chapter11. Apply family

Jongrak Jeong

2023-01-20

1. apply()

a <- rep(0, n)

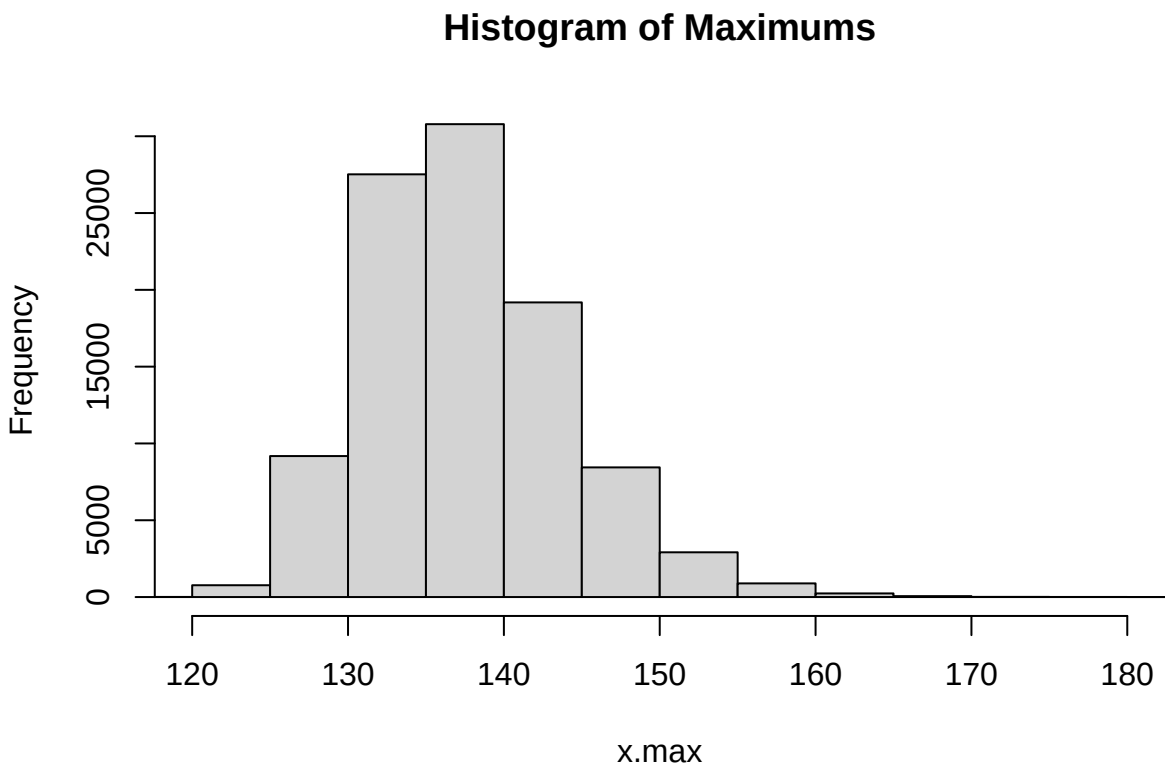
for (i in 1:n) a[i] <- f(A[i,]) == a <- apply(A, 1, f) # 1 means 'by columns'

```
n <- 100000
m <- 100
A <- matrix(rnorm(n * m, 100, 15), n, m)
```

```
time <- proc.time()
a.1 <- apply(A, 1, max)
proc.time() - time
```

```
##    user  system elapsed
##   0.189   0.034   0.227
```

```
hist(a.1, xlim = c(120, 180),
     xlab = "x.max", main = "Histogram of Maximums")
```



```r
apply.1 <- function(X, f) {
  out <- rep(0, nrow(X))
  for (i in 1:nrow(X)) out[i] <- f(X[i,])
}
```

```r
time <- proc.time()
a.1 <- apply.1(A, max)
proc.time() - time
```

```
##    user  system elapsed
##   0.145   0.017   0.168
```

2. lapply() and sapply()

lapply(): list > list

```r
n <- 10000
m <- 100
s <- rpois(n, m)

A <- vector(mode = "list", length = n)
for (i in 1:n) A[[i]] <- rnorm(s[i], 100, 15)
```

```r
time <- proc.time()
a.1 <- lapply(A, max)
proc.time() - time
```

```
##    user  system elapsed
##   0.011   0.001   0.012
```

```r
head(a.1, 5)
```

```
## [[1]]
## [1] 140.4304
##
## [[2]]
## [1] 129.3383
##
## [[3]]
## [1] 132.8531
##
## [[4]]
## [1] 135.9561
##
## [[5]]
## [1] 138.309
```

```r
head(unlist(a.1), 5)
```

```
## [1] 140.4304 129.3383 132.8531 135.9561 138.3090
```

