homework 1b

Yi Chen(yc3356)

September 6, 2018

Homework 1

Name: Yi Chen UNI: yc3556 Email: yc3556@columbia.edu (mailto:yc3556@columbia.edu)

1. Exercise 1.9 of BDA

(a)

```
### from 9 am to 4 pm there are total 420 miniutes

simulate_process <- function(times=1){
        set.seed(1)  # ensure the result wil not change
        number_of_patient <- c()
        number_of_wait <- c()
        average_wait_time <- c()
        office_time <- c()

        for(a in 1:times){

        # simulate the patient: how many patient will come into the office is indepandent to the opration condiat
ion of the office
        time <- 0 ## time that the office has been operating
        patient_time <- c()
        while(time <= 420){
                x <- rexp(n = 1,rate = 1/10)
                time <- time + x
                patient_time <- c(patient_time,x)
        }
        number_of_patient <- c(number_of_patient,length(patient_time))

        # simulate the treatment time: a patient that comes before 4 pm would be treated anyway.
        treatment_time <- c()
        for(i in 1:length(patient_time)){
                y <- runif(n = 1, min = 5, max = 20)
                treatment_time <- c(treatment_time,y)
        }


        doctor_remain_time <- c(0,0,0)
        number_o_wait <- 0
        wait_time <- 0
        operation_time <- 0
        for(i in 1:length(patient_time)){
                doctor_remain_time <- doctor_remain_time - patient_time[i]
                operation_time <- operation_time + patient_time[i]
                if(min(doctor_remain_time) <= 0){                        ## do not need to wait
                        doctor_remain_time[which(doctor_remain_time < 0)] <- 0
                        doctor_remain_time[which.min(doctor_remain_time)] <-  treatment_time[i]
```

```
                }else{                                          ## no doctor has finish the job, need to
 wait
                        number_o_wait <- number_o_wait + 1
                        wait_time <- wait_time + min(doctor_remain_time)
                        operation_time <- operation_time + min(doctor_remain_time)
                        doctor_remain_time <- doctor_remain_time - min(doctor_remain_time)
                        doctor_remain_time[which.min(doctor_remain_time)] <- treatment_time[i]
                }
        }
        average_wait <- wait_time / number_o_wait

        number_of_wait <- c(number_of_wait,number_o_wait)
        average_wait_time <- c(average_wait_time,average_wait)
        office_time <- c(office_time,operation_time)
}
        result <- list(number_of_patient,number_of_wait,average_wait_time,office_time)
        return(result)
}
```

```
problem_one <- simulate_process(1)
cat('number of patient:',problem_one[1][[1]],'\n')
```

```
## number of patient: 43
```

```
cat('number of wait:',problem_one[2][[1]],'\n')
```

```
## number of wait: 5
```

```
cat('average waiting time:',problem_one[3][[1]],'\n')
```

```
## average waiting time: 3.922496
```

```
cat('office opetation time:', problem_one[4][[1]],'\n')
```

```
## office opetation time: 442.4461
```

