LAB11 activity

Yingtong Lyu

November 21, 2018

1. Write a function that generates numbers from binomial(n, p) distribution using runif() function. Hint: binomial(n, p) random variable can be defined as a sum of n independent Bernoulli(p) random variables.

```r
#use runif to generate s numbers from binomial(n,p)

set.seed(123)

n <- 10

p <- 0.5

s <- 10

x <- 0

newbinom <- function(s, n, p){

  rbi <- vector(length = s)

  for (i in 1: s) {

    rb <- as.numeric(runif(n) < p)

    rbi[i] <- sum(rb)

  }

    return(rbi)

}


newbinom(s,n,p)
```

```
##  [1] 4 5 2 5 9 6 5 6 5 6
```

```r
rbinom(s,n,p)
```

```
##  [1] 5 4 5 8 5 7 7 5 5 3
```

2. Compare performance of your function with rbinom() using microbenchmark() function.

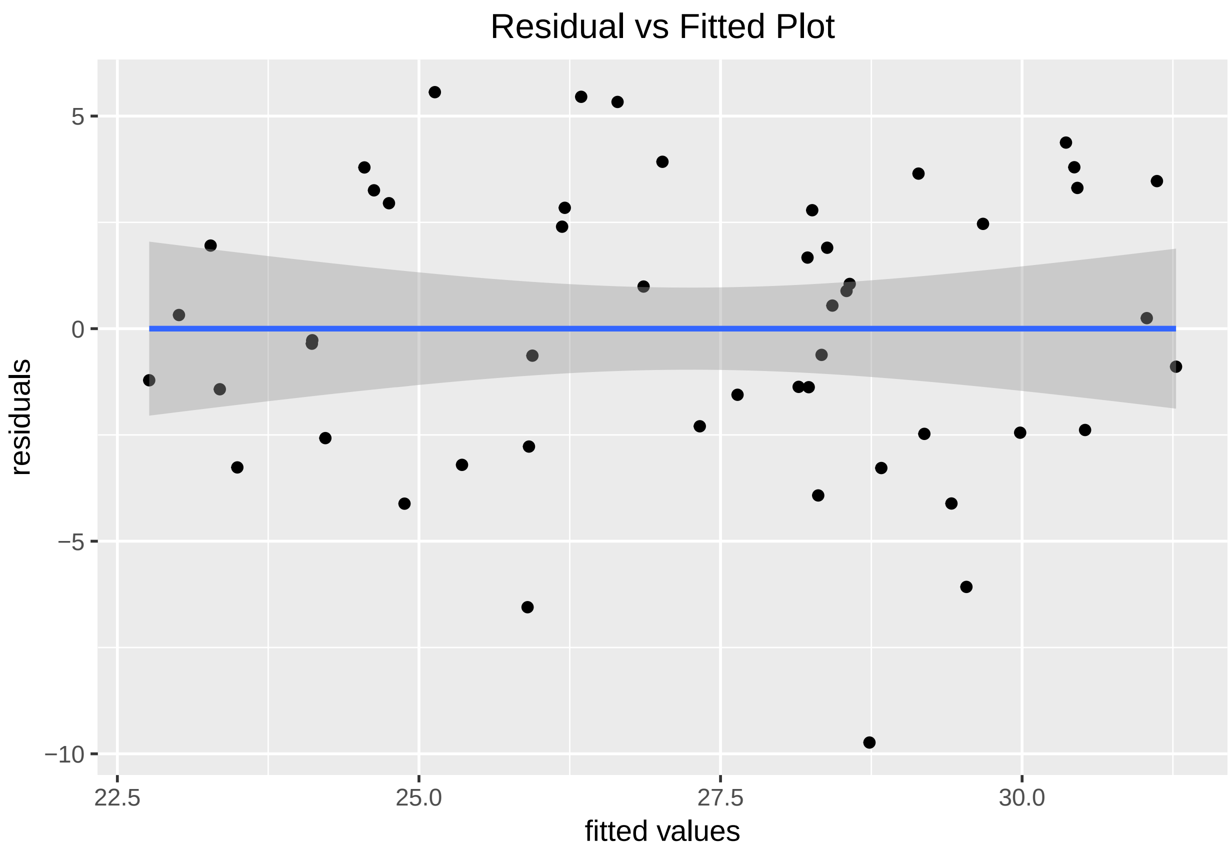
```r
library(microbenchmark)
```

```
microbenchmark(rbinom(s,n,p), newbinom(s,n,p))
```

```
## Unit: microseconds
##               expr    min      lq     mean median      uq    max neval
##     rbinom(s, n, p)  1.028  1.5420  1.86099  1.542  1.7995 12.338   100
##   newbinom(s, n, p) 18.506 19.2775 22.64933 20.562 24.6750 39.582   100
```

3.

