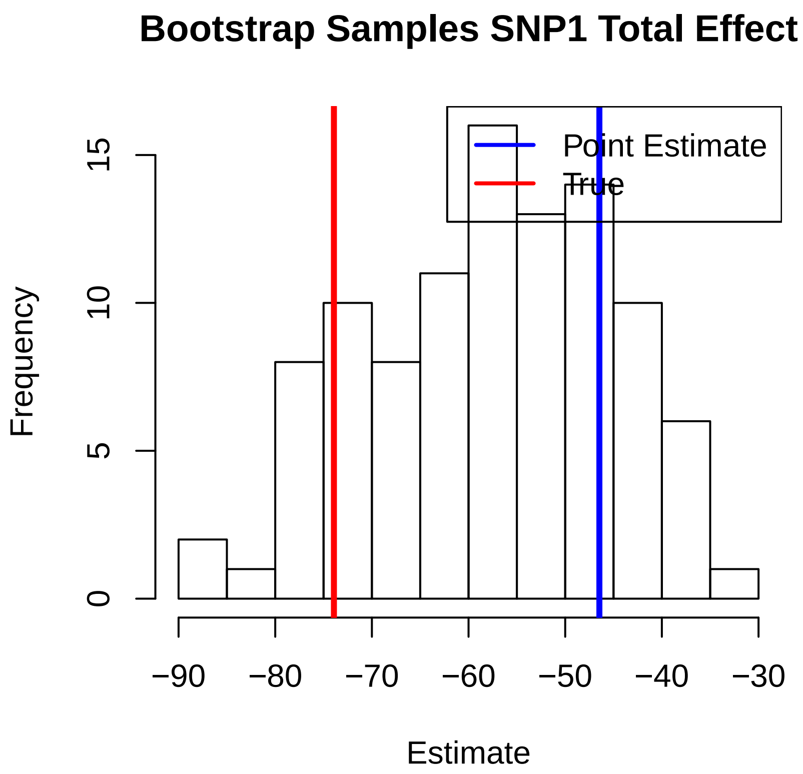
```r
direct <- ComputeEffectxx(dat,fit,"direct",rmean=rmean)
indirect <- ComputeEffectxx(dat,fit,"indirect",rmean=rmean)
total <- ComputeEffectxx(dat,fit,"total",rmean=rmean)
eff_comb <- cbind(direct,direct_t,indirect,indirect_t,total,total_t)
rownames(eff_comb) <- names(fit$xx_direct)
colnames(eff_comb) <- c("est","true","est","true","est","true")
xx1total_sample <- eff_comb[1,5]
xx1total_true <- eff_comb[1,6]
```

```r
B <- 100
total_xx1 <- rep(0,length=B)
for(ii in 1:B){
  ix <- sample(1:sim_params$n,replace=TRUE)
  dat_boot <- mediateR:::SubsetDat(dat,ix)
  fit_boot <- ComputePath(dat_boot)
  total_xx1[ii] <- ComputeEffectxx(dat_boot,fit_boot,"total",rmean=rmean)[1]
}
```

```r
kable(eff_comb,digits=2) %>%
  kable_styling(bootstrap_options = "striped", full_width = F) %>%
  add_header_above(c(" ", "direct" = 2, "indirect" = 2, "total" = 2))
```

	direct		indirect		total	
	est	true	est	true	est	true
x1	-13.75	-24.86	-48.72	-48.11	-46.46	-73.93
x2	-21.66	-27.77	9.71	2.04	-37.47	-27.46
x3	-21.31	-28.01	-15.60	-49.78	-59.35	-79.49

```r
hist(total_xx1,main="Bootstrap Samples SNP1 Total Effect",xlab="Estimate")
abline(v=xx1total_sample,lwd=3,col='blue')
abline(v=xx1total_true,lwd=3,col='red')
legend("topright",c("Point Estimate","True"),col=c("blue","red"),lwd=2)
```


Bootstrap Samples SNP1 Total Effect
Point Estimate
True
Frequency
0
5
10
15
−90
−80
−70
−60
−50
−40
−30
Estimate