Assigment 4 Week 8

Viviana Romero Alarcon

```r
#Libraries
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.3     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.4.3     v tibble    3.2.1
v lubridate 1.9.2     v tidyr     1.3.0
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
library(ISLR)
library(dplyr)
library(MASS)
```

```
Attaching package: 'MASS'

The following object is masked from 'package:dplyr':

    select
```

```r
library(leaps)
```

Question 1:

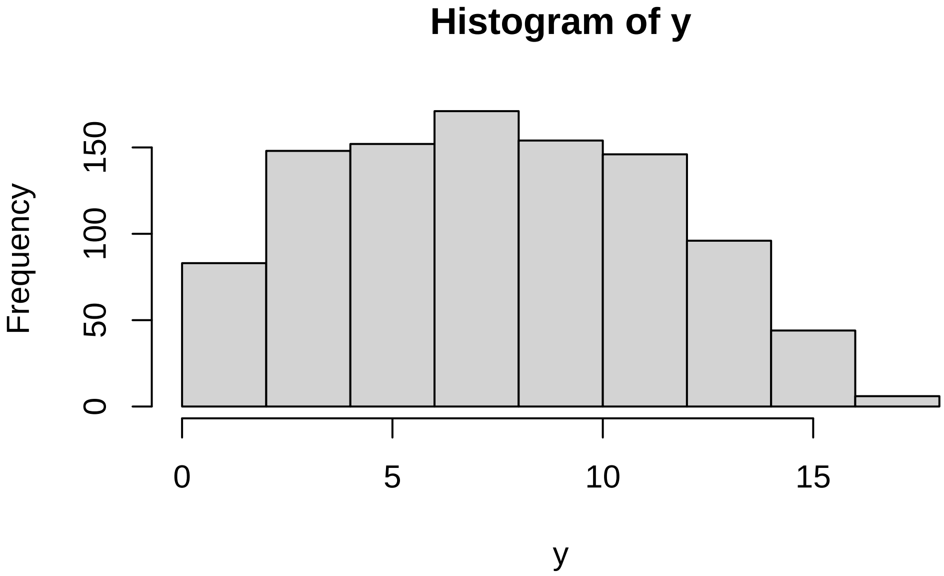
Generate a data set with $p = 20$ features, $n = 1000$ observations

```
# Set seed

set.seed(9999)

# Simulation Data

### NOTE : I could di the simulation as a multivarible rnorm, but every predictor would ha

 # n <- 1000
 # m <- 50
 # d <- lapply(1:m, function(i){
 # xy <- mvrnorm(n, c(mu.x, mu.y), S) ## one sample of size n
 #   return(data.frame(xy, sim = i))
 # })

p <- data.frame(x1 = round(abs(c(rnorm(n = 250, mean =3 ,sd = 2),rnorm(n = 250, mean = 6,s

           x2 = round(abs(c(rnorm(n = 250, mean =23 ,sd = 2),rnorm(n = 250, mean = 26,sd =

           x3 = round(abs(c(rnorm(n = 250, mean =85 ,sd = 2),rnorm(n = 250, mean = 80,sd =

           x4 = round(abs(c(rnorm(n = 250, mean =13 ,sd = 2),rnorm(n = 250, mean = 16,sd =

           x5 = round(abs(c(rnorm(n = 250, mean =45 ,sd = 2),rnorm(n = 250, mean = 40,sd =

           x6 = round(abs(c(rnorm(n = 250, mean =3 ,sd = 2),rnorm(n = 250, mean = 6,sd = 2

           x7 = round(abs(c(runif(n = 250,min = 1,max = 6 ),runif(n = 250,min = 3,max = 9

            x8 = round(abs(c(runif(n = 250,min = 113,max = 120 ),runif(n = 250,min = 115,m

            x9 = round(abs(c(runif(n = 250,min = 43,max = 56 ),runif(n = 250,min = 55,max

           x10 = round(abs(c(runif(n = 250,min = 146,max = 150 ),runif(n = 250,min = 139,m