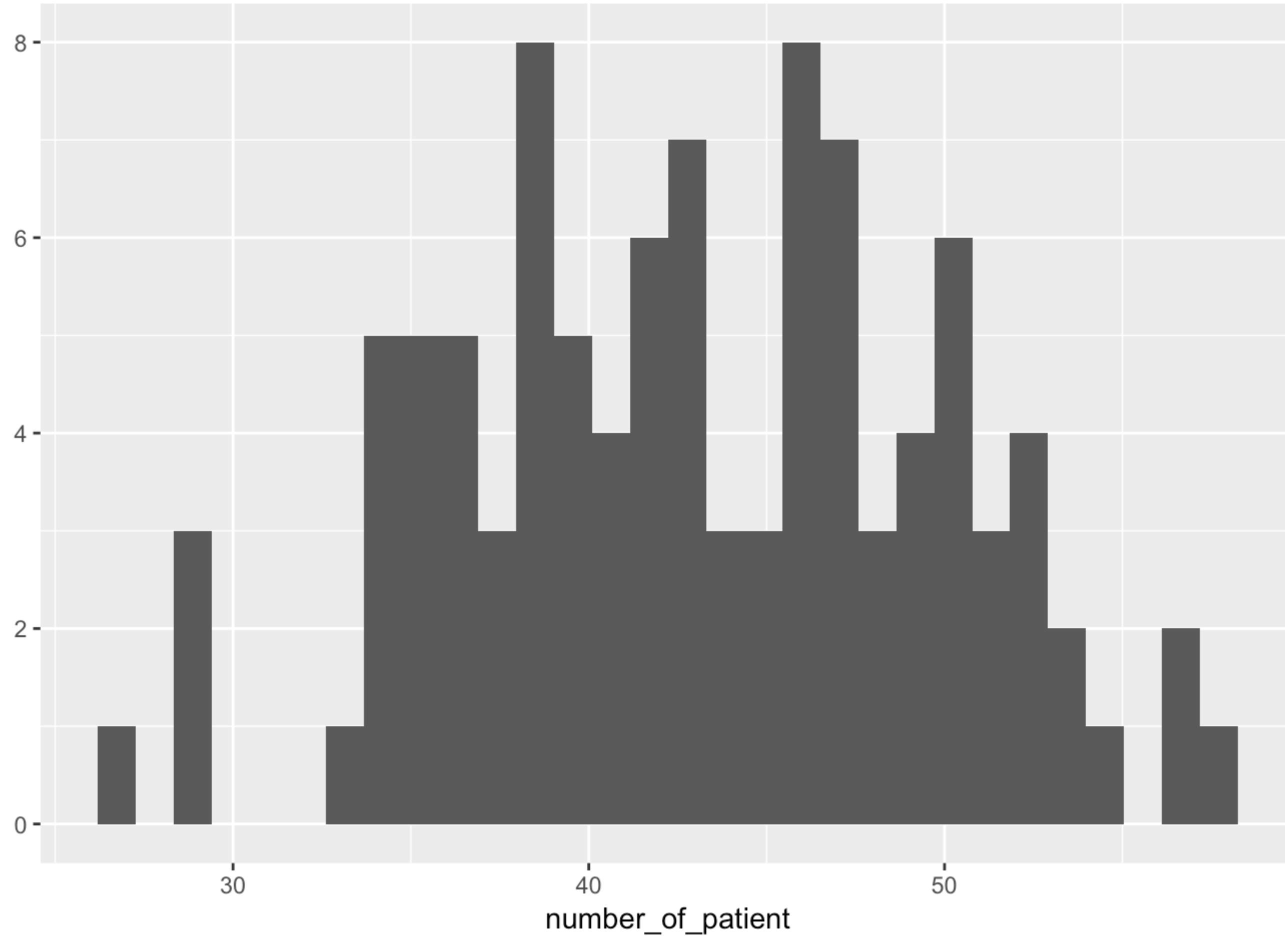
Note

The total operation time from 9:00 am is 442.45 mins which means the office really close (stop the last patient's treatment) at approximately 4:23 pm.

(b)

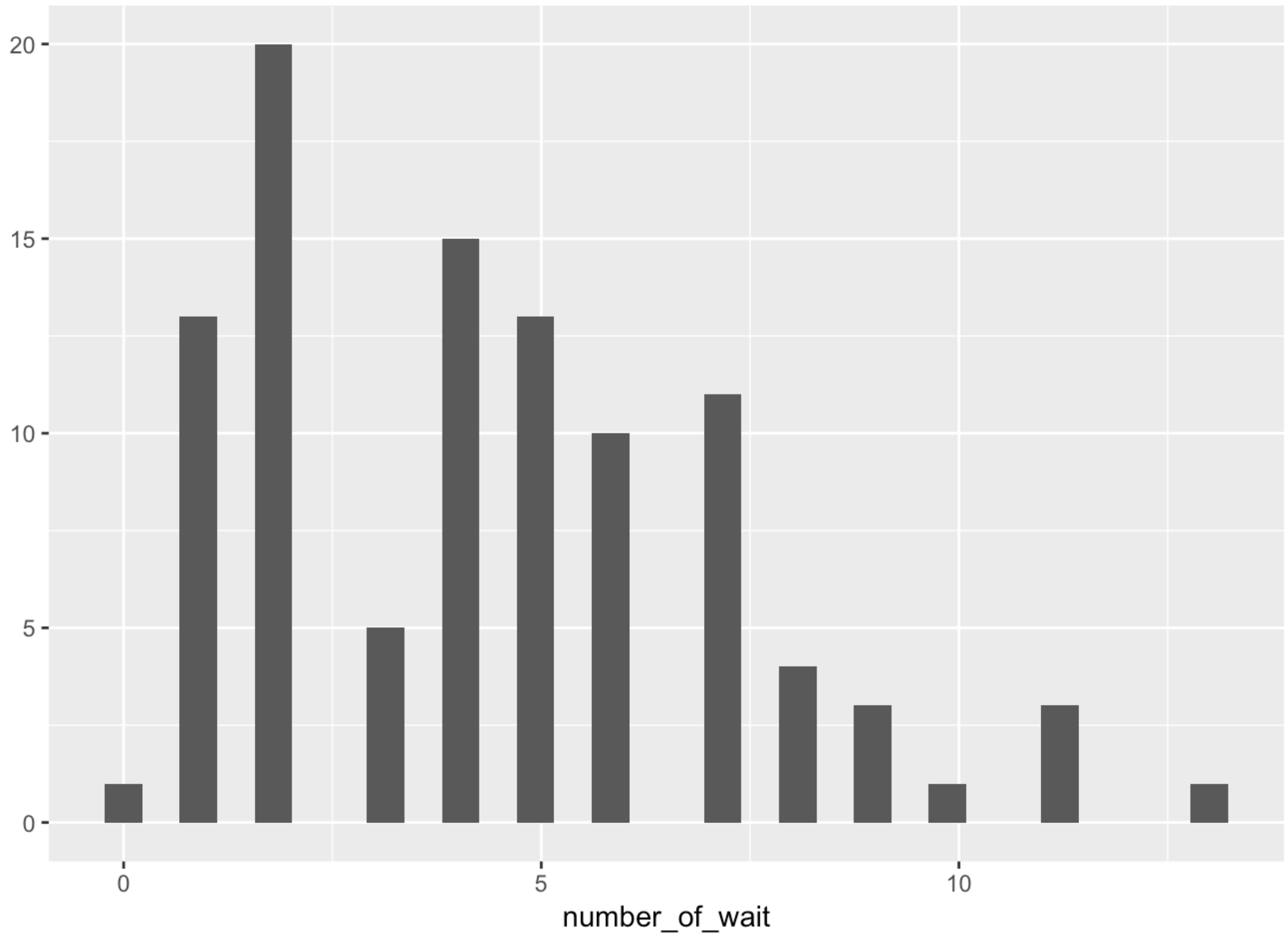
```
library('ggplot2')
number_of_patient <- simulate_process(100)[1][[1]]
number_of_wait <- simulate_process(100)[2][[1]]
average_wait_time <- simulate_process(100)[3][[1]]
office_time <- simulate_process(100)[4][[1]]
qplot(number_of_patient)
```

