Judgment sampling

Consider again the `blocks` data, as our study population $\mathcal{P}_{Study}$ consisting of $N = 100$ blocks labelled $u = 1, 2, 3, \ldots, 100$.

Recall that the blocks are of of uniform thickness and density (all blocks were cut from the same opaque plastic sheet of about 5mm thickness), but have different convex shapes such as shown below:

#knitr::include_graphics(path_concat(imageDirectory, "blocks.png"))

---

24 marks

A number of graduate data science students were actually presented with the physical blocks whose values are given as the `blocks` data from loon.data.

The students could examine all 100 blocks (without touching them) but were not given access to the recorded variate values (i.e. the actual measurements in the `blocks` data). Each block had its unique identification number marked on it.

Each student was asked to use their judgment and, by carefully considering the various shapes and sizes of the blocks, to choose a sample of 10 blocks which, **in their judgment**, would have an average weight (over those 10 blocks) that came close to matching the average weight of **all** $N = 100$ blocks in the population.

The competition was to choose a sample whose *sample error* $a(\mathcal{S}) - a(\mathcal{P}_{Study})$ was as small in absolute magnitude as possible.

Having been presented with all 100 blocks and asked to **judge** which 10 blocks have an average weight nearest the average weight of all 100 blocks, each student would have come up with their own sampling plan based on their judgment. This type of sampling is called **judgment sampling**.

The id numbers of the students and the blocks they selected are recorded in another data set from loon.data called `judgment`.

```
library(loon.data)
data("judgment", package = "loon.data")
head(judgment, n = 3)
```

```
##   studentID first second third fourth fifth sixth seventh eighth ninth tenth
## 1      5086    12     18    17     11    15    20      14     13    16    18
## 2      3848    34     35    70     56    32    14       5     88    81    73
## 3      6656    14     34    41     29    32    55      74     40    16    70
```

The variates of `judgment` identify the student and the id numbers of the blocks they selected, in the order they selected and recorded them.

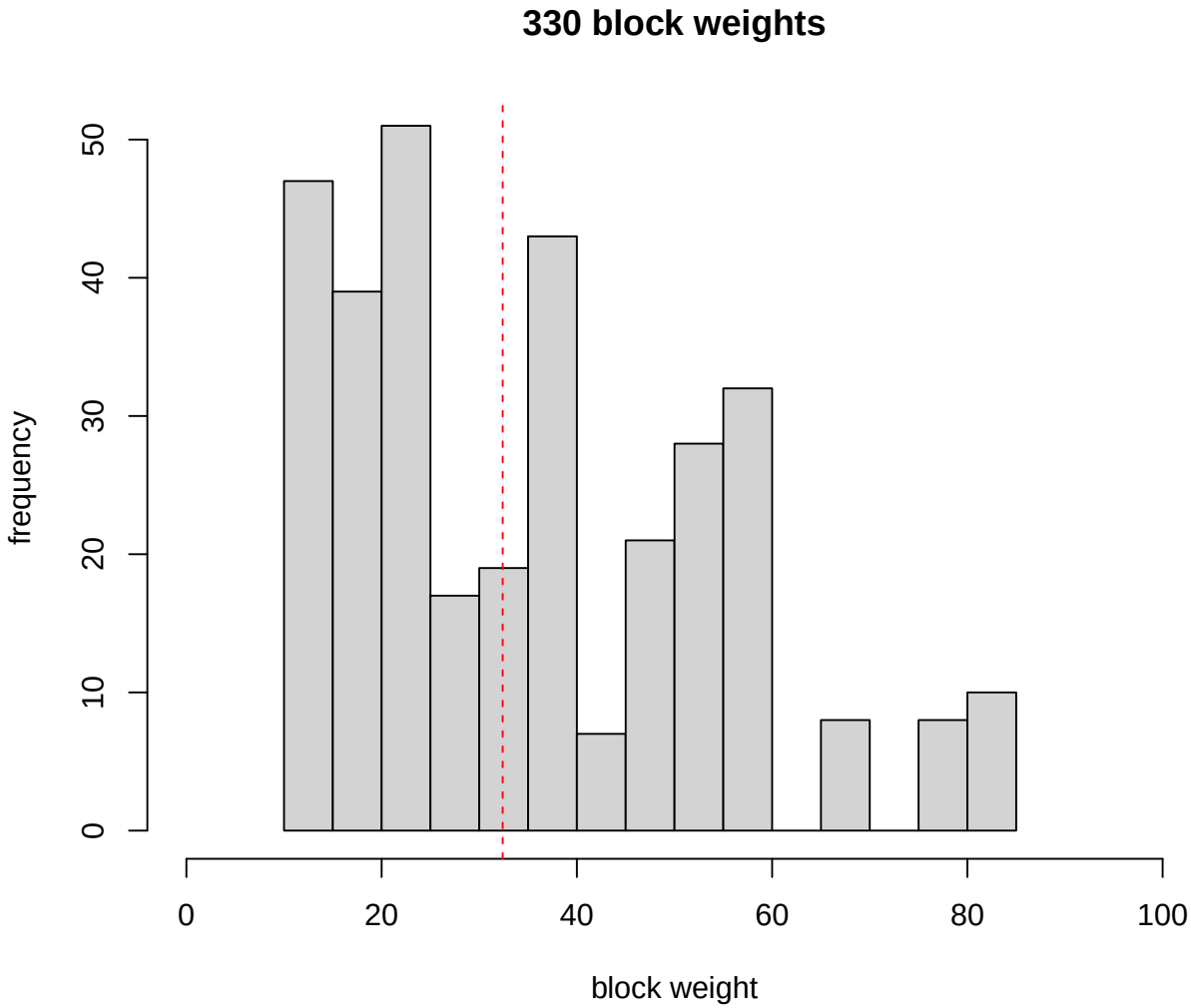
a. *(4 marks)* Draw a histogram of all of the block weights selected by the students. If any block was selected by more than one student, include its weight as often as it was selected.

