solution_732A90_November2017

1.1

```
fsimNorm<-function(N){
    mTD<-matrix(runif(2*N),ncol=N,nrow=2)
    mTD[1,]<-mTD[1,]*2*pi
    two_rows <- apply(mTD,2,function(TD){Th<-TD[1];D<-TD[2];a<-sqrt(-2*log(D));c(a*sin(Th),a*cos(Th))})
    set.seed(123456)
    sample(as.vector(two_rows),N)
}


fsimGeom <- function(N,prob=0.2,k=3){

  ## inverse cdf method
  cdf_geometric <- function(prob,k){
    (1-(1-prob)^k)
  }

  cdf_geometric(prob=runif(N), k=3)
}
```

1.2

```
generate_bivariate <- function(N){

  m <- cbind(fsimNorm(N), fsimGeom(N,prob=1/3))
  colnames(m) <- c("u","v")
  return(m)
}

## bivariate sample of 10
b10 <- generate_bivariate(10)

## bivariate sample of 50
b50 <- generate_bivariate(50)

dist1 <- function(b,m){
  abs(b[1]-mean(m[,1]))+abs(b[2]-mean(m[,2]))
}

dist2 <- function(b,m){
  (b[1]-mean(m[,1]))^2+(b[2]-mean(m[,2]))^2
}


## function for first distance measure
ff1 <- function(b) { b[which.min(apply(b,1,dist1, m=b)), ] }
```

```
## function for second distance measure
ff2 <- function(b) { b[which.min(apply(b,1,dist2, m=b)), ] }

ff1(b10)
```

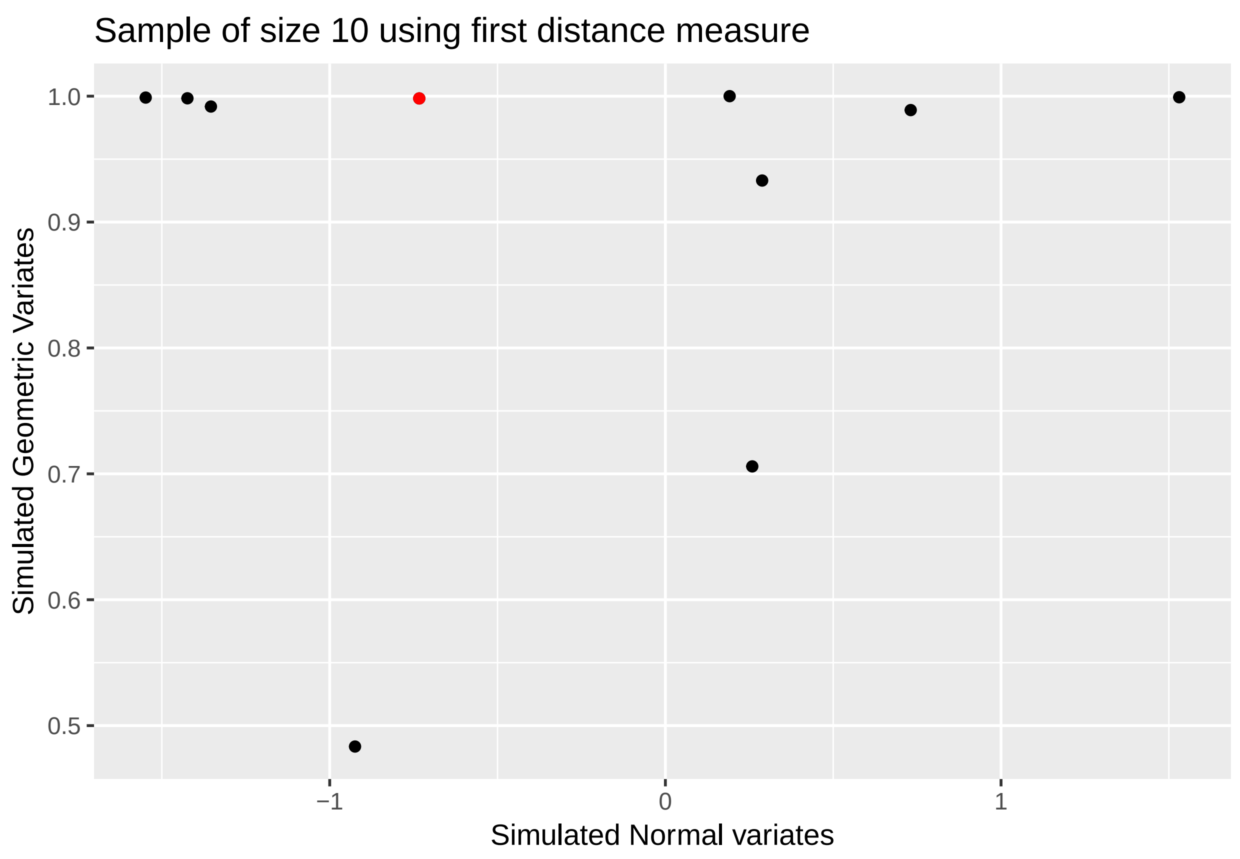
```
##          u          v
## -0.7332196  0.9982127
```

