Lab11

Lingchen Lou

2018/11/25

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------------------- tidyverse 1.2
## <U+221A> ggplot2 3.0.0     <U+221A> purrr   0.2.5
## <U+221A> tibble  1.4.2     <U+221A> dplyr   0.7.8
## <U+221A> tidyr   0.8.1     <U+221A> stringr 1.3.1
## <U+221A> readr   1.1.1     <U+221A> forcats 0.3.0
## -- Conflicts ------------------------------------------------------------------ tidyverse_conflicts
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(microbenchmark)
library(ggplot2)
```

Question 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.

```
set.seed(1)

binom <- function(l,n,p) {
  bin_data<-NULL
  for (i in 1:l){
    x<-runif(n, min=0, max=1)
    bin_data[i] = sum(as.numeric(x < p))
  }
  return(bin_data)
}

binom(1,100, 0.4)
```

```
## [1] 38
```

```
rbinom(1,100,0.4)
```

```
## [1] 38
```

Question 2.

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

```
microbenchmark(binom(1,100,0.4),rbinom(1,100,0.4))
```

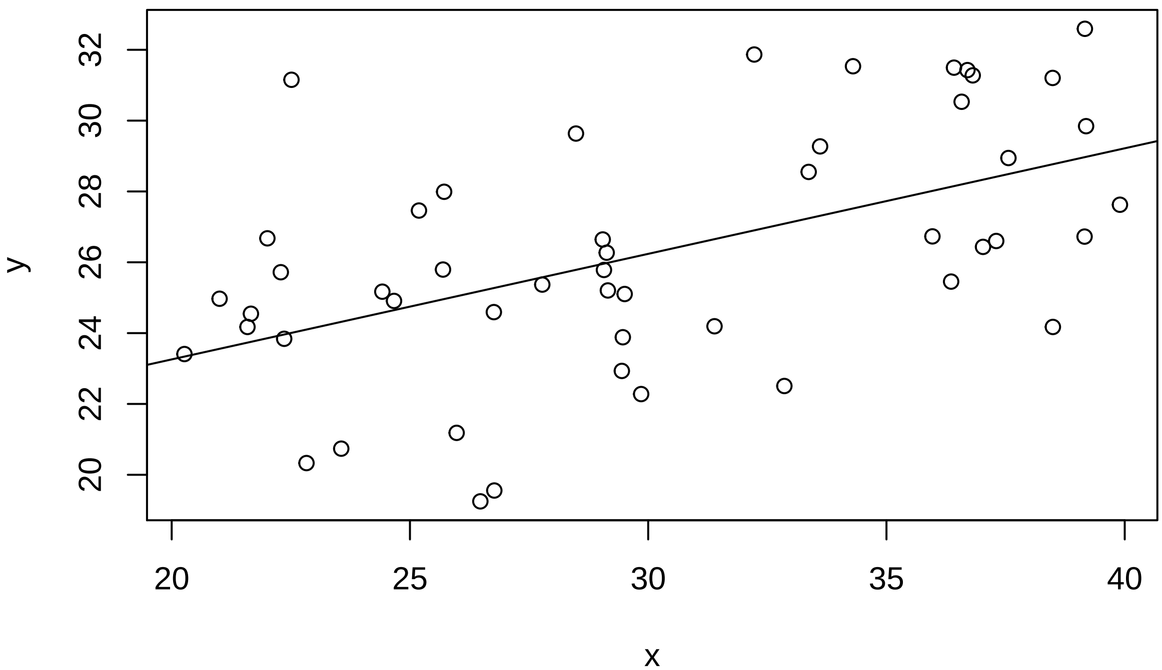
```
## Unit: microseconds
##                 expr   min    lq    mean median      uq    max neval
```

