R Notebook

Title: "IST687 – Samples HW"
Name: Sathish Kumar Rajendiran
Week: 4
Date: 05/01/2020

Exercise: Let's continue our exploration of sampling.

Step 1: Write a summarizing function to understand the distribution of a vector

```
# Intall moments package
#install.packages('moments')
library(moments)
```

```
# 1. The function, call it 'printVecInfo' should take a vector as input
# 2. The function should print the following information:
#     a. Mean
#     b. Median
#     c. Min & max
#     d. Standard deviation
#     e. Quantiles (at 0.05 and 0.95)
#     f. Skewness
#   Note for skewness, you can use the function in the 'moments' library.
# 3. Test the function with a vector that has (1,2,3,4,5,6,7,8,9,10,50).

      myVector <- c(1,2,3,4,5,6,7,8,9,10,50)

      # define function printVecInfo
      printVecInfo <- function(myVector)
        {
          meaninfo <- mean(myVector)
          cat("mean:",meaninfo,"\n")
          medianinfo <- median(myVector)
          cat("median:",medianinfo,"\n")
          mininfo <- min(myVector)
          # print(paste("min:",mininfo))
          maxinfo <- max(myVector)
          cat("min:",mininfo, "max:",maxinfo,"\n")
          stddevinfo <- sd(myVector)
          cat("sd:",stddevinfo,"\n")
          quantile5percent <- quantile(myVector, probs = 0.05)
          quantile95percent <- quantile(myVector, probs = 0.95)
          cat("quantile (0.05 - 0.95):",quantile5percent,"--",quantile95percent,"\n")
          skewnessinfo <- skewness(myVector)
          cat("skewness:",skewnessinfo,"\n")
```

```r
    }
  printVecInfo(myVector)
```

```
## mean: 9.545455
## median: 6
## min: 1 max: 50
## sd: 13.72125
## quantile (0.05 - 0.95): 1.5 -- 30
## skewness: 2.620396
```

Step 2: Creating Samples in a Jar

```r
# 4. Create a variable 'jar' that has 50 red and 50 blue marbles
# (hint: the jar can have strings as objects, with some of the strings being 'red' and some of the stri

    jar <- c(replicate(50,"red"),replicate(50,"blue"))
    # jar

# 5. Confirm there are 50 reds by summing the samples that are red

    tJar <- table(jar)
    tJar
```

```
## jar
## blue  red
##   50   50
```

```r
    nbrBycolor <- function(v,c)
    {
      # n <- tJar[names(tJar)== c]
      l <- length(v)
      n <- length(grep(c, v))
      cat( "\n number of ",c, "color marble(s) :",n)
      cat( "\n % of ",c, "marble(s) :",n/l*100,"\n")
    }

    nbrBycolor(jar,"red")
```

```
##
##  number of  red color marble(s) : 50
##  % of  red marble(s) : 50
```

```r
    nbrBycolor(jar,"blue")
```

```
##
##  number of  blue color marble(s) : 50
##  % of  blue marble(s) : 50
```

Sampling

```r
#     6. Sample 10 'marbles' (really strings) from the jar. How many are red? What was the
# percentage of red marbles?

    sampleSize <- 10

    sjar <- sample(jar,sampleSize, replace = TRUE)

    nbrBycolor(sjar,"red")
```

```
##
##  number of  red color marble(s) : 5
##  % of  red marble(s) : 50
```

```r
    # nbrBycolor(sjar,"blue")
```

```r
#   7. Do the sampling 20 times, using the 'replicate' command.
#   This should generate a list of 20 numbers.
#   Each number is the mean of how many reds there were in 10 samples.
#   Use your printVecInfo to see information of the samples.
#   Also generate a histogram of the samples.

    ncolor <- function(v,c,s)
      {
        sjar <- sample(v,s,replace = TRUE)
        n <- length(grep(c, sjar))
        return(n)
      }

    ncolor(jar,"red",10)  # how many reds there were in 10 samples.
