Homework 05 Functions and Permutation Tests

Due by 11:59pm, Saturday, February 24, 2024, 11:59pm

S&DS 230/530/ENV 757

This homework uses data from both the 2017 and 2018 New Haven Road Races - in particular, we look at 5k times. You can get data for 2018 HERE and for 2017 HERE.

1) Function for Data Cleaning *(25 points)*

a) *(2 pts)* Load in both .csv files into objects called `nh2017` and `nh2018`.

```
nh2017 <- read.csv("http://reuningscherer.net/s&ds230/data/NHRR2017.csv", as.is = TRUE)
nh2018 <- read.csv("http://reuningscherer.net/s&ds230/data/NHRR2018.csv", as.is = TRUE)
```

(1.1) *(5 pts)* Use `head()`, `names()`, and `str()` to check if both datasets have the same variable names and the same format (i.e does each variable have the same format in each dataset). Comment on what you observe.

```
head(nh2017)
```

```
##    No.            Name      City    Div  Time Pace Nettime
## 1 3376  Patrick Dooley  Brooklyn M30-39 15:17 4:56   15:16
## 2 2884     Calvin Park  Trumbull M20-29 15:19 4:56   15:18
## 3 2839  Jake Duckworth    Monroe M20-29 15:29 4:59   15:28
## 4 1150  Scott Rodilitz New Haven M20-29 15:37 5:02   15:36
## 5 1567   Robert Dillon   Shelton M13-19 15:47 5:05   15:46
## 6 4256 Nicholas Migani  Higganum M20-29 16:00 5:09   15:59
```

```
head(nh2018)
```

```
##    No.            Name        City    Div  Time Pace Nettime
## 1 4606 Matthew Farrell Glastonbury M13-19 15:19 4:56   15:19
## 2 2643   Robert Dillon     Shelton M13-19 15:38 5:02   15:38
## 3 4037    Azaan Dawson   New Haven M13-19 15:51 5:07   15:51
## 4 3712   Travis Martin   New Haven M13-19 16:03 5:10   16:00
## 5 4633   Mustafe Dahir Wallingford M13-19 16:19 5:15   16:17
## 6 2731       Ethan Puc   Naugatuck M13-19 16:27 5:18   16:25
```

```
names(nh2018)
```

```
## [1] "No."     "Name"    "City"    "Div"     "Time"    "Pace"    "Nettime"
```

```
names(nh2017)
```

```
## [1] "No."     "Name"    "City"    "Div"     "Time"    "Pace"    "Nettime"
```

```
str(nh2017)
```

```
## 'data.frame':    2736 obs. of  7 variables:
##  $ No.    : int  3376 2884 2839 1150 1567 4256 3963 4307 5131 5740 ...
##  $ Name   : chr  "Patrick Dooley" "Calvin Park" "Jake Duckworth" "Scott Rodilitz" ...
##  $ City   : chr  "Brooklyn" "Trumbull" "Monroe" "New Haven" ...
##  $ Div    : chr  "M30-39" "M20-29" "M20-29" "M20-29" ...
```

```
##  $ Time   : chr  "15:17" "15:19" "15:29" "15:37" ...
##  $ Pace   : chr  "4:56" "4:56" "4:59" "5:02" ...
##  $ Nettime: chr  "15:16" "15:18" "15:28" "15:36" ...
```

```
str(nh2018)
```

```
## 'data.frame':    2685 obs. of  7 variables:
##  $ No.    : int  4606 2643 4037 3712 4633 2731 4800 3710 4618 3142 ...
##  $ Name   : chr  "Matthew Farrell" "Robert Dillon" "Azaan Dawson" "Travis Martin" ...
##  $ City   : chr  "Glastonbury" "Shelton" "New Haven" "New Haven" ...
##  $ Div    : chr  "M13-19" "M13-19" "M13-19" "M13-19" ...
##  $ Time   : chr  "15:19" "15:38" "15:51" "16:03" ...
##  $ Pace   : chr  "4:56" "5:02" "5:07" "5:10" ...
##  $ Nettime: chr  "15:19" "15:38" "15:51" "16:00" ...
```

These data sets appear to have the same structure, matching columns and formats

(1.2) *(18 pts)* Since the two datasets seem to have the same structure, we can write a function that creates new variables in each dataset. This function will be called `cleanNHData()`. As a first step, I've already included code to load the `lubridate` package and define a function called `convertTimes()` similar to that we used in Class 10.