sapply(): > vector

```r
time <- proc.time()
a.1 <- sapply(A, max)
```

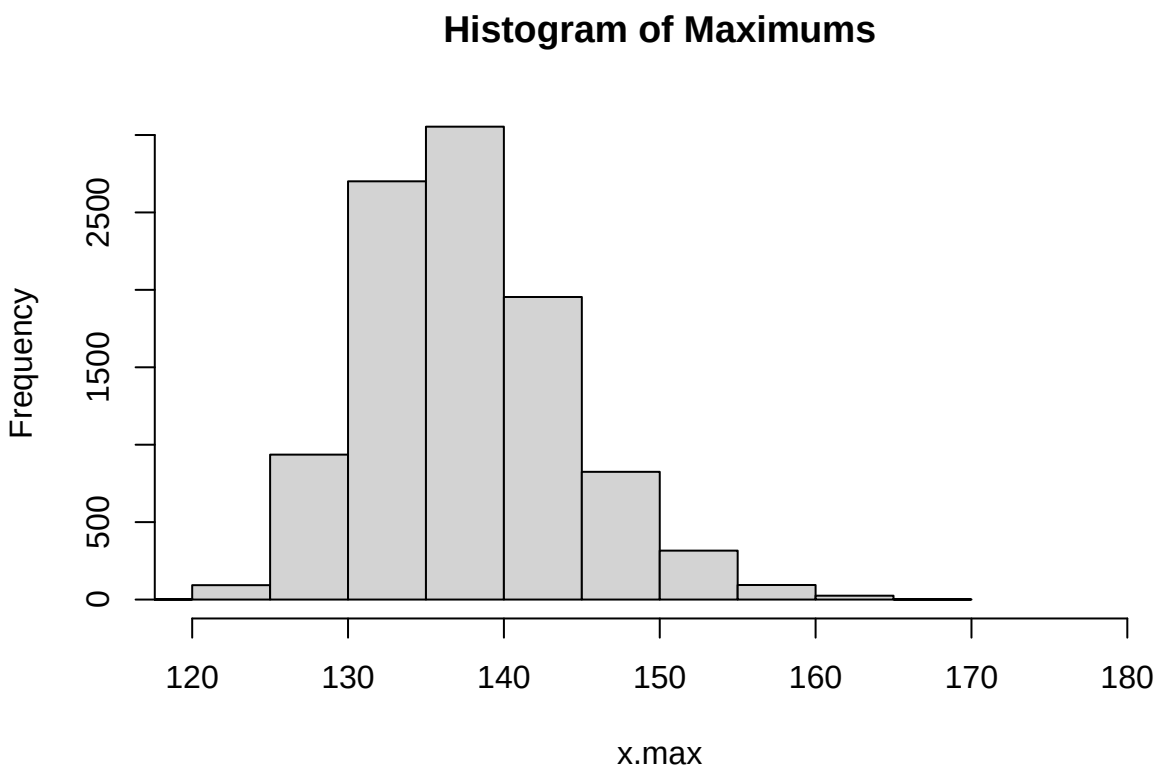
```
proc.time() - time
```

```
##    user  system elapsed
##   0.007   0.000   0.007
```

```
head(a.1, 5)
```

```
## [1] 140.4304 129.3383 132.8531 135.9561 138.3090
```

```
hist(a.1, xlim = c(120, 180), xlab = "x.max", main = "Histogram of Maximums")
```



ex1. trimmed mean

```
a.2 <- lapply(A, mean, trim = 0.1)
head(a.2, 5)
```

```
## [[1]]
## [1] 99.95123
##
## [[2]]
## [1] 100.6119
##
## [[3]]
## [1] 100.1837
##
## [[4]]
## [1] 99.4752
##
## [[5]]
## [1] 100.1926
```

```r
a.2 <- sapply(A, mean, trim = 0.1)
head(a.2, 5)
```

```
## [1]  99.95123 100.61186 100.18366  99.47520 100.19261
```

ex2. range, min, and max

```r
a.3 <- lapply(A, range)
head(a.3, 5)
```

```
## [[1]]
## [1]  65.54375 140.43039
##
## [[2]]
## [1]  66.14984 129.33834
##
## [[3]]
## [1]  74.68637 132.85311
##
## [[4]]
## [1]  67.05423 135.95606
##
## [[5]]
## [1]  69.55738 138.30899
```

```r
a.4 <- sapply(A, range)
str(a.4)
```

```
##  num [1:2, 1:10000] 65.5 140.4 66.1 129.3 74.7 ...
```

```r
head(t(a.4), 5)
```

```
##          [,1]     [,2]
## [1,] 65.54375 140.4304
## [2,] 66.14984 129.3383
## [3,] 74.68637 132.8531
## [4,] 67.05423 135.9561
## [5,] 69.55738 138.3090
```