```

```
## [1] 4
```

```r
    x <- replicate(20,mean(replicate(10,ncolor(jar,"red",10),simplify = TRUE)),simplify = TRUE)

    cat("\n")
```

```r
    x
```

```
##  [1] 5.4 4.7 5.0 4.7 4.9 5.6 5.6 4.3 5.5 4.8 5.9 5.6 5.1 5.4 4.8 4.5 5.6 5.4 4.6
## [20] 5.2
```

```r
    printVecInfo(x)
```

```
## mean: 5.13
## median: 5.15
## min: 4.3 max: 5.9
## sd: 0.4508472
## quantile (0.05 - 0.95): 4.49 -- 5.615
## skewness: -0.1379543
```

```r
    # hist(x)
```

```r
# 8. Repeat #7,
# but this time, sample the jar 100 times.
# You should get 20 numbers, this time each number represents
# the mean of how many reds there were in the 100 samples.
# Use your printVecInfo to see information of the samples.
# Also generate a histogram of the samples.

      ncolor(jar,"red",100)  # how many reds there were in 10 samples.
```

```
## [1] 57
```

```r
      newX <- replicate(20,mean(replicate(100,ncolor(jar,"red",100),simplify = TRUE)),simplify = TRUE)

      cat("\n")
```

```r
      newX
```

```
##  [1] 49.73 49.63 49.91 50.60 50.42 50.47 50.44 50.18 49.83 49.86 49.80 50.15
## [13] 49.61 49.65 49.23 49.80 50.50 50.56 49.77 49.85
```

```r
      printVecInfo(newX)
```

```
## mean: 49.9995
## median: 49.855
## min: 49.23 max: 50.6
## sd: 0.3878887
## quantile (0.05 - 0.95): 49.591 -- 50.562
## skewness: 0.1149311
```

```r
      # hist(newX)
```

```r
# 9. Repeat #8,
# but this time, replicate the sampling 100 times.
# You should get 100 numbers,
# this time each number represents the mean of how many reds there were in the 100 samples.
# Use your printVecInfo to see information of the samples.
# Also generate a histogram of the samples.

      ncolor(jar,"red",100)  # how many reds there were in 10 samples.
```

```
## [1] 44
```

```r
      brandnewX <- replicate(100,mean(replicate(100,ncolor(jar,"red",100),simplify = TRUE)),simplify = '

      cat("\n")
```

```r
brandnewX
```

```
##   [1] 49.31 49.14 49.89 49.61 49.47 50.55 49.27 49.54 49.48 49.88 50.47 50.39
##  [13] 49.63 50.38 49.79 49.72 50.33 49.93 50.61 49.71 50.85 50.18 49.63 49.46
##  [25] 50.65 49.78 51.11 50.36 51.05 50.45 49.54 49.92 50.16 49.74 50.02 49.37
##  [37] 49.91 49.82 49.87 50.40 50.81 49.71 49.64 50.73 49.52 49.36 50.79 49.75
##  [49] 50.38 50.08 49.99 49.47 50.94 49.82 50.09 49.51 49.56 50.92 50.77 49.85
##  [61] 50.30 50.21 50.33 49.90 49.75 50.65 50.39 49.78 50.44 51.13 50.80 49.33
##  [73] 50.97 49.84 49.04 50.79 49.94 51.38 50.25 50.27 49.59 50.16 49.41 50.12
##  [85] 49.71 50.03 50.57 50.83 49.64 50.50 49.78 50.35 49.78 50.09 50.12 50.12
##  [97] 50.64 49.68 48.95 50.25
```

```r
printVecInfo(brandnewX)
```

```
## mean: 50.0687
## median: 50.005
## min: 48.95 max: 51.38
## sd: 0.5233568
## quantile (0.05 - 0.95): 49.329 -- 50.9415
## skewness: 0.2704251
```

```r
# hist(brandnewX)
```

Step 3: Explore the airquality dataset

```r
# 10. Store the 'airquality' dataset into a temporary variable
# 11. Clean the dataset (i.e. remove the NAs)
# 12. Explore Ozone, Wind and Temp by doing a 'printVecInfo' on each as well as
# generating a histogram for each

?airquality