I've started the outline of the function below. Your job is to follow the exact process we used in class 9 to clean the 2018 data. You need to replace each comment line in the `cleanNHData()` function with the code that will perform this task. You literally just need to find the relevant line in the class code and put this into the `cleanNHData()` function. The one exception is a new line you'll need to write that deletes rows where `Name` is missing (i.e. equal to " ")

Then, run the function on `nh2017` and `nh2018`.

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
convertTimes <- function(v) {
  hourplus <- nchar(v) == 7
  wrongformat <- nchar(v) == 8
  outtimes <- ms(v)
  if (sum(hourplus) > 0) { # if there is at least 1 time that exceeds 1 hr
    outtimes[hourplus] <- hms(v[hourplus])
  }
  if (sum(wrongformat) > 0) { # if there is at least 1 time in wrong format
    outtimes[wrongformat] <- ms(substr(v[wrongformat],1,5))
  }
  outtimes <- as.numeric(outtimes)/60
  return(outtimes)
}

cleanNHData <- function(data) {
  #Replace Div = "" with NAz
  data$Div[data$Div == ""] <- NA
  #Make a dataset variable called Gender from the variable Div
  data$Gender <- substr(data$Div, 1, 1)
```

```
  #Make a dataset variable called AgeGrp from the variable Dif
  data$AgeGrp <- substr(data$Div, 2, nchar(data$Div))
  #Make a dataset variable called Nettime_min using the convertTimes function
  data$Nettime_min <- convertTimes(data$Nettime)
  #Make a dataset variable called Time_min using the convertTimes function
  data$Time_min <- convertTimes(data$Time)
  #Make a dataset variable called Pace_min using the convertTimes function
  data$Pace_min <- convertTimes(data$Pace)
  #Replace dataset with same dataset such that Name is not equal to ""

  data <- data[data$Name != "", ]
  #Return the dataset
  return(data)
}

#run cleanNHData on nh2018 and nh2017
nh2017 <- cleanNHData(nh2017)
```

```
## Warning in .parse_hms(..., order = "MS", quiet = quiet): Some strings failed to
## parse

## Warning in .parse_hms(..., order = "MS", quiet = quiet): Some strings failed to
## parse
```

```
nh2018 <- cleanNHData(nh2018)
```

```
## Warning in .parse_hms(..., order = "MS", quiet = quiet): Some strings failed to
## parse

## Warning in .parse_hms(..., order = "MS", quiet = quiet): Some strings failed to
## parse

## Warning in .parse_hms(..., order = "MS", quiet = quiet): Some strings failed to
## parse
```

```
head(nh2017)
```

```
##    No.            Name      City    Div  Time Pace Nettime Gender AgeGrp
## 1 3376  Patrick Dooley  Brooklyn M30-39 15:17 4:56   15:16      M  30-39
## 2 2884     Calvin Park  Trumbull M20-29 15:19 4:56   15:18      M  20-29
## 3 2839  Jake Duckworth    Monroe M20-29 15:29 4:59   15:28      M  20-29
## 4 1150  Scott Rodilitz New Haven M20-29 15:37 5:02   15:36      M  20-29
## 5 1567   Robert Dillon   Shelton M13-19 15:47 5:05   15:46      M  13-19
## 6 4256 Nicholas Migani  Higganum M20-29 16:00 5:09   15:59      M  20-29
##   Nettime_min Time_min Pace_min
## 1    15.26667 15.28333 4.933333
## 2    15.30000 15.31667 4.933333
## 3    15.46667 15.48333 4.983333
## 4    15.60000 15.61667 5.033333
## 5    15.76667 15.78333 5.083333
## 6    15.98333 16.00000 5.150000
```

```
head(nh2018)
```

```
##    No.            Name        City    Div  Time Pace Nettime Gender AgeGrp
## 1 4606 Matthew Farrell Glastonbury M13-19 15:19 4:56   15:19      M  13-19
```