3. mapply()

same function to several targets

mapply(f, a, b, c) SIMPLIFY = F -> list SIMPLIFY = T -> vector

sampling

```r
# k = 1, 2, 3, ..., 100. From N(k, k), 40 observations.
# rnorm(40, k, k)
X <- mapply(rnorm, rep(40, 100), 1:100, 1:100, SIMPLIFY = T)

str(X)
```

```
##  num [1:40, 1:100] 2.323 2.35 3.377 0.596 1.123 ...
```

practice 1

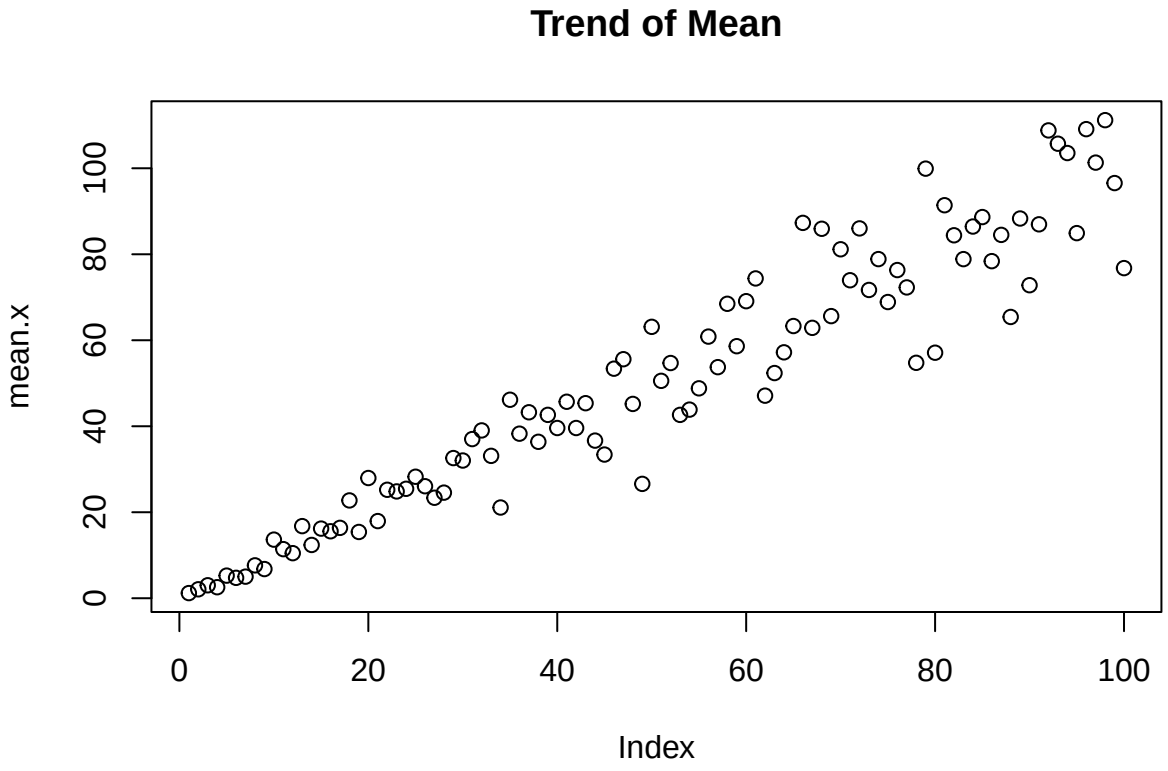
```
mean.x <- apply(X, 2, mean)
sd.x <- apply(X, 2, sd)
head(cbind(mean.x, sd.x), 5)
```

```
##        mean.x     sd.x
## [1,] 1.196353 1.047715
## [2,] 2.077087 2.391907
## [3,] 3.014577 2.386258
## [4,] 2.560027 4.860618
## [5,] 5.258242 5.731982
```