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




```
qplot(number_of_wait)
```

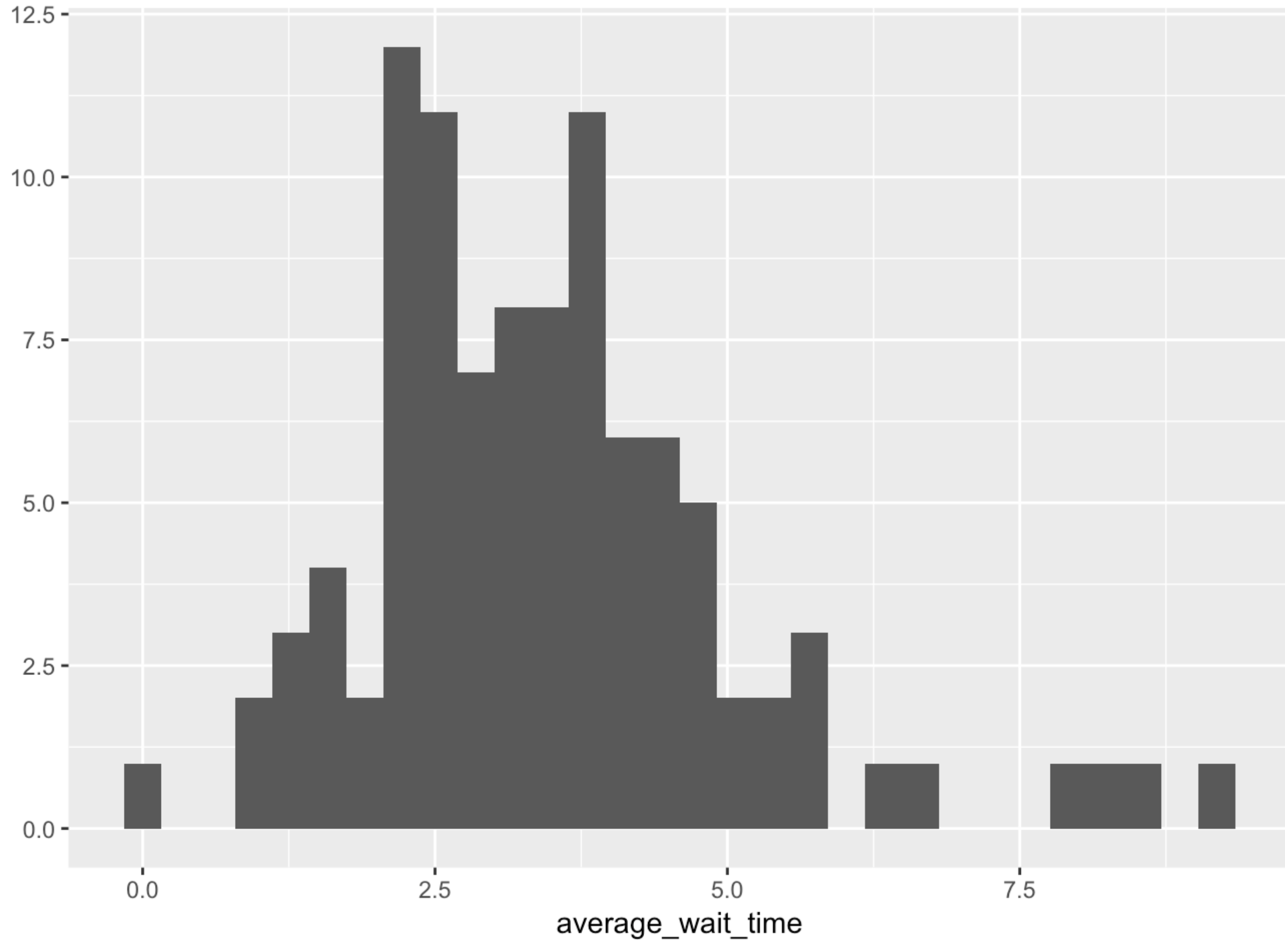
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
qplot(average_wait_time)
```

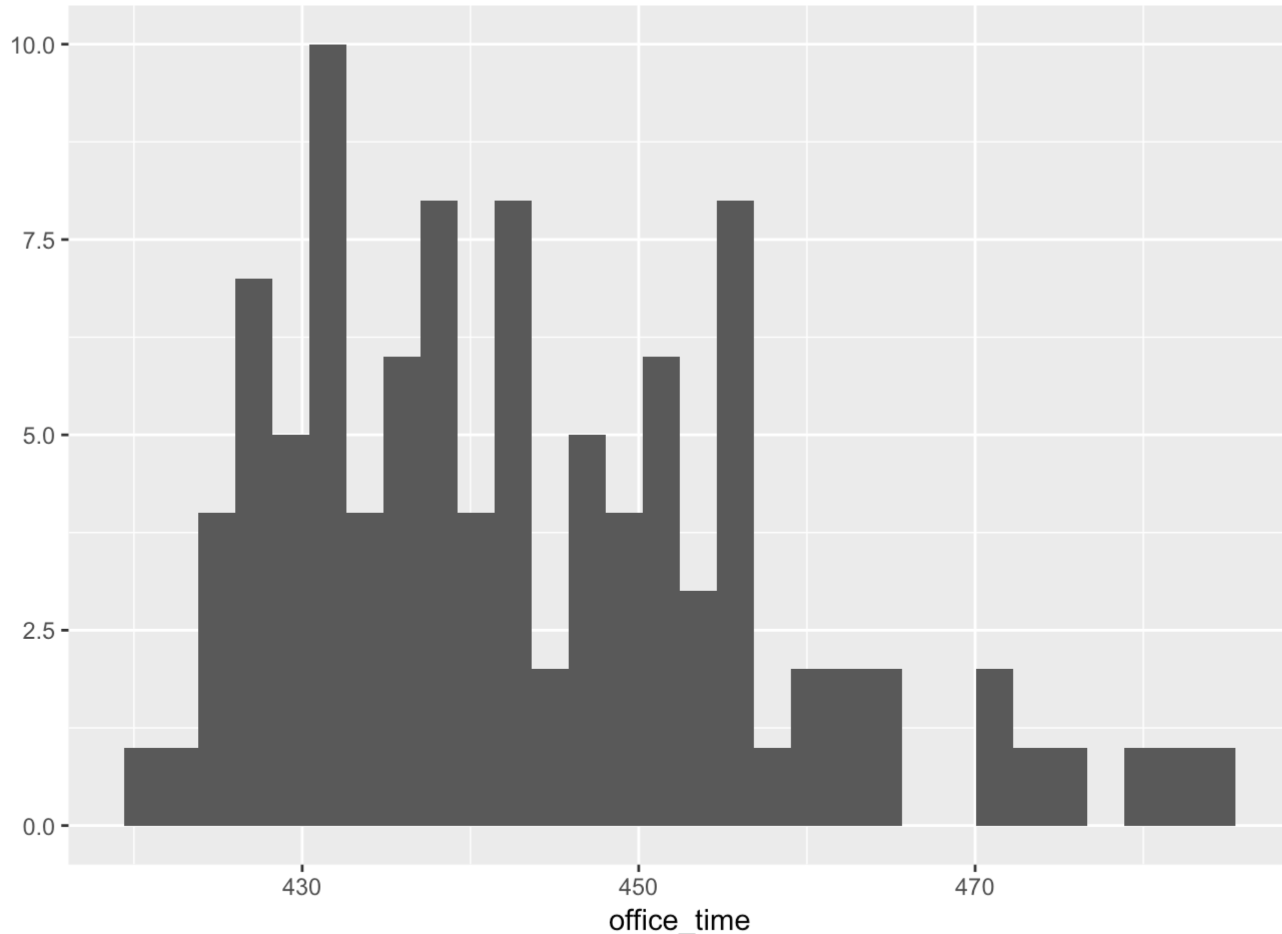
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 1 rows containing non-finite values (stat_bin).
```




```
qplot(office_time)
```