   That is, there will be a total of 330 weights used to construct the histogram.

   - Make sure the histogram is suitably labelled
   - Add a vertical dashed red line at the at the average of all 100 weights in the entire population of 100 blocks (i.e. not just those selected by students).

Show your code.

```
judgment <- within(judgment,
                   {
                     first <- blocks[judgment$first,]['weight']
                     second <- blocks[judgment$second,]['weight']
                     third <- blocks[judgment$third,]['weight']
                     fourth <- blocks[judgment$fourth,]['weight']
                     fifth <- blocks[judgment$fifth,]['weight']
                     sixth <- blocks[judgment$sixth,]['weight']
                     seventh <- blocks[judgment$seventh,]['weight']
                     eighth <- blocks[judgment$eighth,]['weight']
                     ninth <- blocks[judgment$ninth,]['weight']
                     tenth <- blocks[judgment$tenth,]['weight']
                   })
all_weights = c(judgment$first,judgment$second,judgment$third,judgment$fourth,judgment$fifth,judgme
xlim = c(0,100)
hist(all_weights, xlim = xlim,breaks = 20, main = "330 block weights", xlab="block weight", ylab="f
abline(v = mean(blocks['weight']), col = "red", lty = 2)
```



b. *(5 marks)* For each student, calculate the sample average weight of the blocks they selected. Create a data frame called `judgmentErrors` of the student ids and their sample errors. Print out the ids and

sample errors for both the top five and the bottom five students in increasing order of their *absolute* sample error.

Show your code.

```
judgment <- within(judgment,
                   {
                     averageWeight <- (first+second+third+fourth+fifth+sixth+seventh+eighth+ninth+te
                   }
)
judgment <- within(judgment,
                   {
                     sampleErrors = averageWeight - mean(blocks[,'weight'])
                   })
judgmentErrors <- data.frame(studentID = judgment$studentID, sampleErrors = judgment$sampleErrors)
judgmentErrors <- judgmentErrors[order(abs(judgmentErrors$sampleErrors)),]
topFive <- tail(judgmentErrors, n = 5)
bottomFive <- head(judgmentErrors, n = 5)
topFive
```

```
##    studentID sampleErrors
## 1       5086         11.6
## 3       6656         11.6
## 22      7231         12.1
## 5       4114         12.6
## 27      7582         12.6
```

```
bottomFive
```

```
##    studentID sampleErrors
## 14      7656          2.6
## 17      7626          2.6
## 31      8395          2.6
## 12       842          3.1
## 26      7954         -3.4
```

c. *(3 marks)* Estimate the sampling bias and the sampling standard deviation for judgment sampling on this data. Show your code.

```
#calculate sample errors without abs()
mean(judgmentErrors$sampleErrors)
```