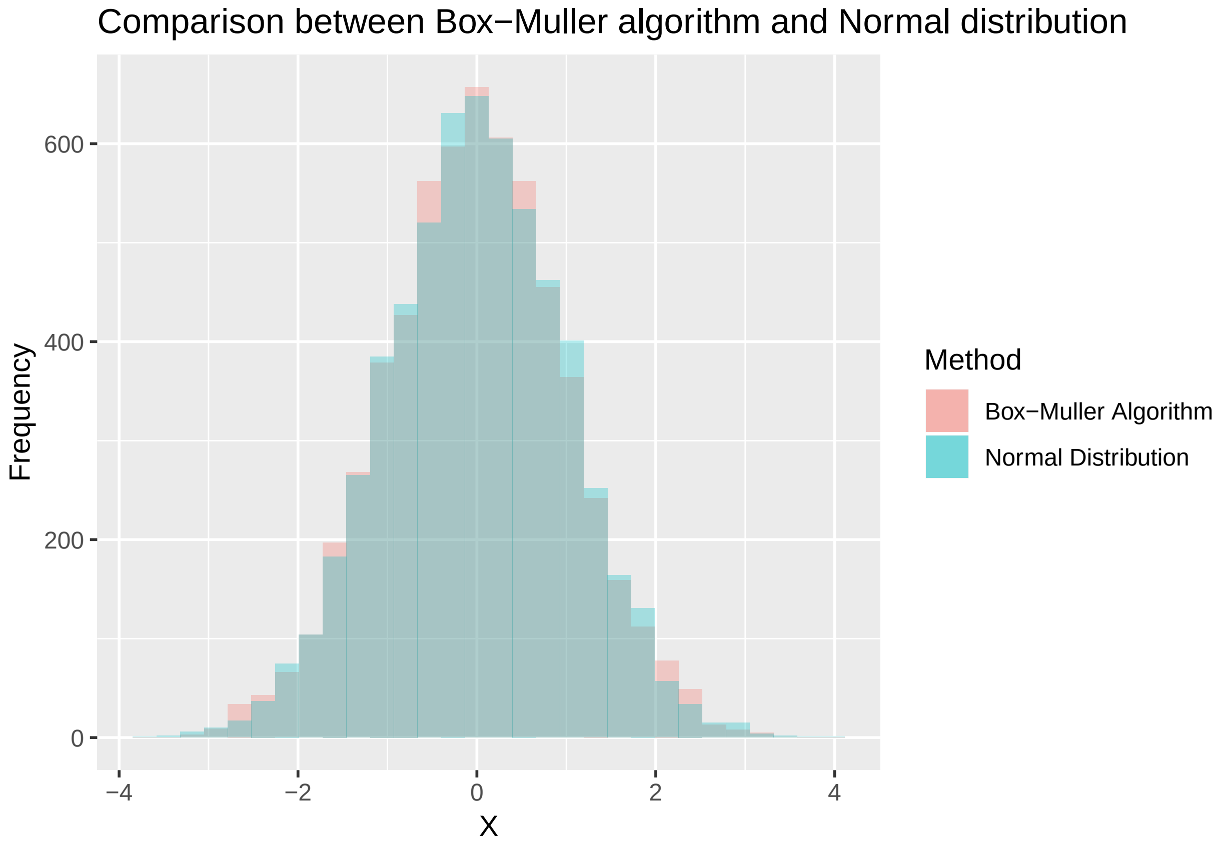
```
library(ggplot2)

# use for loop to creat data.matrix

set.seed(123)

dist <- vector(length = 50)

x <- vector(length = 50)

for (i in 1:50){

 x[i]<- runif(1,20,40)

 dist[i] <- 15 + 0.4*x[i] + rnorm(1,0,3)}

# or we could use another simple way to do that

x1 <- runif(50, 20, 40)

dist1 <- 15 + 0.4 * x + rnorm(50, 0, 3)

#built model  and plot it

fit <- lm(dist ~ x)

ggplot(fit, aes(.fitted, .resid))+

  geom_point() +

  geom_smooth(method='lm')+

  labs(x="fitted values", y= "residuals")+

  ggtitle("Residual vs Fitted Plot")+

  theme(plot.title = element_text(hjust = 0.5))
```



4.

```
creatrnorm <- function (n)

    {
        U1 <- runif(n/2, 0, 1)

        U2 <- runif(n/2, 0, 1)

        x <- sqrt(-2 * log(U1)) * cos(2 * pi * U2)

        y <- sqrt(-2 * log(U1)) * sin(2 * pi * U2)

        z <- c(x, y)

        z
}

y1 <- creatrnorm(6000)

y2 <- rnorm(6000)

Y <- data.frame(y1, y2)

ggplot(data=Y, aes(x= y1, fill = "1"))+

    geom_histogram(alpha = 0.3)+
    geom_histogram(aes(x= y2, fill = "2"), alpha = 0.3)+

    labs(x="X", y="Frequency")+

    ggtitle("Comparison between Box-Muller algorithm and Normal distribution ") +

    scale_fill_discrete(name = "Method",

                        labels=c("Box-Muller Algorithm","Normal Distribution"))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


Comparison between Box−Muller algorithm and Normal distribution
Frequency
X
Method
🟥 Box−Muller Algorithm
🟦 Normal Distribution