... OPTIMIZATION NOT WORKING!! WHY!!?

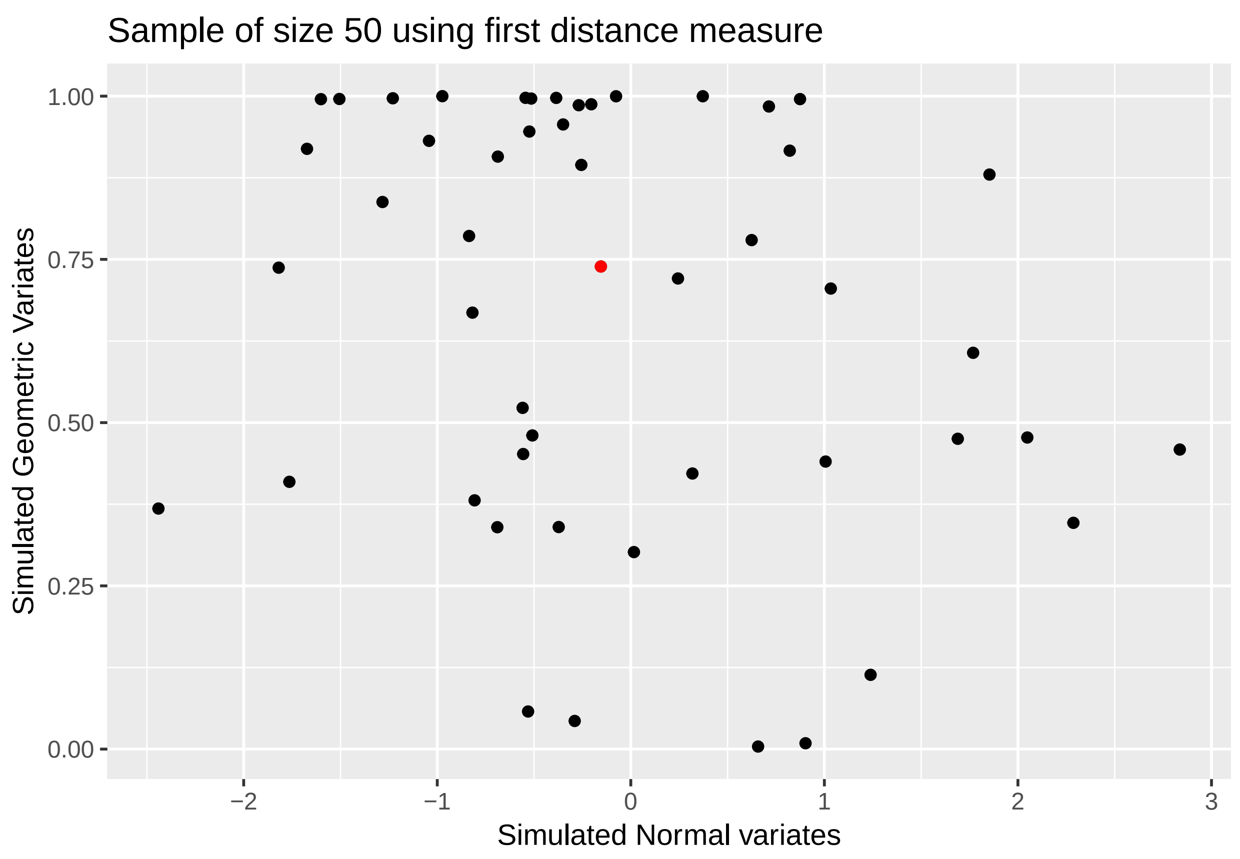
```
# ... optimising output function, generates error
# ... Error in fn(par, ...) : unused argument (par)
optim(par=c(median(b10[,1]), median(b10[,2])), fn=ff1, b=b10)

# ... optimising distance function directly
# ... Error in fn(par, ...) : unused argument (par)
optim(par=c(median(b10[,1]), median(b10[,2])), fn=dist1, b=b10, m=b10)
```