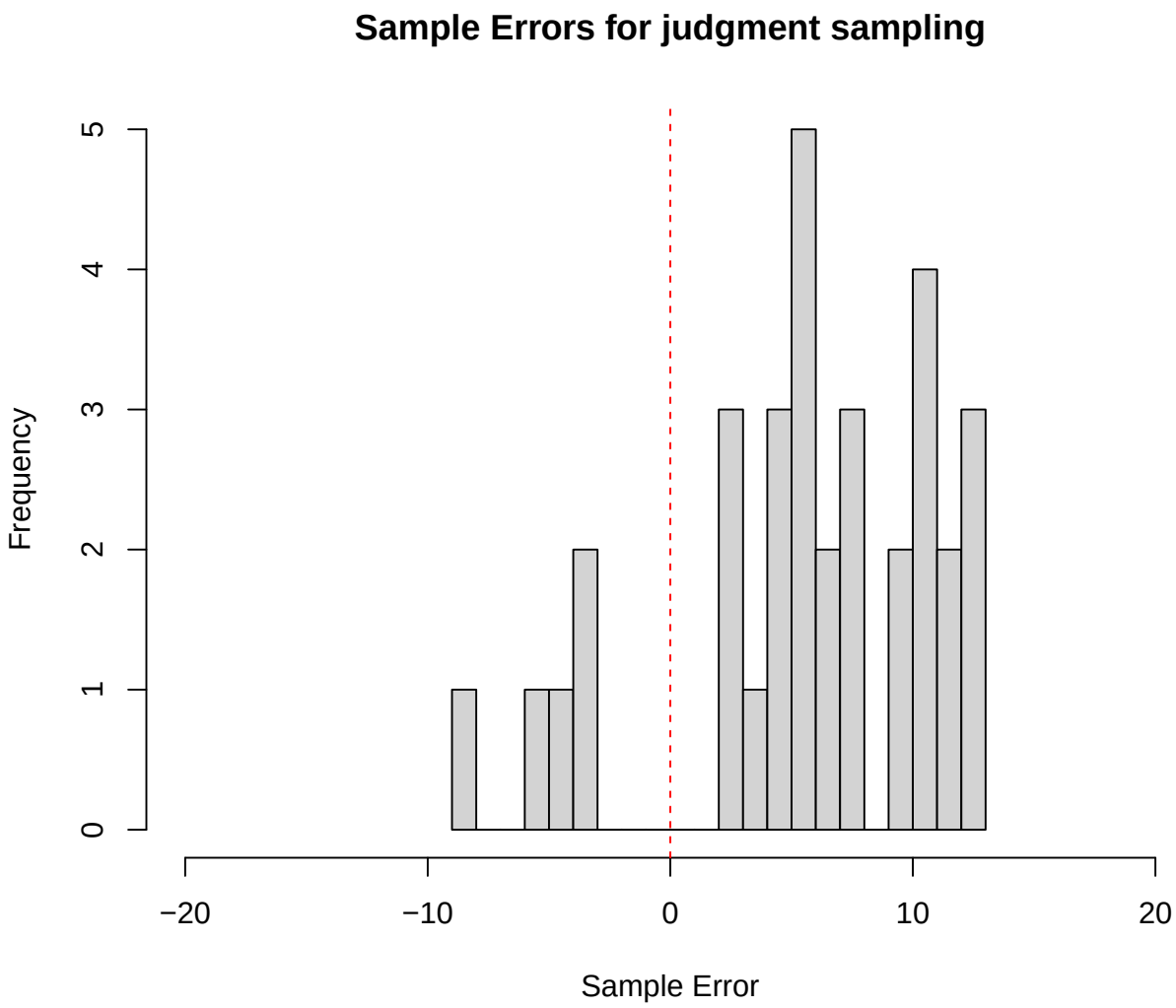
```
## [1] 5.418182
```

```
# standard deviation
sd(judgmentErrors$sampleErrors)
```

```
## [1] 5.508258
```

d. *(3 marks)* Provide a (suitably labelled) histogram of the sample errors. Add a vertical red dashed line at 0.