           x11 = round(abs(c(rnorm(n = 250, mean =63 ,sd = 65),rnorm(n = 250, mean = 66,sd

           x12 = round(abs(c(rnorm(n = 250, mean =0 ,sd = 1),rnorm(n = 250, mean = 0,sd =
           x13 = round(abs(c(rnorm(n = 250, mean =80 ,sd = 4),rnorm(n = 250, mean = 0,sd =

           x14 = round(abs(c(rnorm(n = 250, mean =13 ,sd = 8),rnorm(n = 250, mean = 16,sd

           x15 = round(abs(c(rnorm(n = 250, mean =45 ,sd = 0.1),rnorm(n = 250, mean = 40,s

           x16 = round(abs(c(rnorm(n = 250, mean =3 ,sd = 0.5),rnorm(n = 250, mean = 6,sd

           x17 = round(abs(c(runif(n = 250,min = 3,max = 6 ),runif(n = 250,min = 5,max = 9

            x18 = round(abs(c(runif(n = 250,min = 33,max = 46 ),runif(n = 250,min = 45,max

            x19 = round(abs(c(runif(n = 250,min = 43,max = 56 ),runif(n = 250,min = 55,max

           x20 = round(abs(c(runif(n = 250,min = 46,max = 50 ),runif(n = 250,min = 39,max
           )


head(p)
```

```
    x1    x2    x3    x4    x5   x6   x7     x8    x9    x10    x11  x12   x13
1 5.17 22.83 84.78 12.77 44.32 2.66 1.68 115.67 55.12 147.27  68.83 0.53 83.26
2 4.69 20.52 85.84 15.10 46.38 5.96 5.22 117.37 52.20 149.54  10.75 1.64 81.88
3 3.99 21.81 84.50 12.87 43.21 1.61 4.49 115.86 47.13 148.78  23.85 1.42 81.67
4 1.45 24.12 87.44 14.55 47.31 2.88 2.31 118.77 45.39 148.85  19.04 0.23 81.01
5 8.81 23.94 85.69 14.26 40.93 5.28 2.59 114.72 53.25 146.72  83.31 1.06 82.46
6 4.82 24.92 83.73 12.33 46.13 5.67 2.23 119.36 53.42 147.92 154.60 2.07 84.09
    x14 x15  x16  x17   x18   x19   x20
1 25.68  45 2.97 4.41 34.67 47.52 48.04
2  9.58  45 2.61 5.55 43.63 54.78 46.62
3 19.07  45 3.96 3.64 41.79 49.84 49.64
4  1.98  45 3.80 5.39 42.36 55.57 46.70
5  7.74  45 2.67 4.06 40.92 53.41 47.90
6 16.98  45 2.87 5.76 45.45 55.22 49.14
```