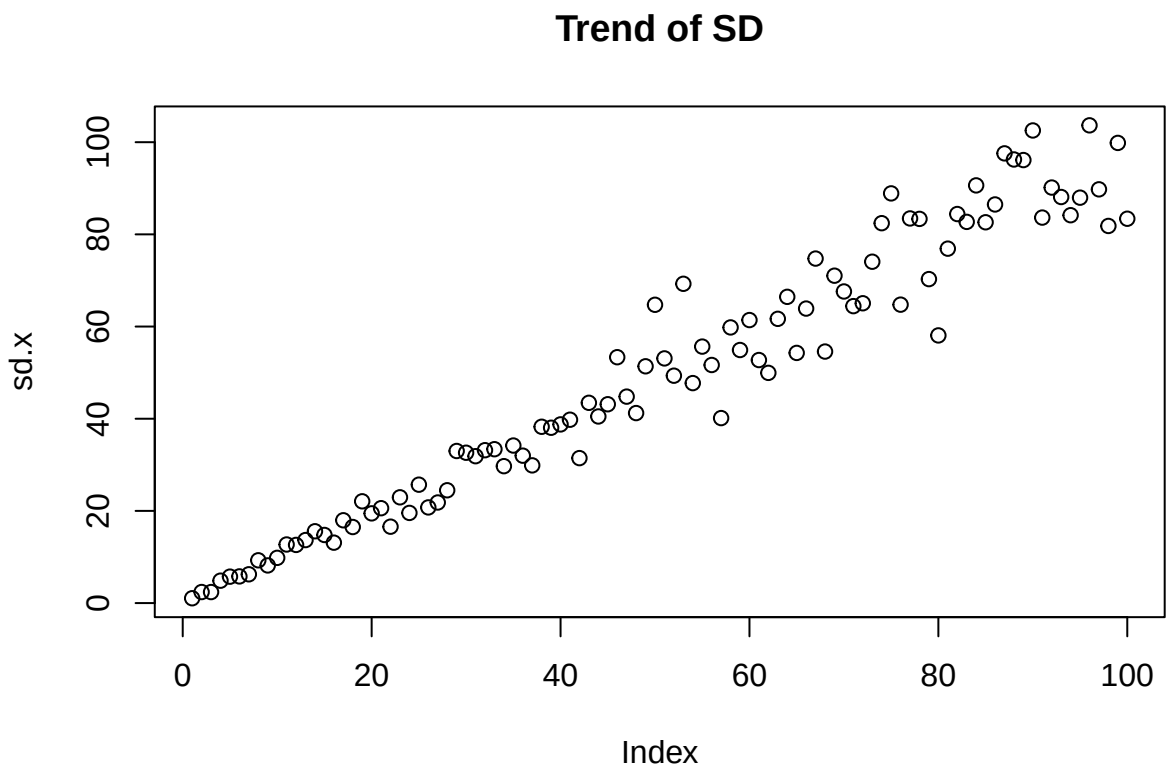
```
tail(cbind(mean.x, sd.x), 5)
```

```
##           mean.x      sd.x
##  [96,] 109.10123 103.65408
##  [97,] 101.31698  89.76954
##  [98,] 111.18971  81.82250
##  [99,]  96.55987  99.84069
## [100,]  76.78043  83.39202
```

```
plot(mean.x, main = "Trend of Mean")
```




```
plot(sd.x, main = "Trend of SD")
```



4. tapply(): summarize

```
n <- 10000
m <- 100
N <- n * m

x <- rnorm(N, 100, 15)
grp <- sample(1:n, N, replace = T)

time <- proc.time()
x.1 <- tapply(x, grp, max)
proc.time() - time
```

```
##    user  system elapsed
##   0.050   0.003   0.052
```

practice 2

```
class(summary(x, grp))
```

```
## [1] "summaryDefault" "table"
```

practice 3

```
time <- proc.time()
head(aggregate(x, by = list(grp), max), 5)
```

```
##   Group.1        x
```

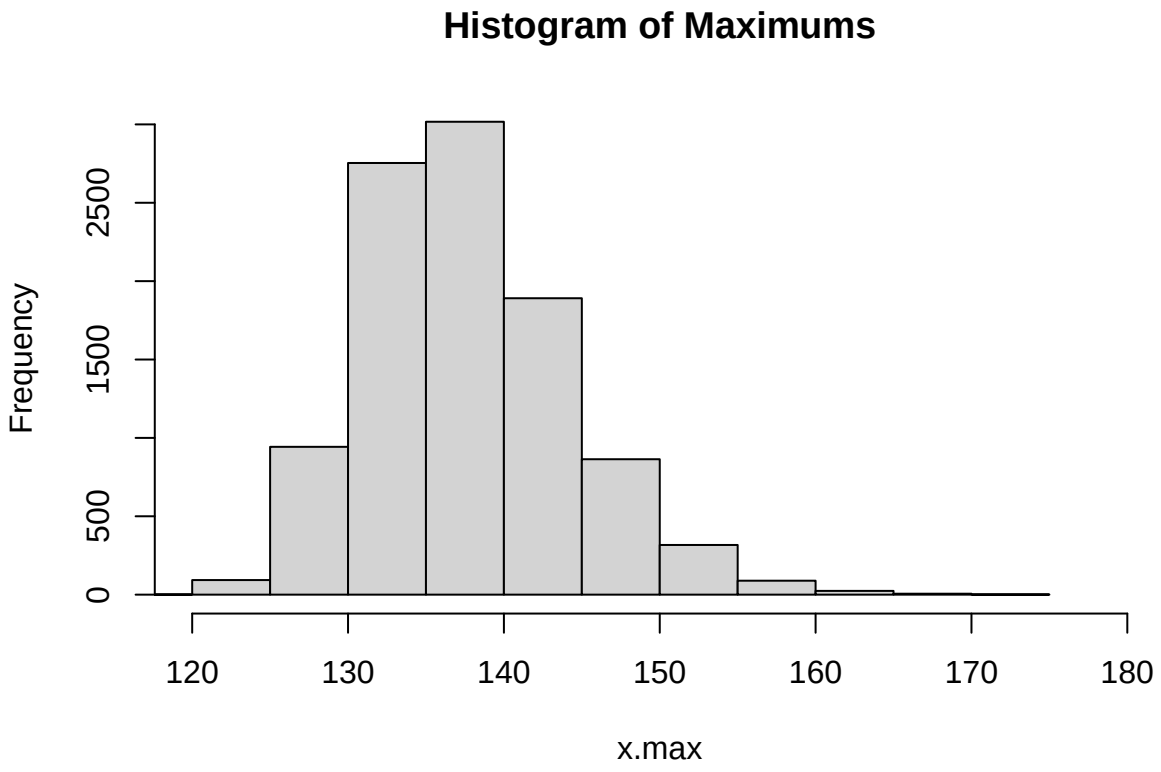
```
## 1       1 137.6851
## 2       2 137.0693
## 3       3 129.8610
## 4       4 125.6014
## 5       5 137.0163
```

```
proc.time() - time
```

```
##    user  system elapsed
##   0.208   0.016   0.224
```

practice 4

```
hist(x.1, xlim = c(120, 180), xlab = "x.max", main = "Histogram of Maximums")
```