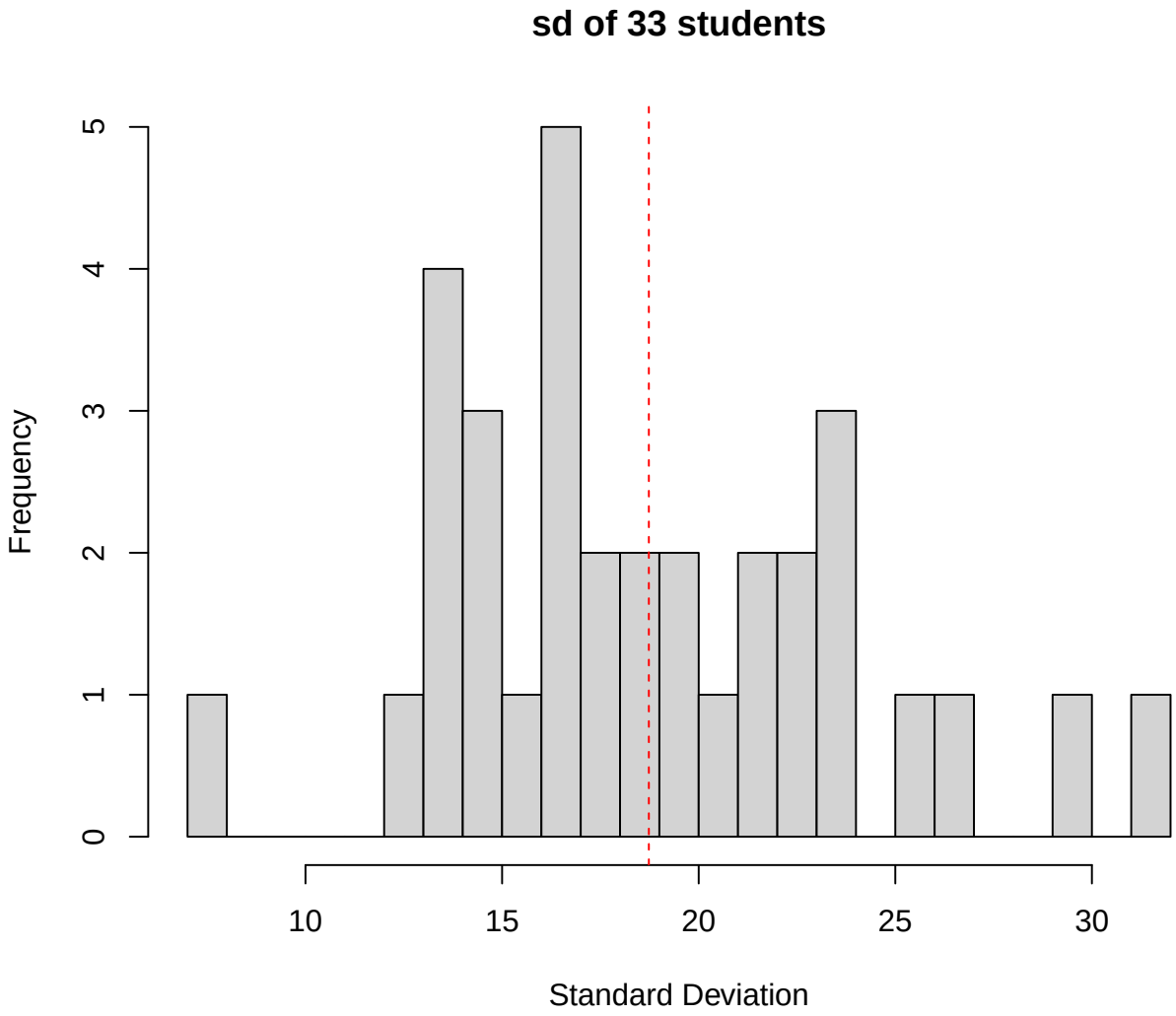
```
xlim = c(-20,20)
hist(judgmentErrors$sampleErrors, xlim = xlim,breaks = 20, main = "Sample Errors for judgment sampl
abline(v = 0, col = "red", lty = 2)
```



e. *(3 marks)* Calculate the sample standard deviation of the weights selected for each of the judgment samples. Draw a histogram of these standard deviations (suitably labelled). Draw a vertical dashed red line at the average of these standard deviations.

Show your code.

```
sd_judgment<-judgment[, c(2,3,4,5,6,7,8,9,10,11)]
standard_deviation <- apply(sd_judgment, 1, sd)
hist(standard_deviation, breaks = 30, main = "sd of 33 students", xlab = "Standard Deviation")
abline(v = mean(standard_deviation), col = "red", lty = 2)
```



f. *(6 marks)* Identify which student had the smallest sample standard deviation **and** which student had the largest sample standard deviation. Report their standard deviations.

Draw histograms (suitably labelled **and** having the same `xlim = extendrange(blocks$weight)`) of the weights of the blocks selected by each of these students. Add a vertical dashed red line to each histogram at the average of all 100 block weights in the population. What do you conclude about the sampling plan of each of these students?