and an associated quantitative response vector generated according to the model $Y = X\beta + \epsilon$, where β has some elements that are exactly equal to zero. Split your data set into a training data set containing $Question1:$ 100 observations and a test set containing 900 observations.

```
# Response variable  Y

y <- round(abs(c(rnorm(n = 250, mean =3 ,sd = 2),rnorm(n = 250, mean = 6,sd = 2),rnorm(n =

hist(y)
```



```
# Data set

Data <- cbind(y,p)

# Split data

Sample <- sample(x = nrow(Data), size = 100, replace = F)

training <- Data[Sample,]

test <- Data[-Sample,]
```

2. Perform best subset selection on training set and plot the training set MSE associated with the best model of each size.

```
## Perform best subset selection using leaps package
reg.fit.full = regsubsets(y~.,training, nvmax = 20)
```

```r
reg.summary = summary(reg.fit.full)

str(reg.summary)
```

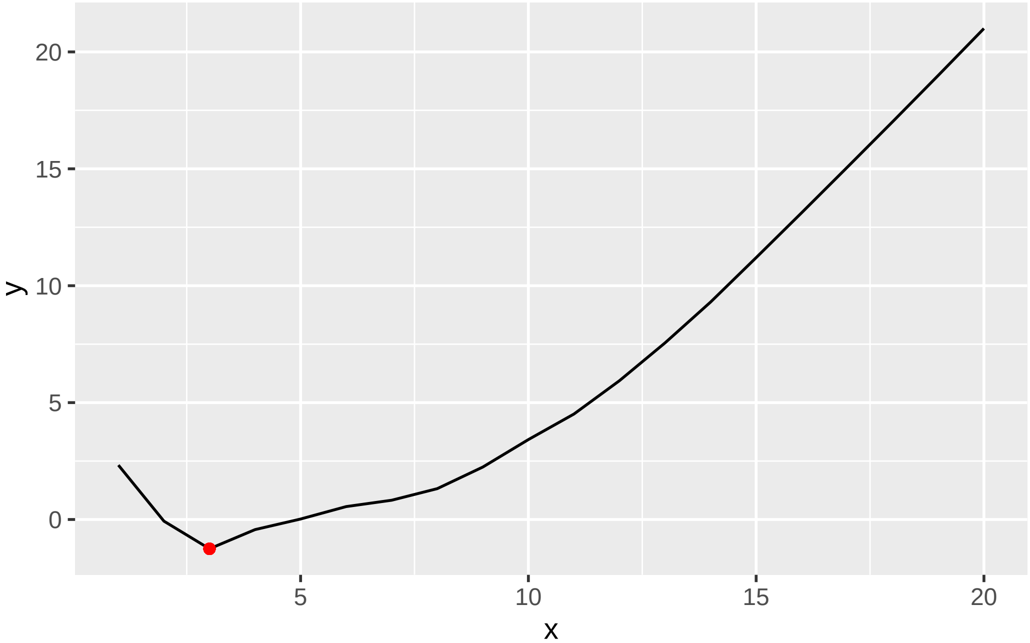
```
List of 8
 $ which : logi [1:20, 1:21] TRUE TRUE TRUE TRUE TRUE TRUE ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:20] "1" "2" "3" "4" ...
  .. ..$ : chr [1:21] "(Intercept)" "x1" "x2" "x3" ...
 $ rsq   : num [1:20] 0.735 0.747 0.755 0.759 0.763 ...
 $ rss   : num [1:20] 400 382 369 364 358 ...
 $ adjr2 : num [1:20] 0.732 0.742 0.748 0.748 0.75 ...
 $ cp    : num [1:20] 2.3255 -0.0699 -1.2525 -0.4318 0.0209 ...
 $ bic   : num [1:20] -124 -124 -122 -119 -116 ...
 $ outmat: chr [1:20, 1:20] " " " " " " " " ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:20] "1  ( 1 )" "2  ( 1 )" "3  ( 1 )" "4  ( 1 )" ...
  .. ..$ : chr [1:20] "x1" "x2" "x3" "x4" ...
 $ obj   :List of 28
  ..$ np       : int 21
  ..$ nrbar    : int 210
  ..$ d        : num [1:21] 100 3209 908 253 436 ...
  ..$ rbar     : num [1:210] 37.9 12.33 7.48 38.06 78.01 ...
  ..$ thetab   : num [1:21] 6.8081 -0.5876 0.0389 -0.0672 0.047 ...
  ..$ first    : int 2
  ..$ last     : int 21
  ..$ vorder   : int [1:21] 1 16 5 8 6 4 3 19 21 14 ...
  ..$ tol      : num [1:21] 5.00e-09 2.51e-07 8.50e-08 7.45e-08 2.45e-07 ...
  ..$ rss      : num [1:21] 1508 400 398 397 396 ...
  ..$ bound    : num [1:21] 1508 400 382 369 364 ...
  ..$ nvmax    : int 21
  ..$ ress     : num [1:21, 1] 1508 400 382 369 364 ...
  ..$ ir       : int 21
  ..$ nbest    : int 1
  ..$ lopt     : int [1:231, 1] 1 1 16 1 14 16 1 10 14 16 ...
  ..$ il       : int 231
  ..$ ier      : int 0
  ..$ xnames   : chr [1:21] "(Intercept)" "x1" "x2" "x3" ...
  ..$ method   : chr "exhaustive"
  ..$ force.in : Named logi [1:21] TRUE FALSE FALSE FALSE FALSE FALSE ...
  .. ..- attr(*, "names")= chr [1:21] "" "x1" "x2" "x3" ...
  ..$ force.out: Named logi [1:21] FALSE FALSE FALSE FALSE FALSE FALSE ...
  .. ..- attr(*, "names")= chr [1:21] "" "x1" "x2" "x3" ...
  ..$ sserr    : num 321
  ..$ intercept: logi TRUE
  ..$ lindep   : logi [1:21] FALSE FALSE FALSE FALSE FALSE FALSE ...
  ..$ nullrss  : num 1508
  ..$ nn       : int 100
  ..$ call     : language regsubsets.formula(y ~ ., training, nvmax = 20)
  ..- attr(*, "class")= chr "regsubsets"
 - attr(*, "class")= chr "summary.regsubsets"
```

1. plot AIC (or Cp) for the best model of each size.

```
plotCP <-data.frame(x = 1:length(reg.summary$cp), y = reg.summary$cp)

which.min(reg.summary$cp)
```

```
[1] 3
```