```
##   binom(1, 100, 0.4) 6.957 7.326 8.95846  7.634 10.7845 35.334   100
##  rbinom(1, 100, 0.4) 1.547 1.826 2.08644  2.023  2.2145  5.300   100
```

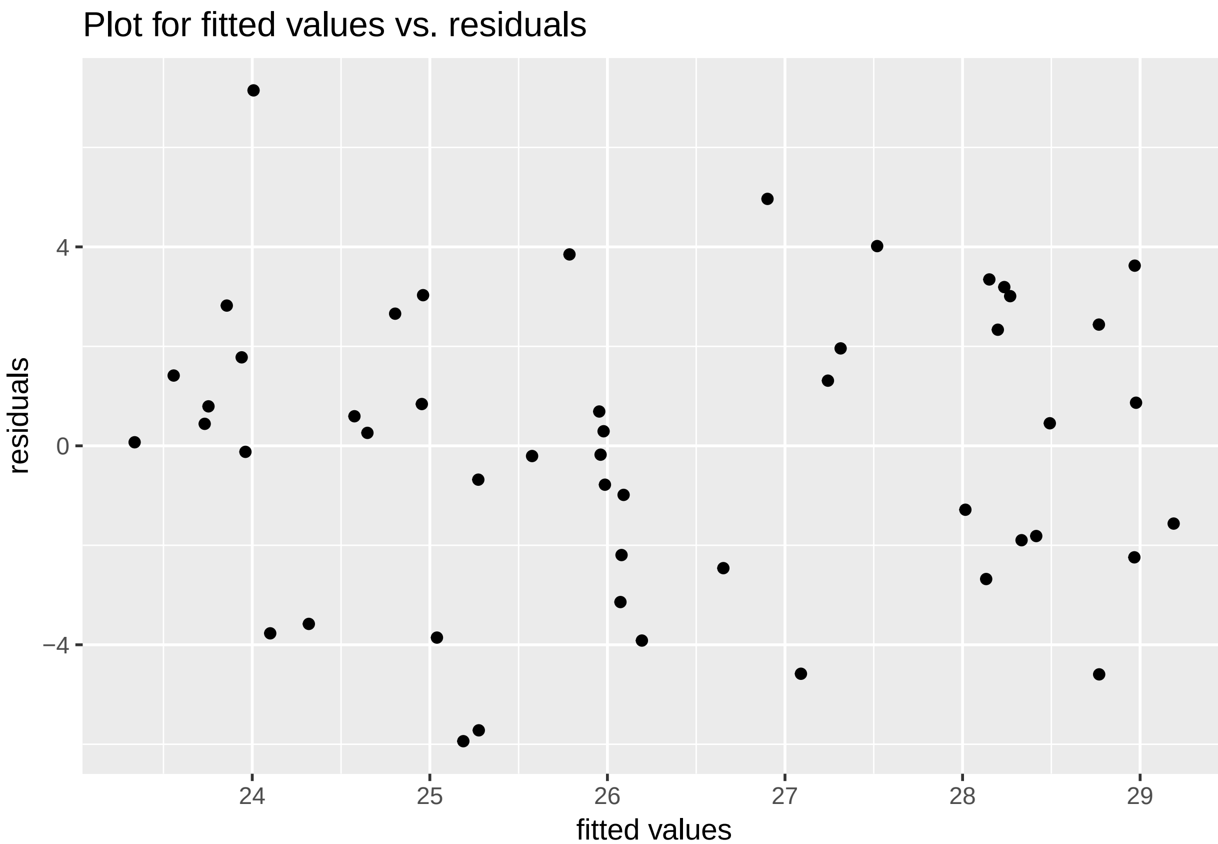
Question 3.

Suppose we want to simulate data from a linear regression model: $Y_i = \beta_0 + \beta_1 \times x_i + \epsilon_i$, $i = 1, ..., N$, where $\epsilon \sim N(0, 3)$ and X is a covariate that ranges between 20 and 40. Let $\beta_0 = 15$ and $\beta_1 = 0.4$ are known coefficients. Generate data (N = 50) from this models with given coefficients. Fit a linear regression model and plot fitted values vs residuals using ggplot() function. Please do not forget to use set.seed() function for reproducibility.

```r
set.seed(999)