Show your code.

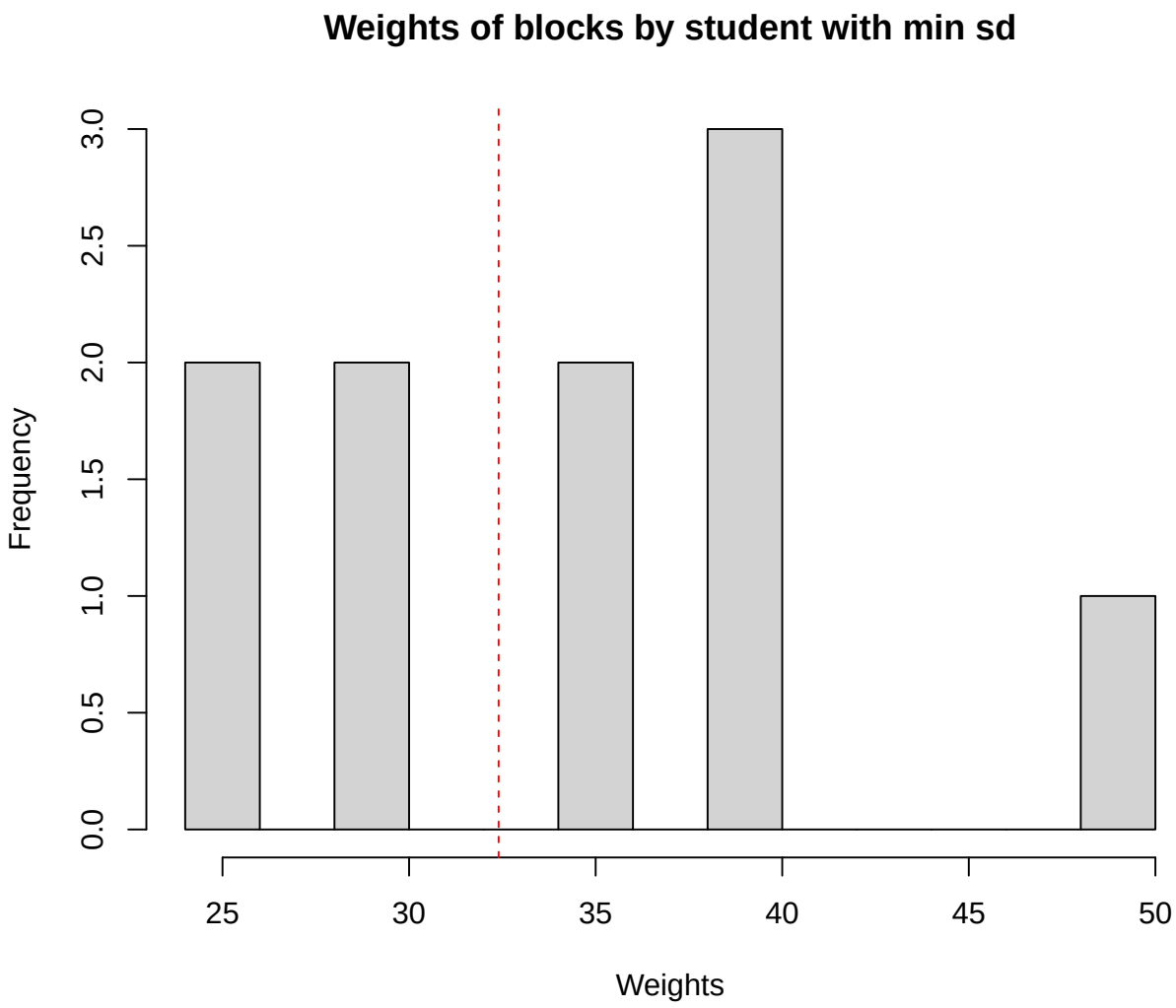
```
sd_judgment$sd = standard_deviation
sd_judgment$studentID = judgment$studentID
min_index <- which.min(sd_judgment$sd)
max_index <- which.max(sd_judgment$sd)
sd_judgment[min_index,c("sd","studentID")]
```

```
##          sd studentID
## 14 7.81736      7656
```