5. Loop

```
n <- 100000
m <- 100
A <- matrix(rnorm(n * m, 100, 15), n, m)

time <- proc.time()
a.1 <- apply(A, 1, max)
proc.time() - time
```

```
##    user  system elapsed
##   0.179   0.023   0.202
```

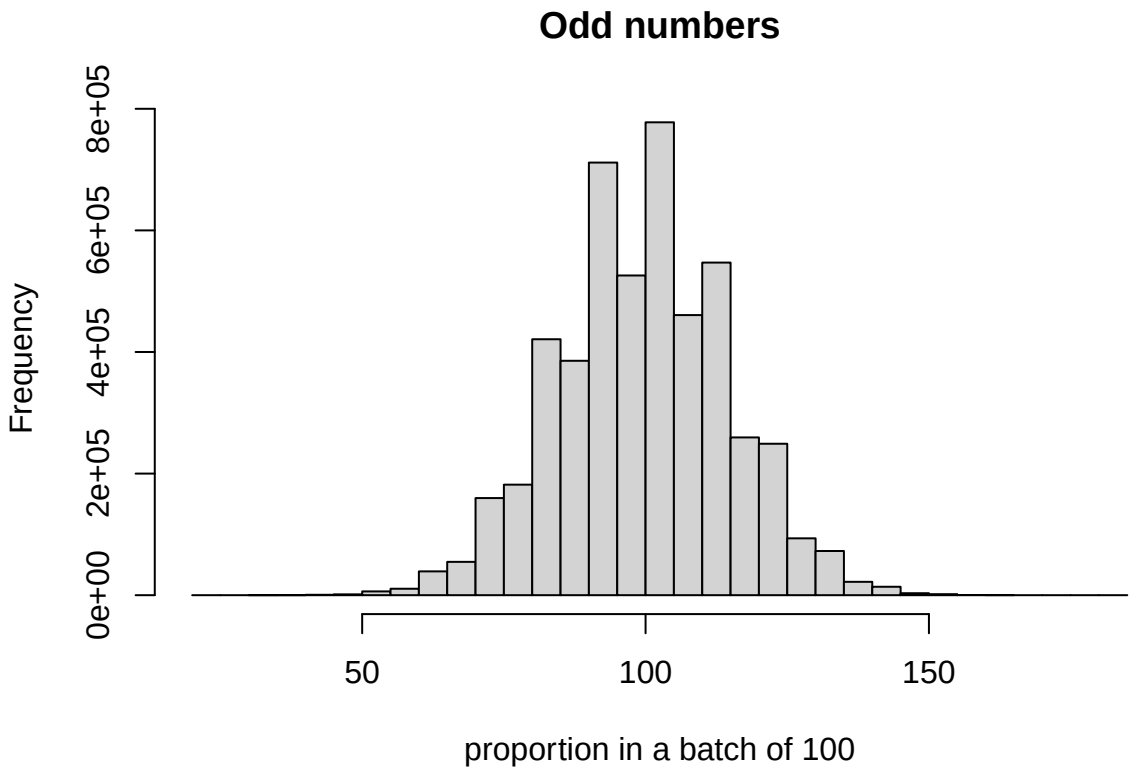
Loop

```
time <- proc.time()
a.1 <- matrix(0, n)
for (i in 1:n) {
  a.1[i] <- A[i, 1]
  for (j in 2:m) if(A[i, j] > a.1[i]) a.1[i] <- A[i, j]
}
proc.time() - time
```

```
##    user  system elapsed
##   0.338   0.005   0.343
```

practice 5.

```
A.1 <- round(A)
A.1 <- A.1[(A.1 %% 2) == 1]
hist(A.1, xlab = "proportion in a batch of 100",
     main = "Odd numbers", freq = T)
```