n <- 50
e<-rnorm(n,0,3)
x<-runif(50,min = 20,max=40)
y<-15+0.4*x+e

mod1 <- lm(y~x)
#scatter plot
plot(x,y)
abline(lm(y~x))
```




```r
#plot fitted values vs residuals
ggplot() +
  geom_point(aes(x=mod1$fitted.values, y=mod1$residuals))+
  labs(x = "fitted values", y = "residuals", title = "Plot for fitted values vs. residuals")
```

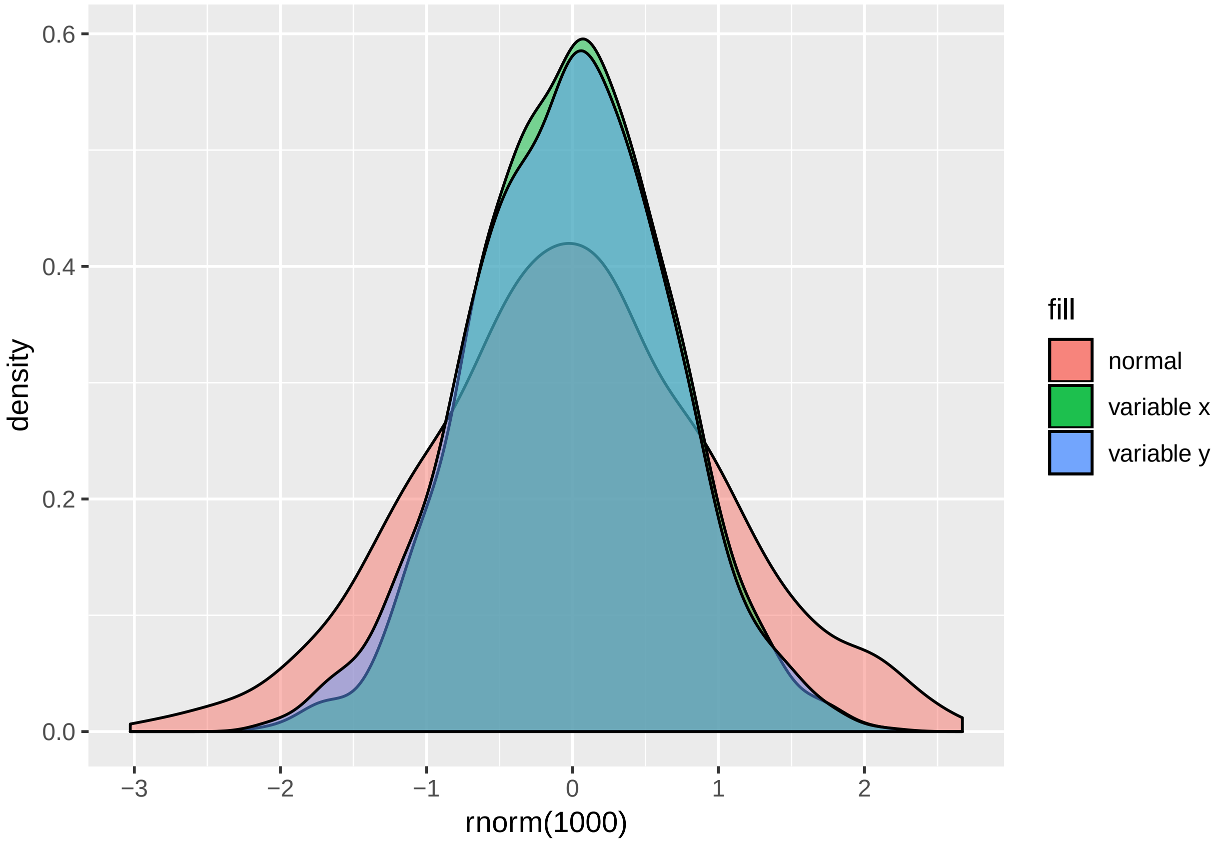


Question 4

Box-Muller algorithm: generate U_1 and U_2 two independent uniform(0, 1) random variables and set: $R = -2log(U_1)$ and $\theta = 2\pi U_2$ then $X = R\cos(\theta)$ and $Y = R\sin(\theta)$ are two independent normal variables. Write a function that generates normal variates using Box-Muller algorithm. Compare simulated data from your function with simulated data from rnorm() function using ggplot() (histogram?).

```
set.seed(999)
box_muller <- function() {
  u1 <- runif(1)
  u2 <- runif(1)

  r <- sqrt(-2*log10(u1))
  theta <- 2*pi*u2

  x <- r * cos(theta)
  y <- r * sin(theta)
  return(c(x,y))
}

data_box<-replicate(1000,box_muller())
data_box<-data.frame(x=data_box[1,],y=data_box[2,])

ggplot()+geom_density(aes(x=rnorm(1000), fill = "normal"),alpha=0.5)+
  geom_density(aes(x=data_box$x, fill = "variable x"),alpha=0.5) +
```

```
  geom_density(aes(x=data_box$y, fill = "variable y"),alpha=0.5)
```



It seems like the variable X and Y produced in Box-Muller alogrithm are normal distributed.