1.3

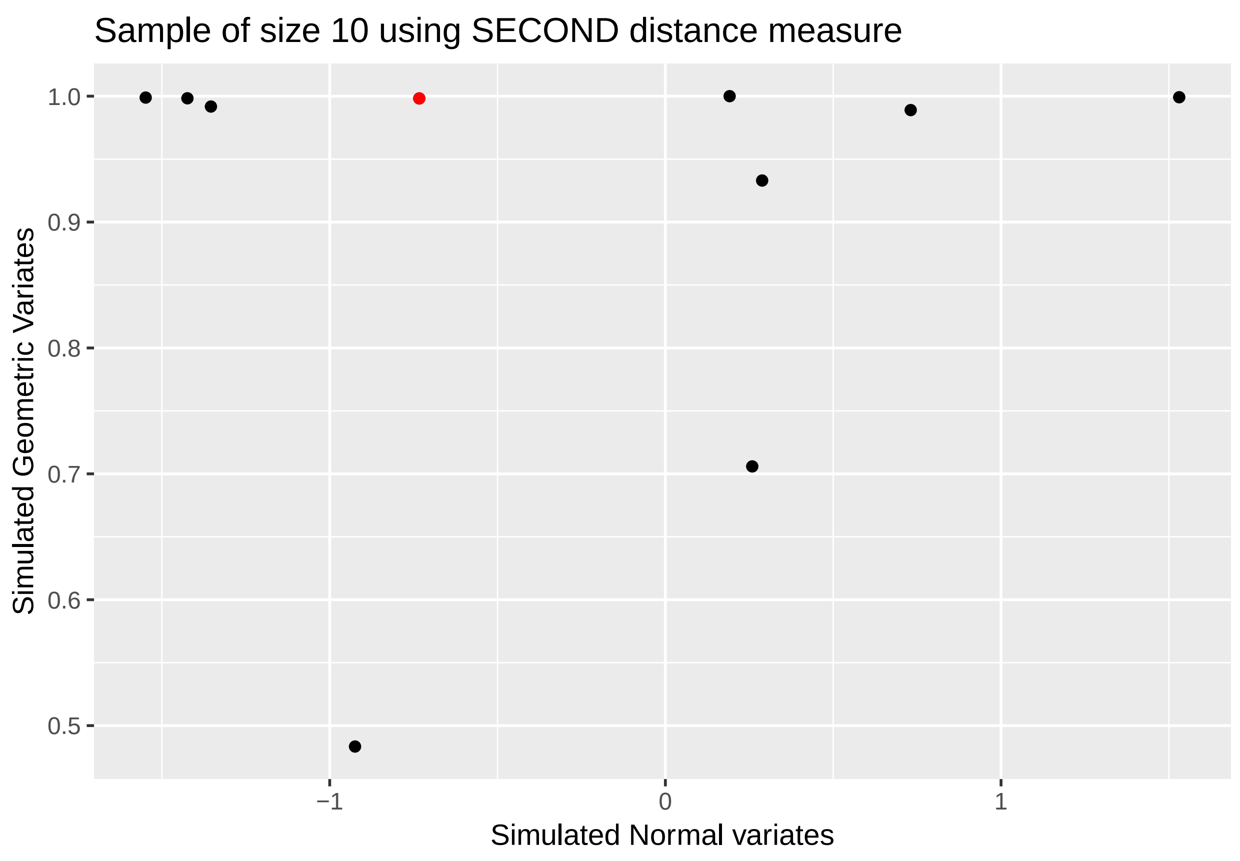
```
library(ggplot2)
ggplot()+geom_point(aes(x=b10[,1],y=b10[,2]), stat = "identity", position = "identity")+
  geom_point(aes(x=ff1(b10)[1],y=ff1(b10)[2]), stat = "identity", position = "identity", color="red")+
  labs(x="Simulated Normal variates", y="Simulated Geometric Variates",
       title = "Sample of size 10 using first distance measure")
```