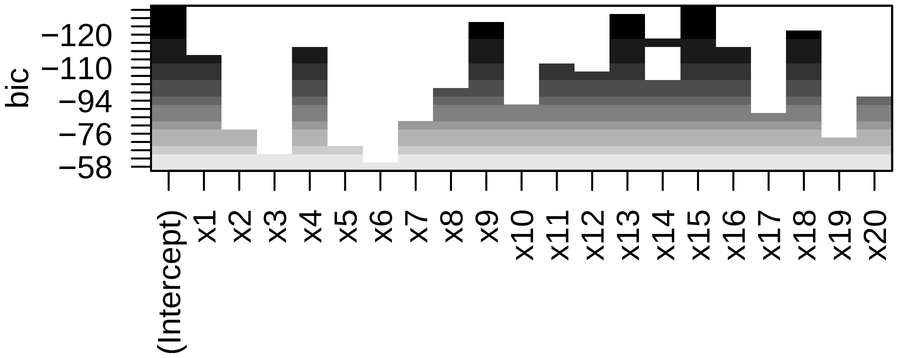
```
ggplot(plotCP, aes(x = x, y = y) )+
  geom_line()+
  geom_point( aes(x = x[which.min(reg.summary$cp)], y = y[which.min(reg.summary$cp)]), col
```



```
str(reg.fit.full)
```

```
List of 28
 $ np       : int 21
 $ nrbar    : int 210
 $ d        : num [1:21] 100 3209 908 253 436 ...
 $ rbar     : num [1:210] 37.9 12.33 7.48 38.06 78.01 ...
 $ thetab   : num [1:21] 6.8081 -0.5876 0.0389 -0.0672 0.047 ...
 $ first    : int 2
 $ last     : int 21
 $ vorder   : int [1:21] 1 16 5 8 6 4 3 19 21 14 ...
 $ tol      : num [1:21] 5.00e-09 2.51e-07 8.50e-08 7.45e-08 2.45e-07 ...
 $ rss      : num [1:21] 1508 400 398 397 396 ...
 $ bound    : num [1:21] 1508 400 382 369 364 ...
 $ nvmax    : int 21
 $ ress     : num [1:21, 1] 1508 400 382 369 364 ...
 $ ir       : int 21
 $ nbest    : int 1
 $ lopt     : int [1:231, 1] 1 1 16 1 14 16 1 10 14 16 ...
 $ il       : int 231
 $ ier      : int 0
 $ xnames   : chr [1:21] "(Intercept)" "x1" "x2" "x3" ...
 $ method   : chr "exhaustive"
 $ force.in : Named logi [1:21] TRUE FALSE FALSE FALSE FALSE FALSE ...
  ..- attr(*, "names")= chr [1:21] "" "x1" "x2" "x3" ...
 $ force.out: Named logi [1:21] FALSE FALSE FALSE FALSE FALSE FALSE ...
  ..- attr(*, "names")= chr [1:21] "" "x1" "x2" "x3" ...
 $ sserr    : num 321
 $ intercept: logi TRUE
 $ lindep   : logi [1:21] FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ nullrss  : num 1508
 $ nn       : int 100
 $ call     : language regsubsets.formula(y ~ ., training, nvmax = 20)
 - attr(*, "class")= chr "regsubsets"
```

```
par(mfrow = c(1,1))
plot(reg.fit.full, scale = "bic")
```



```
coef(reg.fit.full, 6)
```