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




```
cat('median of number of patient:',median(number_of_patient),'\n')
```

```
## median of number of patient: 43
```

```
cat('median of number of wait:',median(number_of_wait),'\n')
```

```
## median of number of wait: 4
```

```
cat('median of average waiting time:',median(average_wait_time,na.rm = T),'\n')
```

```
## median of average waiting time: 3.26708
```

```
cat('median of office time:', median(office_time),'\n')
```

```
## median of office time: 441.0266
```

```
cat('the 50% interval of the number of patient is:[',quantile(number_of_patient,probs = 0.25),',',quantile(number
_of_patient,probs = 0.75),']\n')
```

```
## the 50% interval of the number of patient is:[ 38 , 48 ]
```

```
cat('the 50% interval of the number of wait is:','[',quantile(number_of_wait,probs = 0.25),',',quantile(number_of
_wait,probs = 0.75),']\n')
```

```
## the 50% interval of the number of wait is: [ 2 , 6 ]
```

```
cat('the 50% interval of the average wait time is:','[',quantile(average_wait_time,probs = 0.25,na.rm = T),',',qu
antile(average_wait_time,probs = 0.75,na.rm = T),']\n')
```

```
## the 50% interval of the average wait time is: [ 2.420231 , 4.258315 ]
```

```
cat('the 50% interval of the number of office time is:','[',quantile(office_time,probs = 0.25),',',quantile(offic
e_time,probs = 0.75),']\n')
```

```
## the 50% interval of the number of office time is: [ 431.8638 , 452.378 ]
```

Note

The medain of total office based on 100 simulation is 441.03 which means the median office close time is 4:21 pm. And the 50% interval of the office time is [431.86,452.38] which means the 50% intercal of close time is 4:12 pm to 4:32 pm.

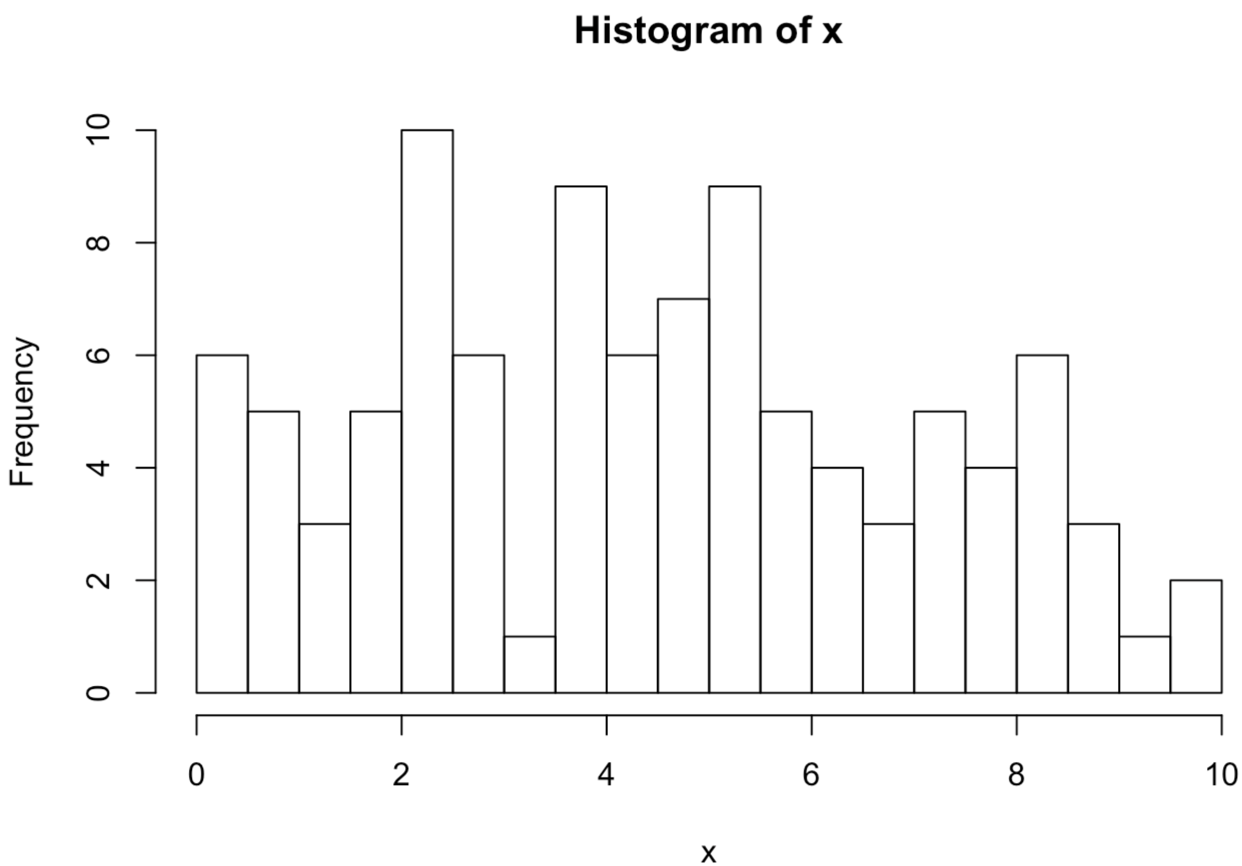
1. Fitting a simple model to simulate data

