Generating sample(part II)_Synthetic method

Weihao Lu

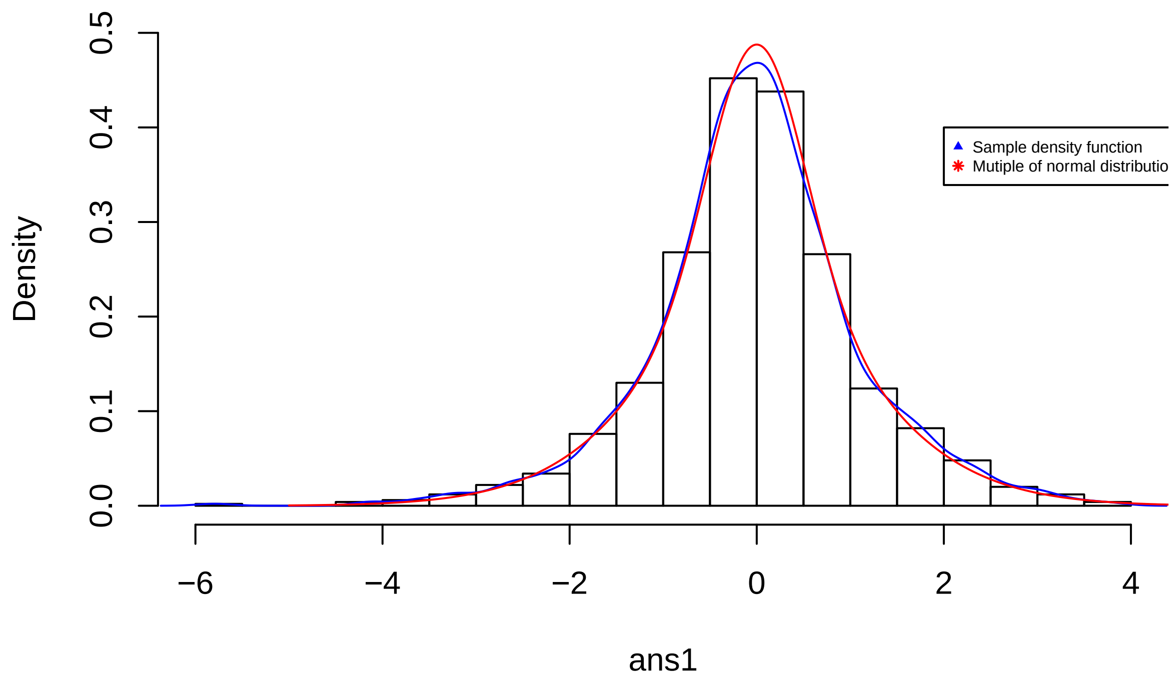
2018/3/30

Ex1: Synthetic method

The following function mfnorm generate a sample with size k, with a distribution of mutiple of normal distribution, whose density function $p(x) = \sum_{i=1}^{n} \alpha_i p_i(x)$, where $\alpha = (\alpha_1, ..., \alpha_n)$

```
mfnorm=function(k, alpha, coef){
    u=runif(k)
    ii=sapply(u, function(u, fx)sum(u>fx)+1, cumsum(alpha))
    x1=rnorm(sum(ii==1), mean=coef[1,1], sd=coef[1,2])
    x2=rnorm(sum(ii==2), mean=coef[2,1], sd=coef[2,2])
    x3=rnorm(sum(ii==3), mean=coef[3,1], sd=coef[3,2])
    return(c(x1, x2, x3))
}
```