```
## 2 2643   Robert Dillon     Shelton M13-19 15:38 5:02   15:38      M  13-19
## 3 4037    Azaan Dawson   New Haven M13-19 15:51 5:07   15:51      M  13-19
## 4 3712   Travis Martin   New Haven M13-19 16:03 5:10   16:00      M  13-19
## 5 4633   Mustafe Dahir Wallingford M13-19 16:19 5:15   16:17      M  13-19
## 6 2731       Ethan Puc   Naugatuck M13-19 16:27 5:18   16:25      M  13-19
##   Nettime_min Time_min Pace_min
## 1    15.31667 15.31667 4.933333
## 2    15.63333 15.63333 5.033333
## 3    15.85000 15.85000 5.116667
## 4    16.00000 16.05000 5.166667
## 5    16.28333 16.31667 5.250000
## 6    16.41667 16.45000 5.300000
```

(1.3) *(2 pts)* Use `str()` to check if the datasets have the same format now. Comment on what you observe.

```
str(nh2017)
```

```
## 'data.frame':    2727 obs. of  12 variables:
##  $ No.        : int  3376 2884 2839 1150 1567 4256 3963 4307 5131 5740 ...
##  $ Name       : chr  "Patrick Dooley" "Calvin Park" "Jake Duckworth" "Scott Rodilitz" ...
##  $ City       : chr  "Brooklyn" "Trumbull" "Monroe" "New Haven" ...
##  $ Div        : chr  "M30-39" "M20-29" "M20-29" "M20-29" ...
##  $ Time       : chr  "15:17" "15:19" "15:29" "15:37" ...
##  $ Pace       : chr  "4:56" "4:56" "4:59" "5:02" ...
##  $ Nettime    : chr  "15:16" "15:18" "15:28" "15:36" ...
##  $ Gender     : chr  "M" "M" "M" "M" ...
##  $ AgeGrp     : chr  "30-39" "20-29" "20-29" "20-29" ...
##  $ Nettime_min: num  15.3 15.3 15.5 15.6 15.8 ...
##  $ Time_min   : num  15.3 15.3 15.5 15.6 15.8 ...
##  $ Pace_min   : num  4.93 4.93 4.98 5.03 5.08 ...
```

```
str(nh2018)
```

```
## 'data.frame':    2685 obs. of  12 variables:
##  $ No.        : int  4606 2643 4037 3712 4633 2731 4800 3710 4618 3142 ...
##  $ Name       : chr  "Matthew Farrell" "Robert Dillon" "Azaan Dawson" "Travis Martin" ...
##  $ City       : chr  "Glastonbury" "Shelton" "New Haven" "New Haven" ...
##  $ Div        : chr  "M13-19" "M13-19" "M13-19" "M13-19" ...
##  $ Time       : chr  "15:19" "15:38" "15:51" "16:03" ...
##  $ Pace       : chr  "4:56" "5:02" "5:07" "5:10" ...
##  $ Nettime    : chr  "15:19" "15:38" "15:51" "16:00" ...
##  $ Gender     : chr  "M" "M" "M" "M" ...
##  $ AgeGrp     : chr  "13-19" "13-19" "13-19" "13-19" ...
##  $ Nettime_min: num  15.3 15.6 15.8 16 16.3 ...
##  $ Time_min   : num  15.3 15.6 15.8 16.1 16.3 ...
##  $ Pace_min   : num  4.93 5.03 5.12 5.17 5.25 ...
```

the datasets still have the same format, with the same number of variables and the same variable types

2) Repeat Runners Dataset *(38 points)*

We now create a dataset that looks at times of runners who ran in both 2018 and 2017.

(2.1) *(5 pts)* We'll have problems if we have instances of two runners having the same name. A crude fix is to delete the second occurance of anyone with a duplicate name.

Run the code below to see how the function `duplicated()` works:

```
duplicated(c("cat","cat","dog","llama"))
```

```
## [1] FALSE  TRUE FALSE FALSE
```

Esentially, this returns a vector that is `FALSE` if an observation value is the first occurrence of this value and `TRUE` when a value has been seen before.

To merge our two datasets, we need to start with unique `Name` values in each dataset. Using the `duplicated()` function, create two new dataframes called `nh2018Unq` and `nh2017Unq` so that each only retains observations for the first occurence of each value of `Name` (if you use the `!` operator, this is two short lines of code).

Get the dimensions of each of the four relevant dataframes. How many observations were eliminated from each year?