# New York Air Quality Measurements
# Description
# Daily air quality measurements in New York, May to September 1973.
#
# Usage
# airquality

# 10. Store the 'airquality' dataset into a temporary variable

myAirquality <- airquality
str(myAirquality)
```

```
## 'data.frame':    153 obs. of  6 variables:
##  $ Ozone  : int  41 36 12 18 NA 28 23 19 8 NA ...
##  $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
##  $ Wind   : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
##  $ Temp   : int  67 72 74 62 56 66 65 59 61 69 ...
##  $ Month  : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ Day    : int  1 2 3 4 5 6 7 8 9 10 ...
```

```r
head(myAirquality)
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 5    NA      NA 14.3   56     5   5
## 6    28      NA 14.9   66     5   6
```

```r
tail(myAirquality)
```

```
##     Ozone Solar.R Wind Temp Month Day
## 148    14      20 16.6   63     9  25
## 149    30     193  6.9   70     9  26
## 150    NA     145 13.2   77     9  27
## 151    14     191 14.3   75     9  28
## 152    18     131  8.0   76     9  29
## 153    20     223 11.5   68     9  30
```

```r
# 11. Clean the dataset (i.e. remove the NAs)

myAirquality <- na.omit(myAirquality)

myAirquality
```

```
##     Ozone Solar.R Wind Temp Month Day
## 1      41     190  7.4   67     5   1
## 2      36     118  8.0   72     5   2
## 3      12     149 12.6   74     5   3
## 4      18     313 11.5   62     5   4
## 7      23     299  8.6   65     5   7
## 8      19      99 13.8   59     5   8
## 9       8      19 20.1   61     5   9
## 12     16     256  9.7   69     5  12
## 13     11     290  9.2   66     5  13
## 14     14     274 10.9   68     5  14
## 15     18      65 13.2   58     5  15
## 16     14     334 11.5   64     5  16
## 17     34     307 12.0   66     5  17
## 18      6      78 18.4   57     5  18
## 19     30     322 11.5   68     5  19
## 20     11      44  9.7   62     5  20