(a) stan model

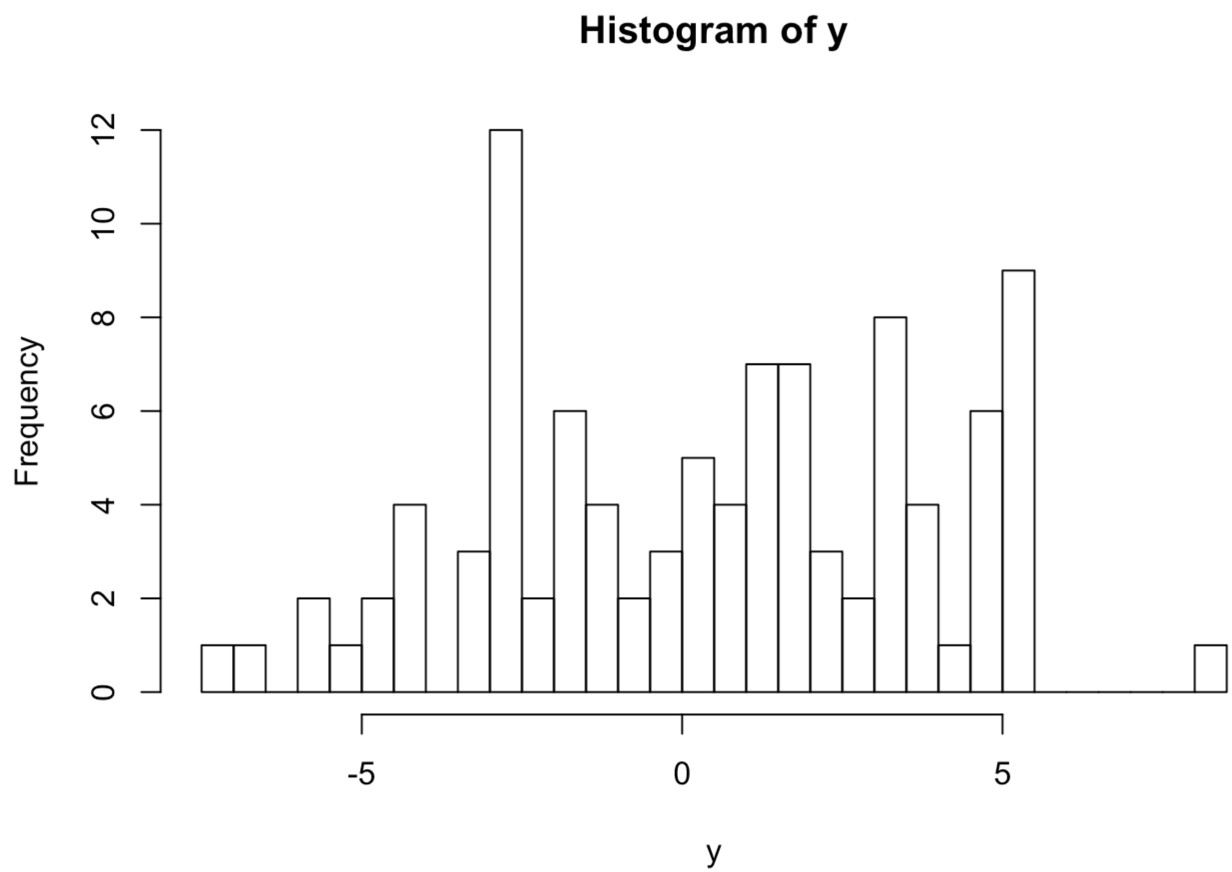
```
library("rstan")
options(mc.cores = parallel::detectCores())
rstan_options(auto_write = TRUE)
```

(b) fake data

```
set.seed(10)
N = 100
a <- 4
b <- 3
sigma <- 2
x <- runif(n = N,min = 0,max = 10)
y <- a* sin(b*x)+ rnorm(N,0,sigma)
hist(x,breaks = 30)
```




```
hist(y,breaks = 30)
```