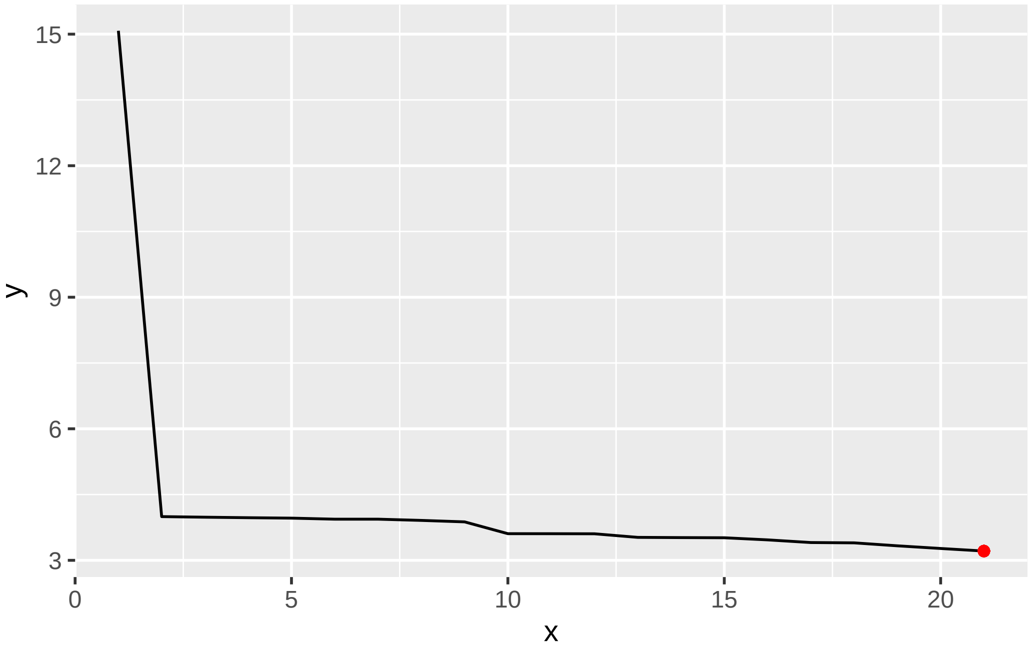
```
(Intercept)          x4          x9         x13         x15         x16 
49.11783841  0.09273467 -0.03787563  0.02988362 -1.07192855 -0.60069415 
        x18 
 0.06112453 
```

2. Plot the test MSE associated with the best model of each size. For which model size does the test MSE takes on its minimum value?

```
MSE <- reg.summary$obj$rss/100

plotMSE <-data.frame(x = 1:length(MSE), y = MSE)

which.min(MSE)
```

```
[1] 21
```

```
ggplot(plotMSE, aes(x = x, y = y) )+
  geom_line()+
  geom_point( aes(x = x[which.min(MSE)], y = y[which.min(MSE)]), color = 'red')
```



3. Compare your results to the true model used to generate the data.

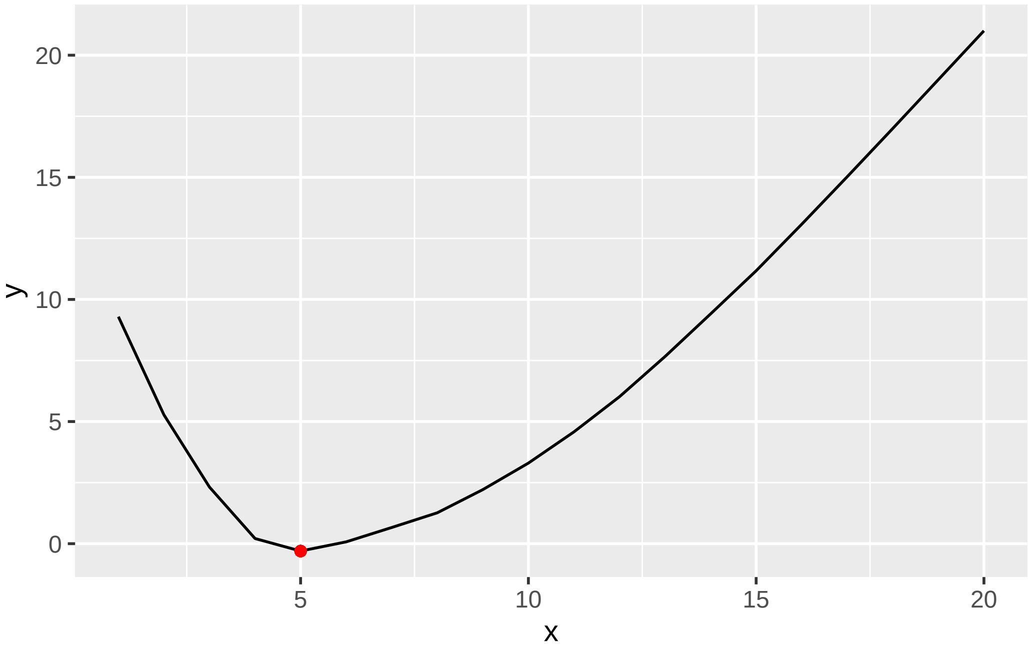
```
## Perform best subset selection using leaps package
reg.fit.test = regsubsets(y~.,test, nvmax = 20)

reg.summary.test = summary(reg.fit.test)

names(reg.summary.test)
```

```
[1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