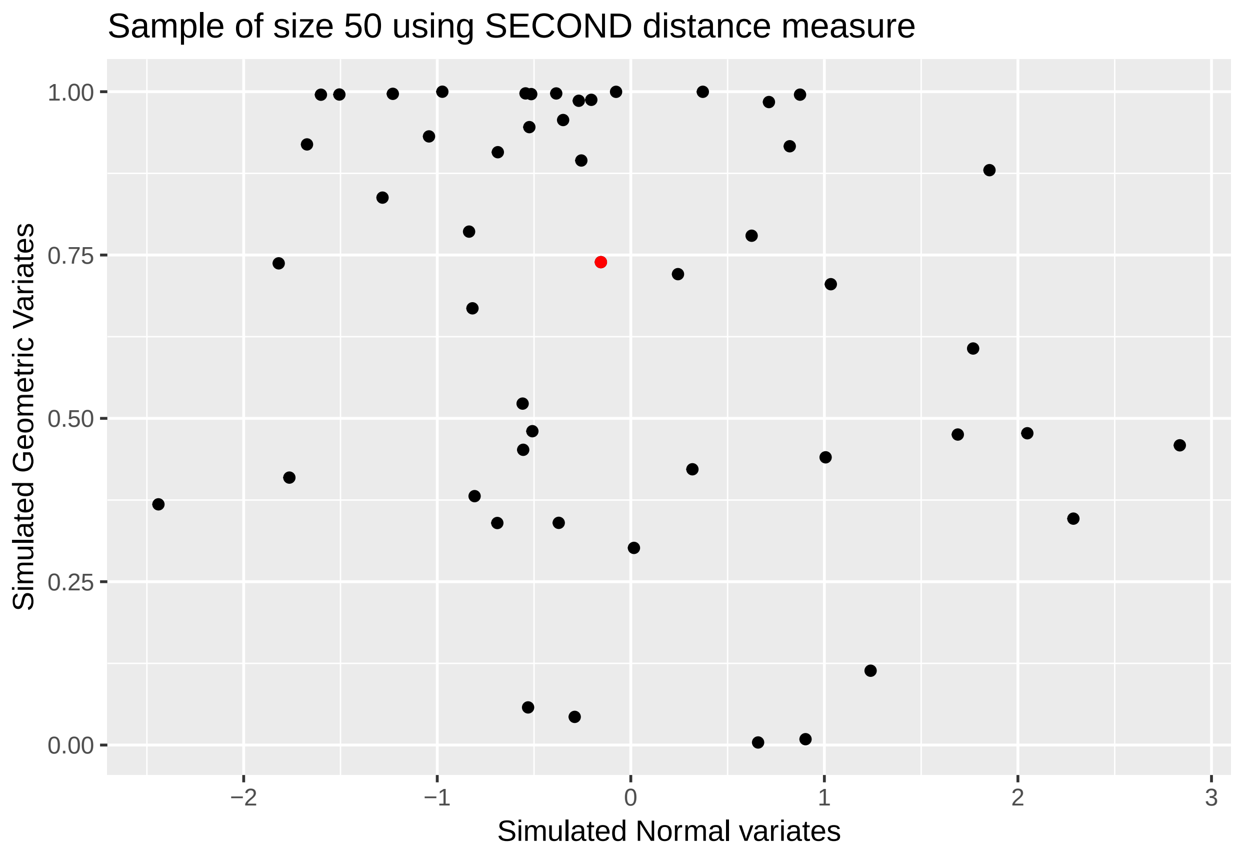
```r
ggplot()+geom_point(aes(x=b50[,1],y=b50[,2]), stat = "identity", position = "identity")+
  geom_point(aes(x=ff1(b50)[1],y=ff1(b50)[2]), stat = "identity", position = "identity", color="red")+
  labs(x="Simulated Normal variates", y="Simulated Geometric Variates",
       title = "Sample of size 50 using first distance measure")
```



```r
ggplot()+geom_point(aes(x=b10[,1],y=b10[,2]), stat = "identity", position = "identity")+
  geom_point(aes(x=ff2(b10)[1],y=ff1(b10)[2]), stat = "identity", position = "identity", color="red")+
  labs(x="Simulated Normal variates", y="Simulated Geometric Variates",
       title = "Sample of size 10 using SECOND distance measure")
```



```r
ggplot()+geom_point(aes(x=b50[,1],y=b50[,2]), stat = "identity", position = "identity")+
  geom_point(aes(x=ff2(b50)[1],y=ff1(b50)[2]), stat = "identity", position = "identity", color="red")+
  labs(x="Simulated Normal variates", y="Simulated Geometric Variates",
       title = "Sample of size 50 using SECOND distance measure")
```



Both distance measures are similarly effective.

It seems my function was designed in a way that I failed to optimize. I should have designed my function in a way that optim() could work with it.