(c) fit the model

Following is the copy of the stan file which called: ‘homework1b.stan’

data {

int N;

vector[N] y;

vector[N] x;

}

parameters {

real a;

real b;

real σ;

}

model {

y ~ normal(a *sin(b*x), σ);

}

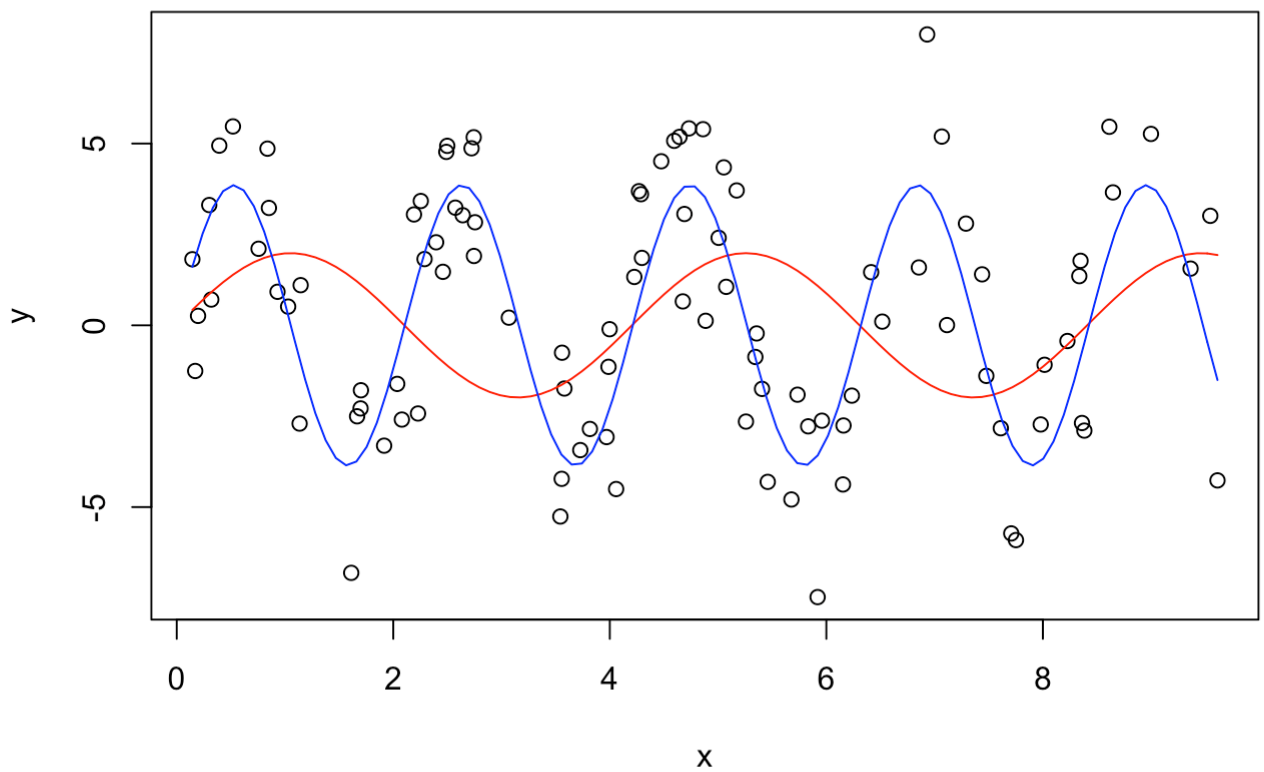
```
data = list(y=y,x=x,N=N)
fit <- stan('homework1b.stan',data = data,seed = 1,)
print(fit)
```

```
## Inference for Stan model: homework1b.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##           mean se_mean   sd    2.5%     25%     50%     75%   97.5% n_eff
## a         1.98    2.44 3.46   -4.33    1.42    3.86    4.09    4.51     2
## b         1.49    1.83 2.59   -3.00    1.47    2.98    2.99    3.01     2
## sigma     1.97    0.00 0.14    1.71    1.87    1.96    2.06    2.28  3264
## lp__   -116.25    0.03 1.23 -119.36 -116.82 -115.94 -115.34 -114.85  2176
##          Rhat
## a       13.10
## b      220.97
## sigma    1.00
## lp__     1.00
##
## Samples were drawn using NUTS(diag_e) at Fri Sep 14 08:27:56 2018.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

we can see that the true value a = 4 , b = 3, sigma = 2 are approximately recorved. Especially, the median of the estimation is very close to the true value. But, the R hat is not fit very well.

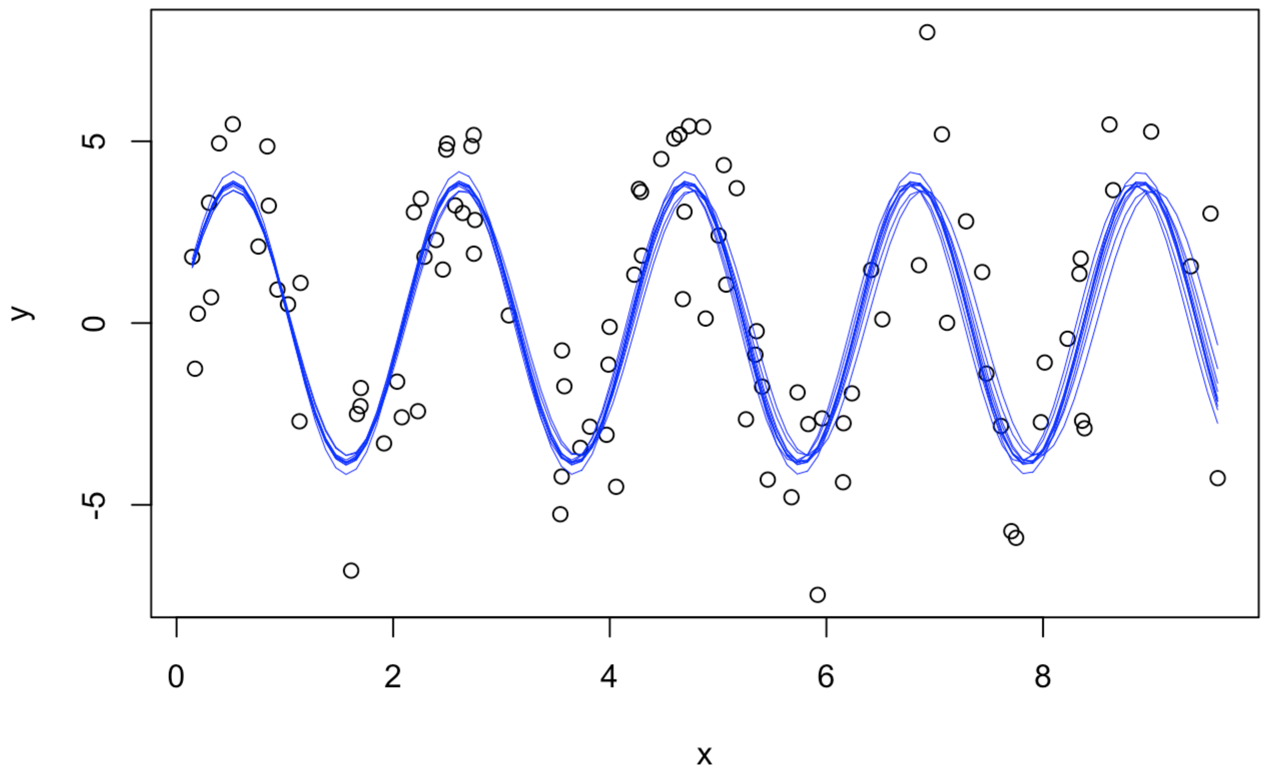
(d) simulated data and fitted model