## 21      1       8  9.7   59     5  21
## 22     11     320 16.6   73     5  22
## 23      4      25  9.7   61     5  23
## 24     32      92 12.0   61     5  24
## 28     23      13 12.0   67     5  28
## 29     45     252 14.9   81     5  29
## 30    115     223  5.7   79     5  30
## 31     37     279  7.4   76     5  31
```

```
## 38      29      127  9.7   82      6   7
## 40      71      291 13.8   90      6   9
## 41      39      323 11.5   87      6  10
## 44      23      148  8.0   82      6  13
## 47      21      191 14.9   77      6  16
## 48      37      284 20.7   72      6  17
## 49      20       37  9.2   65      6  18
## 50      12      120 11.5   73      6  19
## 51      13      137 10.3   76      6  20
## 62     135      269  4.1   84      7   1
## 63      49      248  9.2   85      7   2
## 64      32      236  9.2   81      7   3
## 66      64      175  4.6   83      7   5
## 67      40      314 10.9   83      7   6
## 68      77      276  5.1   88      7   7
## 69      97      267  6.3   92      7   8
## 70      97      272  5.7   92      7   9
## 71      85      175  7.4   89      7  10
## 73      10      264 14.3   73      7  12
## 74      27      175 14.9   81      7  13
## 76       7       48 14.3   80      7  15
## 77      48      260  6.9   81      7  16
## 78      35      274 10.3   82      7  17
## 79      61      285  6.3   84      7  18
## 80      79      187  5.1   87      7  19
## 81      63      220 11.5   85      7  20
## 82      16        7  6.9   74      7  21
## 85      80      294  8.6   86      7  24
## 86     108      223  8.0   85      7  25
## 87      20       81  8.6   82      7  26
## 88      52       82 12.0   86      7  27
## 89      82      213  7.4   88      7  28
## 90      50      275  7.4   86      7  29
## 91      64      253  7.4   83      7  30
## 92      59      254  9.2   81      7  31
## 93      39       83  6.9   81      8   1
## 94       9       24 