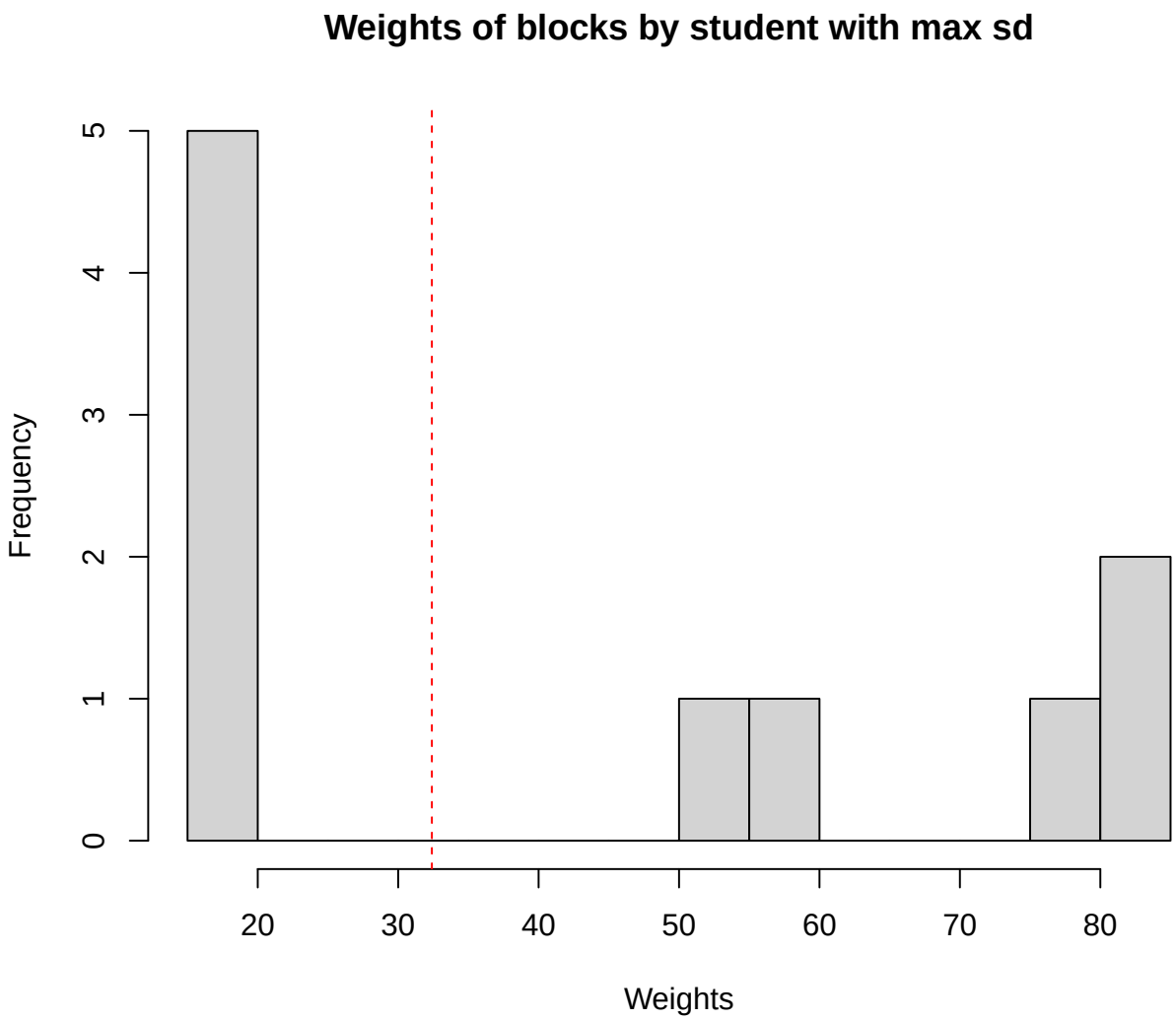
```
sd_judgment[max_index,c("sd","studentID")]
```

```
##          sd studentID
## 27 31.09126      7582
```

```r
xlim = extendrange(blocks$weight)
sd_judgment <- within(sd_judgment,
                      {
                      rm(sd, studentID)
                      }
)
hist(unlist(sd_judgment[min_index,]), breaks = 10, main = "Weights of blocks by student with min sd
abline(v = mean(blocks[,'weight']), col = "red", lty = 2)
```




```r
hist(unlist(sd_judgment[max_index,]), breaks = 10, main = "Weights of blocks by student with max sd
abline(v = mean(blocks[,'weight']), col = "red", lty = 2)
```



From the histogram above, we can see that the student with smallest sd tends to pick the blocks that are not too extreme. On the other hand, the student with highest sd tends to pick blocks that are either too small or too big.