```
## if we fit the model with mean
results <- extract(fit)
number_of_simulation <- length(results$a)
plot(x,y)
a_mean <- mean(results$a)
b_mean <- mean(results$b)
a_median <- median(results$a)
b_median <- median(results$b)
curve(a_mean*sin(b_mean*x),add = T,col='red')
curve(a_median*sin(b_median*x),add = T,col='blue')
```



Note: The red the line fitted model based on the mean and the blue line is the fitted model based on the median. Cleary, the medain is a better estimation.

```
# explore more
plot(x,y)
for(i in sample(number_of_simulation,10)){
        a_post <- results$a[i]
        b_post <- results$b[i]
        curve(a_post*sin(b_post*x),add = TRUE,col='blue',lwd=0.5)
}
```



As we can see that: the result shows that the model fit the data very well.

(e) report

There are some points that I feel confused: 1. the result of the estimation is not robust. The esitimations change a lot. 2. Why median is a better estimation all the time?

3. jitt1b

problem 2

```
jitt_1b_2 <- function(times=1){
        set.seed(1)
        result <- c()
        for(i in 1:times){
          x <- rnorm(n=1,mean = 0,sd=1)
          y <- rnorm(n=1,mean = 0,sd=1)
          if(abs(x)>2*abs(y)){
                  result <- c(result,TRUE)
          }else{
                  result <- c(result,FALSE)
          }
        }
        return(mean(result))
}
print(jitt_1b_2(100000))
```

```
## [1] 0.2936
```

problem 3

```
print(1- pnorm(-7/14))
```

```
## [1] 0.6914625
```