13.8   81      8   2
## 95      16       77  7.4   82      8   3
## 99     122      255  4.0   89      8   7
## 100     89      229 10.3   90      8   8
## 101    110      207  8.0   90      8   9
## 104     44      192 11.5   86      8  12
## 105     28      273 11.5   82      8  13
## 106     65      157  9.7   80      8  14
## 108     22       71 10.3   77      8  16
## 109     59       51  6.3   79      8  17
## 110     23      115  7.4   76      8  18
## 111     31      244 10.9   78      8  19
## 112     44      190 10.3   78      8  20
## 113     21      259 15.5   77      8  21
## 114      9       36 14.3   72      8  22
## 116     45      212  9.7   79      8  24
## 117    168      238  3.4   81      8  25
## 118     73      215  8.0   86      8  26
```

```
## 120    76      203  9.7    97      8   28
## 121   118      225  2.3    94      8   29
## 122    84      237  6.3    96      8   30
## 123    85      188  6.3    94      8   31
## 124    96      167  6.9    91      9    1
## 125    78      197  5.1    92      9    2
## 126    73      183  2.8    93      9    3
## 127    91      189  4.6    93      9    4
## 128    47       95  7.4    87      9    5
## 129    32       92 15.5    84      9    6
## 130    20      252 10.9    80      9    7
## 131    23      220 10.3    78      9    8
## 132    21      230 10.9    75      9    9
## 133    24      259  9.7    73      9   10
## 134    44      236 14.9    81      9   11
## 135    21      259 15.5    76      9   12
## 136    28      238  6.3    77      9   13
## 137     9       24 10.9    71      9   14
## 138    13      112 11.5    71      9   15
## 139    46      237  6.9    78      9   16
## 140    18      224 13.8    67      9   17
## 141    13       27 10.3    76      9   18
## 142    24      238 10.3    68      9   19
## 143    16      201  8.0    82      9   20
## 144    13      238 12.6    64      9   21
## 145    23       14  9.2    71      9   