practice 6.

```
n <- 10000
m <- 100
s <- rpois(n, m)
A <- vector(mode = "list", length = n)
for (i in 1:n) A[[i]] <- rnorm(s[i], 100, 15)


A.1.a <- lapply(A, sort)
str(A.1.a)
```

```
## List of 10000
##  $ : num [1:95] 52.7 55.9 62.6 65.8 68.8 ...
##  $ : num [1:104] 57.4 63.7 66.7 73.8 74 ...
##  $ : num [1:88] 46.9 74.1 76.1 77.4 77.4 ...
##  $ : num [1:101] 54.2 60.2 68.1 71.6 72.3 ...
##  $ : num [1:102] 47.1 53.3 70.8 72.8 75 ...
##  $ : num [1:91] 64.7 67.4 67.8 70.2 76.7 ...
##  $ : num [1:90] 62.7 70.6 75.9 76.4 77 ...
##  $ : num [1:97] 53.3 61.2 64.4 72.1 73.3 ...
##  $ : num [1:94] 60.6 61.6 63.2 65.3 68.5 ...
##  $ : num [1:91] 57.5 63.1 64.2 70.1 73.4 ...
##  $ : num [1:96] 57.4 60.9 62.8 68.2 69.1 ...
##  $ : num [1:115] 58.3 69.4 69.6 70.1 72 ...
##  $ : num [1:111] 50.8 63.2 63.7 66 68.2 ...
##  $ : num [1:93] 60.9 67.2 67.8 76.1 81.1 ...
##  $ : num [1:88] 46.6 68.8 70 72.1 72.1 ...
##  $ : num [1:98] 53 70.8 72.7 72.9 78.1 ...
##  $ : num [1:93] 58 67.1 77.5 81.7 83.4 ...
##  $ : num [1:99] 65.3 70.2 71.4 73.7 73.9 ...
##  $ : num [1:110] 64.8 68.8 69 72.8 73.1 ...
##  $ : num [1:104] 64.8 65.3 70.6 71.9 73.1 ...
##  $ : num [1:106] 48.5 67.8 71.4 72.3 73.5 ...
##  $ : num [1:93] 68.4 68.7 71.3 76.7 77.3 ...
##  $ : num [1:103] 62.1 63.7 67.8 71 72.4 ...
##  $ : num [1:92] 60.6 63 66.2 71 76.1 ...
##  $ : num [1:92] 66.9 71.9 74.8 77.1 79.4 ...
##  $ : num [1:89] 60.8 67.8 76.9 77.2 77.4 ...
##  $ : num [1:115] 59.2 66.1 66.3 67.5 69 ...
##  $ : num [1:105] 65.6 69.5 70.2 72.8 73.3 ...
##  $ : num [1:92] 62.4 64.5 64.6 69.8 73 ...
##  $ : num [1:108] 67.4 69.1 70.2 71.7 71.9 ...
##  $ : num [1:93] 67.4 69.5 69.7 73.1 78.2 ...
##  $ : num [1:95] 63.9 69.8 74.9 75.1 77.9 ...
##  $ : num [1:113] 55.6 61 78.4 78.5 78.8 ...
##  $ : num [1:100] 65.5 68.8 69.9 70.6 72.6 ...
##  $ : num [1:100] 65.7 66.2 66.9 67.1 69.9 ...
##  $ : num [1:84] 74.9 76 76.9 77.4 78 ...
##  $ : num [1:112] 62.4 63.2 72.1 72.1 72.7 ...
##  $ : num [1:96] 71.6 72.4 72.8 73.5 74.2 ...
##  $ : num [1:74] 75.2 75.5 76.5 77.7 79.8 ...
##  $ : num [1:89] 65.6 66.4 69.9 76.2 77.1 ...
##  $ : num [1:83] 49.1 63.2 70.9 77.6 78 ...
##  $ : num [1:107] 61.9 68.4 68.6 69.2 70.2 ...
##  $ : num [1:104] 62.6 68.5 73.5 73.6 73.8 ...
##  $ : num [1:102] 72.7 73.4 73.7 75.1 76.5 ...
##  $ : num [1:93] 62.3 64.4 67.4 67.9 72.4 ...
##  $ : num [1:104] 58.7 66 67.1 68.9 72.7 ...
##  $ : num [1:84] 53.1 60.9 62.9 70.8 72.8 ...
##  $ : num [1:102] 66.8 69.1 70.9 71.4 71.7 ...
##  $ : num [1:107] 52.6 63.9 66.2 71.2 72.1 ...
##  $ : num [1:104] 64.4 65.4 66.7 74.3 75.8 ...
##  $ : num [1:104] 62 71.4 74.4 75.7 75.8 ...
##  $ : num [1:99] 65.8 66.3 70.4 74.2 75.6 ...
##  $ : num [1:94] 64.8 64.8 67.9 68.4 69.8 ...
```

```
##  $ : num [1:95] 55 62.6 65.2 65.9 70.5 ...
##  $ : num [1:106] 65.2 65.3 71.5 71.6 75.3 ...
##  $ : num [1:93] 62.3 67.3 68.5 70.1 72.1 ...
##  $ : num [1:111] 69.9 69.9 72 74 74.2 ...
##  $ : num [1:92] 56.7 56.9 67.1 69.7 71.4 ...
##  $ : num [1:94] 65.7 67.6 69.3 71.6 72.1 ...
##  $ : num [1:94] 58.6 66 69.8 78.3 79.8 ...
##  $ : num [1:109] 59.1 61.9 67.6 72.4 73.2 ...
##  $ : num [1:113] 64 67.4 67.9 69.2 73.7 ...
##  $ : num [1:110] 55.6 61.6 61.8 61.8 65.5 ...
##  $ : num [1:103] 66.2 68.4 72.1 72.4 75 ...