```
plotCP.test <-data.frame(x = 1:length(reg.summary.test$cp), y = reg.summary.test$cp)

which.min(reg.summary.test$cp)
```

```
[1] 5
```

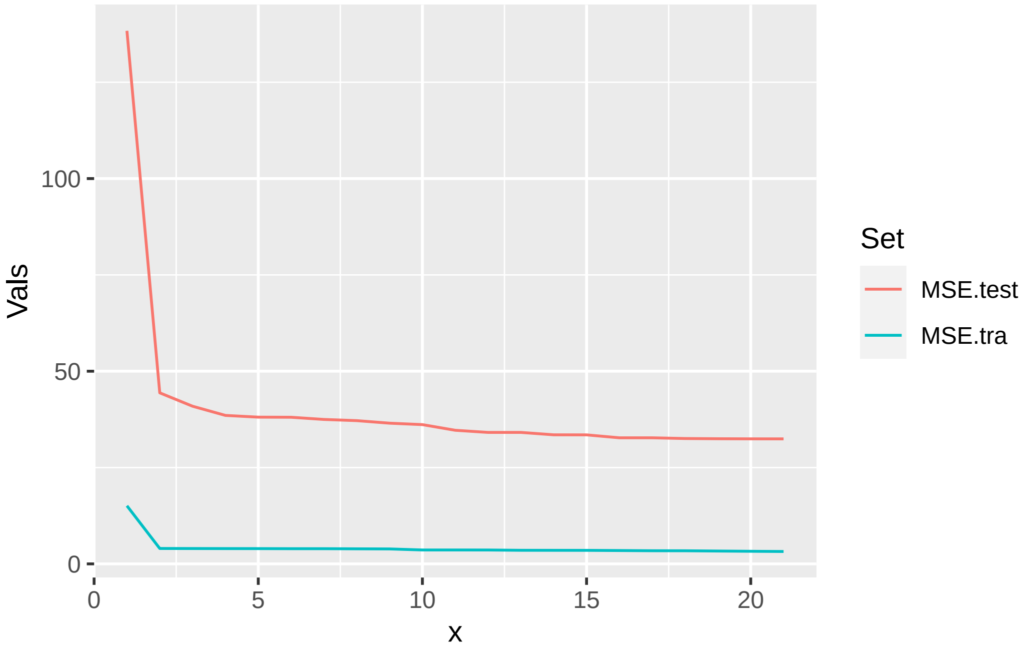
```
ggplot(plotCP.test, aes(x = x, y = y) )+
  geom_line()+
  geom_point( aes(x = x[which.min(reg.summary.test$cp)], y = y[which.min(reg.summary.test$
```



```
MSE.test <- reg.summary.test$obj$rss/100

plotMSE <-data.frame(x = 1:length(MSE.test), MSE.tra = MSE, MSE.test = MSE.test)

plotMSEs <- plotMSE |> pivot_longer(cols = c(MSE.tra,MSE.test), names_to = "Set", values_t

plotMSEs
```

```
# A tibble: 42 x 3
       x Set         Vals
   <int> <chr>      <dbl>
 1     1 MSE.tra    15.1 
 2     1 MSE.test  138.  
 3     2 MSE.tra     4.00
 4     2 MSE.test   44.4 
 5     3 MSE.tra     3.98
 6     3 MSE.test   40.9 
 7     4 MSE.tra     3.97
 8     4 MSE.test   38.5 
 9     5 MSE.tra     3.96
10     5 MSE.test   38.1 
# i 32 more rows
```

```
ggplot(plotMSEs, aes(x = x, y = Vals) )+
  geom_line(aes(color = Set))
```



Question 2:

Suppose that we have $n$ distinguishable samples and that we perform a bootstrap sampling once. Mathematically show that the expected value of the fraction of unique samples is roughly $2/3$. Simulate this process in ‘R‘ and verify your answer.