22
## 146    36      139 10.3    81      9   23
## 147     7       49 10.3    69      9   24
## 148    14       20 16.6    63      9   25
## 149    30      193  6.9    70      9   26
## 151    14      191 14.3    75      9   28
## 152    18      131  8.0    76      9   29
## 153    20      223 11.5    68      9   30
```

```
      myAirquality$Ozone
```

```
##   [1]  41  36  12  18  23  19   8  16  11  14  18  14  34   6  30  11   1  11
##  [19]   4  32  23  45 115  37  29  71  39  23  21  37  20  12  13 135  49  32
##  [37]  64  40  77  97  97  85  10  27   7  48  35  61  79  63  16  80 108  20
##  [55]  52  82  50  64  59  39   9  16 122  89 110  44  28  65  22  59  23  31
##  [73]  44  21   9  45 168  73  76 118  84  85  96  78  73  91  47  32  20  23
##  [91]  21  24  44  21  28   9  13  46  18  13  24  16  13  23  36   7  14  30
## [109]  14  18  20
```

```
# 12. Explore Ozone, Wind and Temp by doing a 'printVecInfo'
      colnames(myAirquality)
```

```
## [1] "Ozone"   "Solar.R" "Wind"    "Temp"    "Month"   "Day"
```

```
      ozoneAir <- myAirquality$Ozone
      windAir <- myAirquality$Wind
      tempAir <- myAirquality$Temp
```

```r
printVecInfo(ozoneAir)
```

```
## mean: 42.0991
## median: 31
## min: 1 max: 168
## sd: 33.27597
## quantile (0.05 - 0.95): 8.5 -- 109
## skewness: 1.248104
```

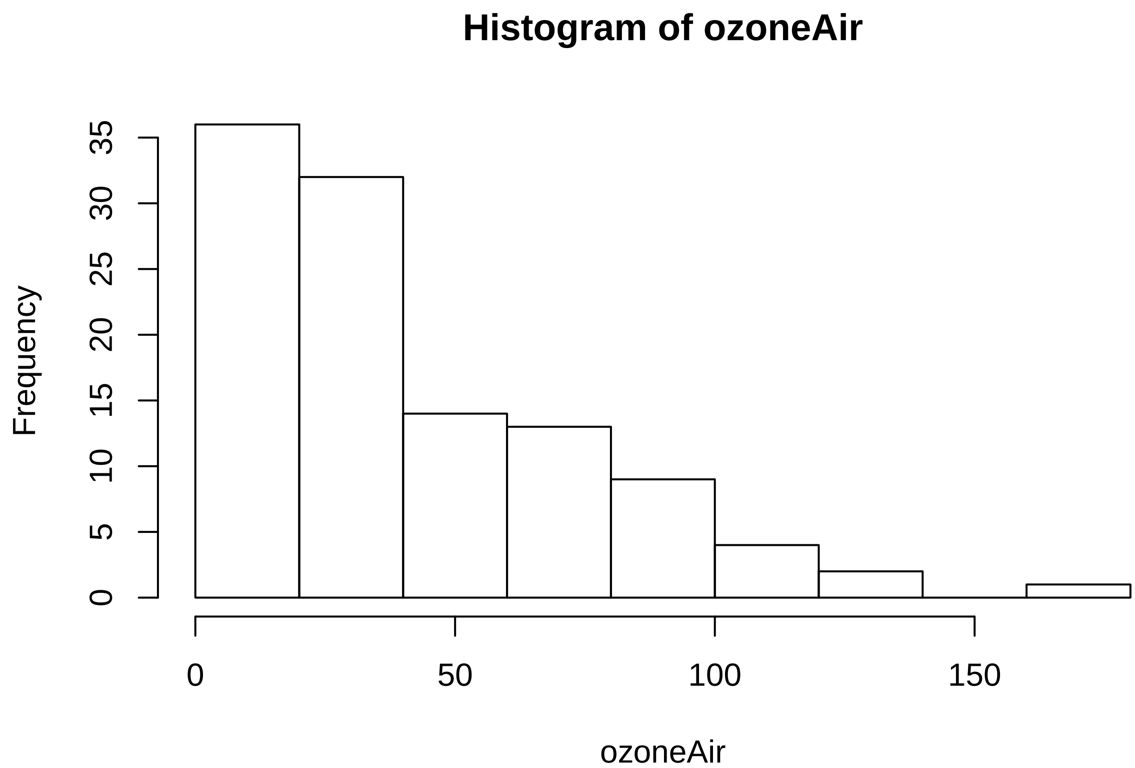
```r
printVecInfo(windAir)
```

```
## mean: 9.93964
## median: 9.7
## min: 2.3 max: 20.7
## sd: 3.557713
## quantile (0.05 - 0.95): 4.6 -- 15.5
## skewness: 0.4556414
```

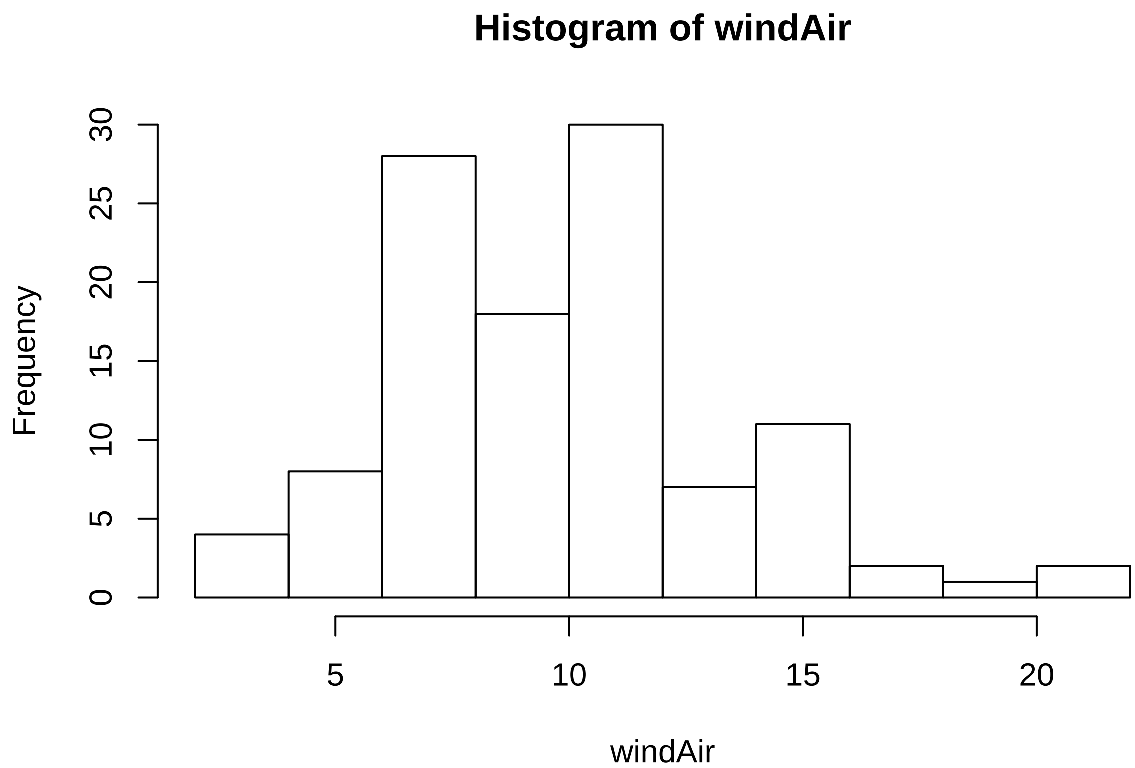
```r
printVecInfo(tempAir)
```

```
## mean: 77.79279
## median: 79
## min: 57 max: 97
## sd: 9.529969
## quantile (0.05 - 0.95): 61 -- 92.5
## skewness: -0.2250959
```

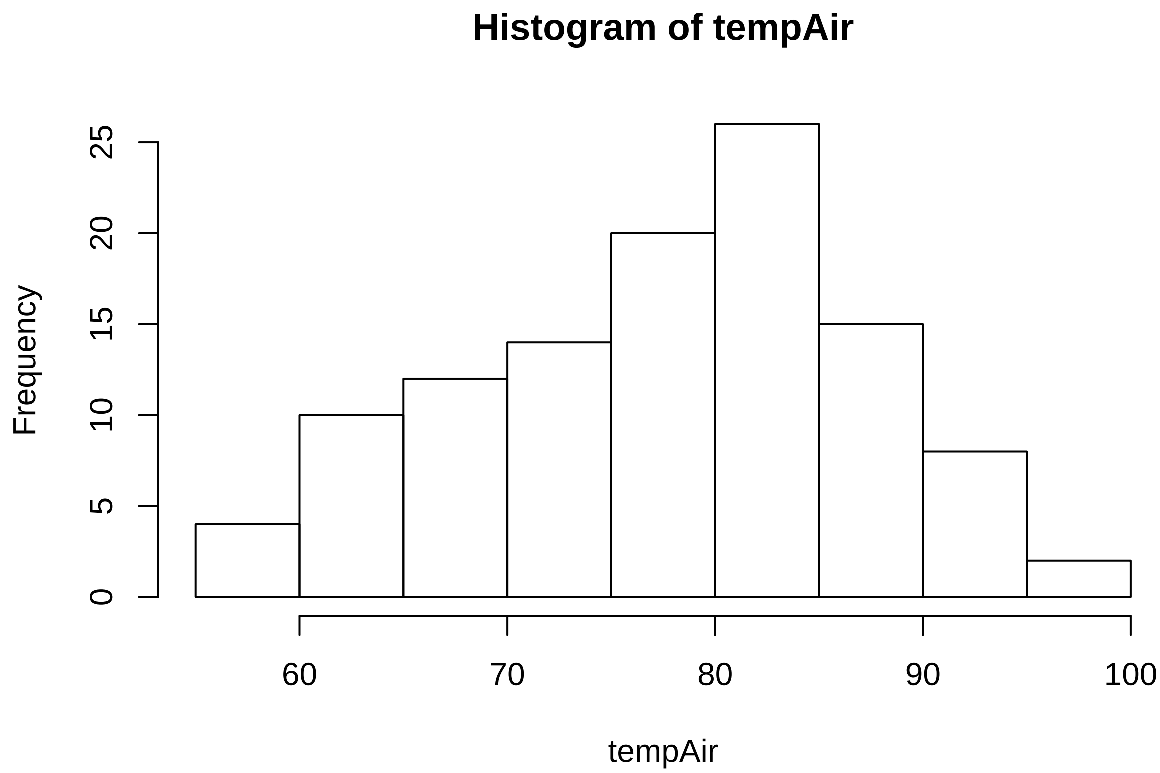
```r
hist(ozoneAir)
```




```r
hist(windAir)
```




```r
hist(tempAir)
```


Histogram of tempAir
Frequency
0
5
10
15
20
25
60
70
80
90
100
tempAir