##  $ : num [1:107] 51.9 59.4 60.3 62.8 74.3 ...
##  $ : num [1:86] 62.1 65.6 66.3 69.1 71.4 ...
##  $ : num [1:107] 63.9 66.4 72.4 76.7 79.3 ...
##  $ : num [1:102] 59.3 67.3 68.5 68.5 70.3 ...
##  $ : num [1:91] 61 66.6 69.8 70.9 71.9 ...
##  $ : num [1:80] 73.2 77.7 78.7 79.2 80.1 ...
##  $ : num [1:88] 50 62 62.8 69.9 71.3 ...
##  $ : num [1:96] 60.2 68.6 71.6 71.6 72.3 ...
##  $ : num [1:116] 65 72.4 73.4 74.8 75.8 ...
##  $ : num [1:99] 51.9 64.6 64.8 65.8 67.2 ...
##  $ : num [1:94] 65.6 67 72.4 74.7 76.6 ...
##  $ : num [1:88] 64.8 67.8 68.2 69.1 71.5 ...
##  $ : num [1:96] 68.2 70.1 76.7 77.2 80.2 ...
##  $ : num [1:119] 62.1 63.3 64.2 64.9 65.4 ...
##  $ : num [1:91] 63.8 68.6 75.4 79.2 80 ...
##  $ : num [1:104] 52.7 55.6 63.9 70.6 74.9 ...
##  $ : num [1:86] 67.6 67.7 72.7 75.5 76.7 ...
##  $ : num [1:109] 65.2 70.4 75.1 76.8 78.1 ...
##  $ : num [1:96] 63.3 63.8 68.6 70.8 74.7 ...
##  $ : num [1:108] 61.9 63.3 70.1 71.9 72.5 ...
##  $ : num [1:104] 61.4 70.7 71.6 72.3 72.4 ...
##  $ : num [1:93] 59 66.9 67.6 71.1 72.4 ...
##  $ : num [1:103] 66.7 69.7 75.5 75.9 76.7 ...
##  $ : num [1:95] 61.2 64.7 68.5 71.2 72.7 ...
##  $ : num [1:86] 60.9 62.2 69.8 70.6 72 ...
##  $ : num [1:107] 56.3 56.9 72.5 77.6 78.4 ...
##  $ : num [1:105] 55.2 62.3 67.5 67.9 68.1 ...
##  $ : num [1:113] 56.2 67.4 70.6 71.3 71.6 ...
##  $ : num [1:90] 52.2 63.2 64.4 69.9 70 ...
##  $ : num [1:100] 62.5 69.3 70.4 70.4 72.5 ...
##  $ : num [1:101] 65.1 66.4 67.1 67.6 71.1 ...
##  $ : num [1:98] 64.5 73.4 77 78.6 78.7 ...
##  $ : num [1:95] 70 70.2 70.8 71.5 72.2 ...
##  $ : num [1:100] 69.6 69.7 76 76 76.2 ...
##  $ : num [1:91] 66.3 72.3 73 75.1 77.2 ...
##   [list output truncated]
```

```
A.2.s <- unlist(lapply(A.1.a, "[", 2))
str(A.2.s)
```

```
##  num [1:10000] 55.9 63.7 74.1 60.2 53.3 ...
```

```
A.1.d <- lapply(A, sort, decreasing = T)
str(A.1.d)
```

```
## List of 10000
##  $ : num [1:95] 150 136 135 128 127 ...
##  $ : num [1:104] 136 126 126 125 124 ...
##  $ : num [1:88] 149 129 129 127 127 ...
##  $ : num [1:101] 139 132 130 127 125 ...
##  $ : num [1:102] 134 128 127 127 126 ...
##  $ : num [1:91] 126 125 120 120 118 ...
##  $ : num [1:90] 138 132 124 124 122 ...
##  $ : num [1:97] 140 133 131 129 127 ...
##  $ : num [1:94] 130 128 128 127 126 ...
##  $ : num [1:91] 143 130 127 126 122 ...
##  $ : num [1:96] 130 130 122 121 121 ...
##  $ : num [1:115] 145 134 130 129 126 ...
##  $ : num [1:111] 136 133 131 130 129 ...
##  $ : num [1:93] 135 134 131 129 128 ...
##  $ : num [1:88] 141 140 130 120 120 ...
##  $ : num [1:98] 140 136 132 130 123 ...
##  $ : num [1:93] 135 134 127 126 126 ...
##  $ : num [1:99] 137 133 125 124 124 ...
##  $ : num [1:110] 146 138 132 129 126 ...
##  $ : num [1:104] 151 133 130 127 124 ...
##  $ : num [1:106] 136 135 132 131 130 ...
##  $ : num [1:93] 136 133 131 130 129 ...
##  $ : num [1:103] 142 131 125 124 122 ...
##  $ : num [1:92] 140 135 128 126 125 ...
##  $ : num [1:92] 138 134 134 133 130 ...
##  $ : num [1:89] 138 136 131 125 124 ...
##  $ : num [1:115] 128 128 127 125 123 ...
##  $ : num [1:105] 139 133 130 128 126 ...
##  $ : num [1:92] 150 138 135 125 124 ...
##  $ : num [1:108] 139 132 127 126 126 ...
##  $ : num [1:93] 137 135 132 127 122 ...