```
library(MASS)
library(tidyverse)

## Simulation for R = a X + (1 - a) Y
mu.x <- 1
mu.y <- 1.4
s.x2 <- 1
s.y2 <- 1.25
s.xy <- 0.5

## Covariance matrix
S <- matrix(c(s.x2, s.xy, s.xy, s.y2), ncol = 2, byrow = T)
```

```
## True alpha
a <- (s.y2 - s.xy) / (s.x2 + s.y2 - 2 * s.xy)

## a.hat is an estimator of a.
n <- 100
m <- 1
d <- lapply(1:m, function(i){
  xy <- mvrnorm(n, c(mu.x, mu.y), S) ## one sample of size n
  return(data.frame(xy, sim = i))
})

d <- bind_rows(d)
colnames(d) <- c('X', 'Y', 'sim')




################################### Bootstraping#################
## Use a Bootstrap strategy for the same problem
## generate 1 data set of size 100
# if m is 1, I do not use a for in this point

head(d) ## one sample of size n from the original sample to calcualte a.hat
```

```
          X          Y sim
1 2.5944395  1.7395690   1
2 2.2921053  1.0834579   1
3 0.8738808  0.2384762   1
4 0.9323266 -0.5111870   1
5 1.5016493  3.2237604   1
6 1.1378291 -0.3992851   1
```

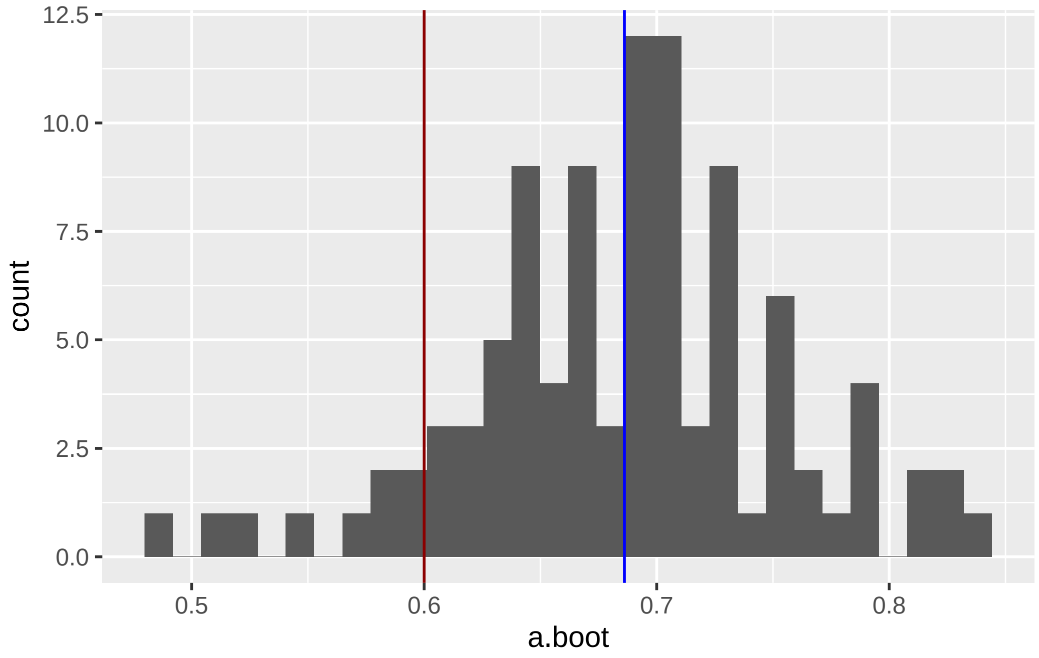
```
## in the loop sample the rows with replacement