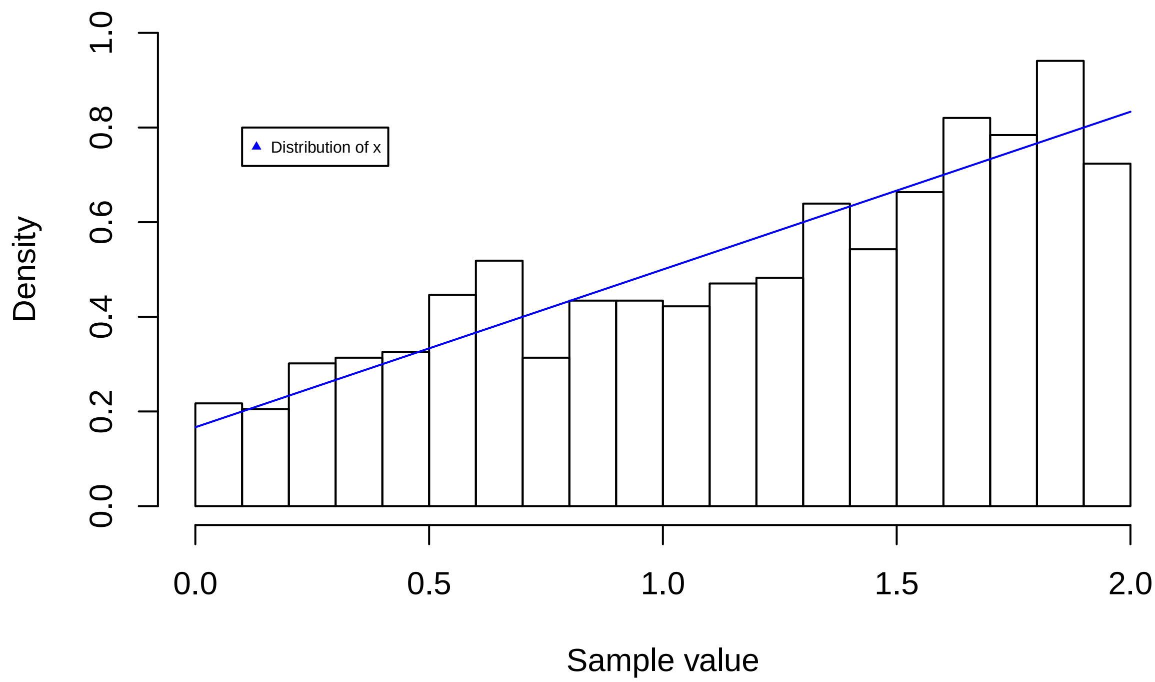
```
alpha=rep(1/3,3)
coef=matrix(c(0, 0.5,
              0, 1,
              0, 1.5),
            3, 2, byrow = TRUE)
ans1=mfnorm(1000, alpha, coef)
hist(ans1, probability = T, breaks = 24, ylim=c(0,0.5), main="")
lines(density(ans1), col="blue")
sequ=seq(-5,5,by=0.01)
px=apply(coef, 1, function(coef, sequ)dnorm(sequ,mean=coef[1],sd=coef[2]),
         sequ)
lines(sequ, apply(alpha*px, 1, sum), col="red")
legend(2, 0.4, pch=c(17,8), col=c("blue","red"), cex=0.5,
       legend = c("Sample density function", "Mutiple of normal distribution"))
```



Ex2: An example of synthetic method

```
mfsyn=function(k, alpha){
    u1=runif(k)
    ii=sapply(u1, function(u, fx)sum(u>fx)+1, cumsum(alpha))
    x1=2*runif(sum(ii==1))
    x2=2*sqrt(runif(sum(ii==2)))
    return(c(x1, x2))
}
```

```
alpha=c(1/3, 1/2)
ans2=mfsyn(1000, alpha)
hist(ans2, probability = T, breaks = 24, xlab="Sample value",
     main="", ylim=c(0,1))
lines(0:2, (1+2*(0:2))/6, col="blue")
legend(0.1, 0.8, pch=17, cex=0.5, col="blue", legend="Distribution of x")
```



Ex3: Logistic Regression Model

Known by the problem, $\pi_i = 1 - \frac{1}{1+e^{\beta x}}$, and calculated by the inverse transformation method, $y_i = 0$ when $x_i < 1 - p$
Function gdata() generates n observations (x_i, y_i)

```
gdata=function(beta, n){
    x=rnorm(n)
    pi=1-1/(1+exp(beta[1]+beta[2]*x))
    u=rnorm(n)
    return(as.numeric(!(u<1-pi)))
}
```

Function main() calls gdata() with coefficient n=1000, $(\beta_0, \beta_1) = (-0.5, 0.5)$, and draws the conclusion
Unnecessary "function"

```
main=function(){
    y=gdata(beta=c(-0.5,0.5), n=1000)
    print(table(y))
}
```

```
main()
```

```
## y
##   0   1
## 697 303
```