```
nh2017Unq <- nh2017[!duplicated(nh2017$Name),]
nh2018Unq <- nh2018[!duplicated(nh2018$Name),]
dim(nh2017)
```

```
## [1] 2727   12
```

```
dim(nh2017Unq)
```

```
## [1] 2720   12
```

```
dim(nh2018)
```

```
## [1] 2685   12
```

```
dim(nh2018Unq)
```

```
## [1] 2640   12
```

In 2017, we went from 2727 to 2720, eliminating 7, and 2685 to 2640 for 45 deleted in 2018

(2.2) *(5 pts)* Next, we need to get a list of names that occur in both datasets. Run the code below to see how the `intersect()` function works.

```
intersect(c("cat", "dog", "llama"), c("cat","llama","chincilla"))
```

```
## [1] "cat"   "llama"
```

Using the `intersect()` function, create an object called `repeatrunners` that is a list of names of people who ran in both years. How many runners ran in both years?

```
repeatrunners <- intersect(nh2017Unq$Name, nh2018Unq$Name)
length(repeatrunners)
```

```
## [1] 986
```

there were 986 repeat runners (2.3) *(18 pts)* The code below will create a combined dataset called `nhcombined`. Your job in this section is to write a one or two line comment above each line of code to describe what the line does. You'll want to run each line, probably see what the result was, and in some cases use the help file for some functions to see what the function does (i.e. for the `merge()` function). Make sure you remove `eval = FALSE` in the r chunk.

```
# create vector w that is TRUE if the Name is in repeatrunners and FALSE otherwise
w <- nh2018Unq$Name %in% repeatrunners

# create a new dataframe nhcombined that is a subset of nh2018Unq that only includes the rows where w i
nhcombined <- data.frame(Name = nh2018Unq$Name[w],
                         Gender = nh2018Unq$Gender[w],
                         Nettime_2018 = nh2018Unq$Nettime_min[w])
```

```r
# merge nhcombined with nh2017Unq, only including the Name and Nettime_min columns
nhcombined <- merge(nhcombined, nh2017Unq[, c("Name", "Nettime_min")])