d2 <- lapply(1:100, function(i){
   xy<- d[sample(100,replace = T),-3] # boots


  return(data.frame(xy, sim = i))
})
```

```r
d2 <- bind_rows(d2)
nrow(d2)
```

```
[1] 10000
```

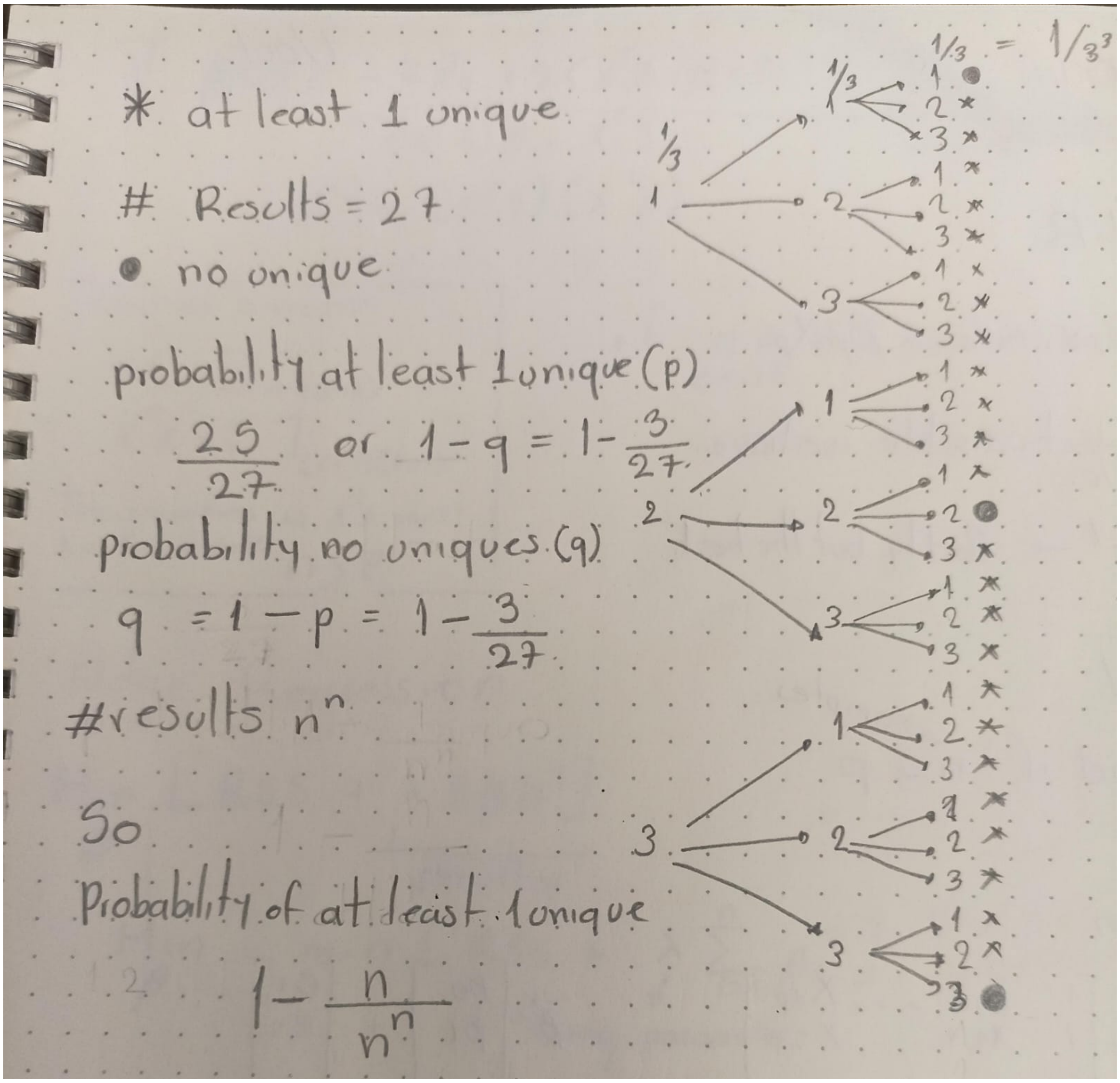
```r
  my.stats.Real <- d2 %>% group_by(sim) %>%
  summarise(s.x2_hat = (1/(n-1)) * sum((X - mean(X))^2),
            s.y2_hat = (1/(n-1)) * sum((Y - mean(Y))^2),
            s.xy_hat = (1/(n-1)) * sum((X - mean(X))*(Y - mean(Y))))

a.boot <- my.stats.Real %>%
  transmute(a.boot = (s.y2_hat - s.xy_hat) / (s.x2_hat + s.y2_hat - 2 * s.xy_hat))


# joint a.hat (actual) + a.boot (simul)


## compare the distributions: real alpha = darkred, a by bootstrap : blue

ggplot(a.boot, aes(x = a.boot)) +
   geom_histogram()+
  geom_vline( xintercept = a, color = "darkred")+
  geom_vline( xintercept = mean(a.boot$a.boot), color = "blue")
```

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



Question 3:

probability of unique observations in sampling



$*$ at least 1 unique

Results = 27

● no unique

probability at least 1 unique (p)

$\frac{25}{27}$ or $1 - q = 1 - \frac{3}{27}$

probability no uniques (q)

$q = 1 - p = 1 - \frac{3}{27}$

#results n^n

So

Probability of at least 1 unique

$1 - \frac{n}{n^n}$

1/3 = 1/3^3
1/3
1/3
1
2
3
1
2
3

Figure 1: unique samples