##  $ : num [1:95] 139 136 131 128 126 ...
##  $ : num [1:113] 141 135 131 130 129 ...
##  $ : num [1:100] 141 139 134 132 126 ...
##  $ : num [1:100] 135 130 128 126 126 ...
##  $ : num [1:84] 141 134 133 131 129 ...
##  $ : num [1:112] 143 141 140 136 127 ...
##  $ : num [1:96] 128 128 127 127 126 ...
##  $ : num [1:74] 143 140 139 129 127 ...
##  $ : num [1:89] 127 125 122 120 119 ...
##  $ : num [1:83] 137 133 130 127 127 ...
##  $ : num [1:107] 133 132 127 122 122 ...
##  $ : num [1:104] 134 132 128 126 124 ...
##  $ : num [1:102] 138 136 130 128 125 ...
##  $ : num [1:93] 141 136 129 128 127 ...
##  $ : num [1:104] 140 138 135 132 132 ...
##  $ : num [1:84] 136 134 133 131 128 ...
##  $ : num [1:102] 141 133 130 127 126 ...
##  $ : num [1:107] 143 137 123 121 119 ...
##  $ : num [1:104] 138 128 127 126 123 ...
##  $ : num [1:104] 129 126 126 125 123 ...
##  $ : num [1:99] 131 125 124 123 122 ...
##  $ : num [1:94] 139 138 136 136 128 ...
```

```
##  $ : num [1:95] 142 128 126 124 124 ...
##  $ : num [1:106] 155 143 136 135 134 ...
##  $ : num [1:93] 148 139 130 128 127 ...
##  $ : num [1:111] 140 130 126 126 124 ...
##  $ : num [1:92] 132 130 128 126 124 ...
##  $ : num [1:94] 131 127 126 123 123 ...
##  $ : num [1:94] 135 129 124 123 121 ...
##  $ : num [1:109] 128 126 125 125 120 ...
##  $ : num [1:113] 148 146 139 135 129 ...
##  $ : num [1:110] 142 130 129 126 125 ...
##  $ : num [1:103] 135 129 126 124 123 ...
##  $ : num [1:107] 133 130 130 127 126 ...
##  $ : num [1:86] 134 132 130 129 127 ...
##  $ : num [1:107] 144 129 128 128 126 ...
##  $ : num [1:102] 148 134 125 125 124 ...
##  $ : num [1:91] 134 126 123 123 122 ...
##  $ : num [1:80] 133 132 130 125 124 ...
##  $ : num [1:88] 139 139 131 129 129 ...
##  $ : num [1:96] 138 137 129 128 127 ...
##  $ : num [1:116] 127 126 126 124 124 ...
##  $ : num [1:99] 138 135 131 129 125 ...
##  $ : num [1:94] 147 139 136 130 126 ...
##  $ : num [1:88] 131 122 120 118 117 ...
##  $ : num [1:96] 149 134 127 127 126 ...
##  $ : num [1:119] 133 131 126 122 121 ...
##  $ : num [1:91] 132 131 129 125 123 ...
##  $ : num [1:104] 138 135 135 133 130 ...
##  $ : num [1:86] 137 136 132 126 126 ...
##  $ : num [1:109] 141 136 134 133 132 ...
##  $ : num [1:96] 132 128 127 126 124 ...
##  $ : num [1:108] 140 135 131 128 125 ...
##  $ : num [1:104] 140 132 127 126 125 ...
##  $ : num [1:93] 131 126 123 122 120 ...
##  $ : num [1:103] 140 135 134 131 128 ...
##  $ : num [1:95] 134 130 130 128 124 ...
##  $ : num [1:86] 131 125 124 124 123 ...
##  $ : num [1:107] 146 135 134 134 129 ...
##  $ : num [1:105] 134 129 127 126 126 ...
##  $ : num [1:113] 139 136 130 128 128 ...
##  $ : num [1:90] 144 131 130 127 126 ...
##  $ : num [1:100] 137 134 128 125 123 ...
##  $ : num [1:101] 153 150 149 141 139 ...
##  $ : num [1:98] 139 132 128 128 126 ...
##  $ : num [1:95] 139 136 132 128 127 ...
##  $ : num [1:100] 141 131 129 128 128 ...
##  $ : num [1:91] 136 134 134 126 125 ...
##   [list output truncated]
```

```
A.2.b <- unlist(lapply(A.1.d, "[", 2))
str(A.2.b)
```

```
##  num [1:10000] 136 126 129 132 128 ...
```

```
A.mid <- (A.2.b - A.2.s) / 2
str(A.mid)
```

```
##  num [1:10000] 40.2 31.4 27.6 35.8 37.5 ...
```

```
hist(A.mid)
```