# remove rows where the gender is missing from nhcombined
nhcombined <- nhcombined[!is.na(nhcombined$Gender),]

# rename the Nettime_min column to Nettime_2017
colnames(nhcombined)[4] <- "Nettime_2017"

# gives the dimensions of nhcombined
dim(nhcombined)
```

```
## [1] 985   4
```

```r
# gives the first 6 rows of nhcombined
head(nhcombined)
```

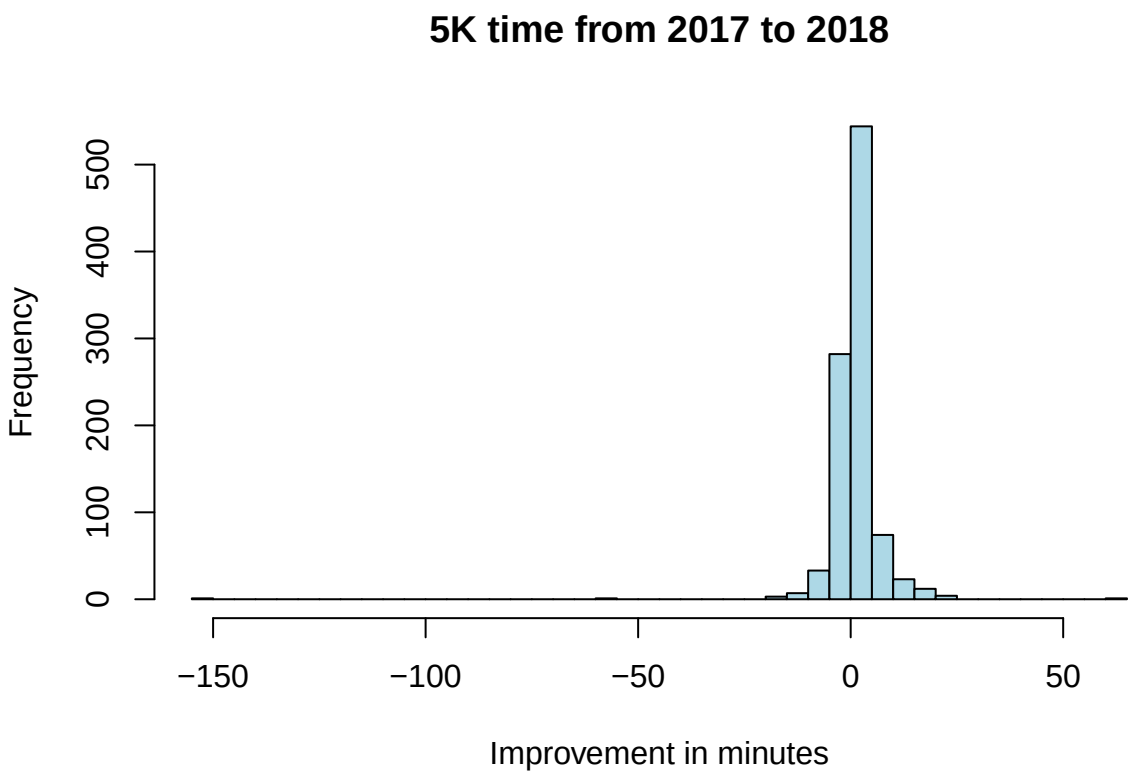
```
##              Name Gender Nettime_2018 Nettime_2017
## 1      Abbey Shaw      F     39.25000     40.25000
## 2     Abby Dziura      F     39.03333     35.63333
## 3      Abby Ganun      F     40.08333     44.65000
## 4     Abi Hawkins      F     35.86667     27.56667
## 5  Abigail Murphy      F     32.88333     34.06667
## 6 Abraham Cordero      M     29.63333     31.83333
```

(2.4) *(6 pts)* Create a new variable in the data frame `nhcombined` called `improvement` that is the improvement in run time from 2017 to 2018 (a positive number here should indicate an improvement,a negative number means they did worse in 2018). Get summary statistics for `nhcombined`. Then make a histogram of `improvement`. Comment on the summary statistics and what you observe in the histogram.

```r
nhcombined$improvement <- nhcombined$Nettime_2018 - nhcombined$Nettime_2017
summary(nhcombined)
```

```
##      Name              Gender           Nettime_2018     Nettime_2017
##  Length:985         Length:985         Min.   : 15.63   Min.   : 15.30
##  Class :character   Class :character   1st Qu.: 26.12   1st Qu.: 25.43
##  Mode  :character   Mode  :character   Median : 30.60   Median : 29.37
##                                        Mean   : 32.04   Mean   : 30.93
##                                        3rd Qu.: 36.28   3rd Qu.: 34.32
##                                        Max.   :132.28   Max.   :188.08
##   improvement
##  Min.   :-150.2667
##  1st Qu.:  -0.5333
##  Median :   0.9333
##  Mean   :   1.1156
##  3rd Qu.:   2.6000
##  Max.   :  64.5167
```

```r
hist(nhcombined$improvement,
breaks = 50,
     main = "5K time from 2017 to 2018",
     xlab = "Improvement in minutes",
      ylab = "Frequency",
      col = "lightblue"
     )
```



The mean change in minutes was 1.1156 minutes. The summary shows that the mean of 1.11556 is greater thatn the median of 0.9333, implying that the data is right-skewed. Also at least a quarter of runners ran slower and at least half improved their times

(2.5) *(4 pts)* You'll notice a few extreme values (i.e. people got amazingly better or worse). Print the rows of `nhcombined` that had improvement times of more than 50 in absolute value. Update the `nhcombined` dataframe to exclude these rows and make the histogram again.

```
print(nhcombined[abs(nhcombined$improvement) > 50,])
```

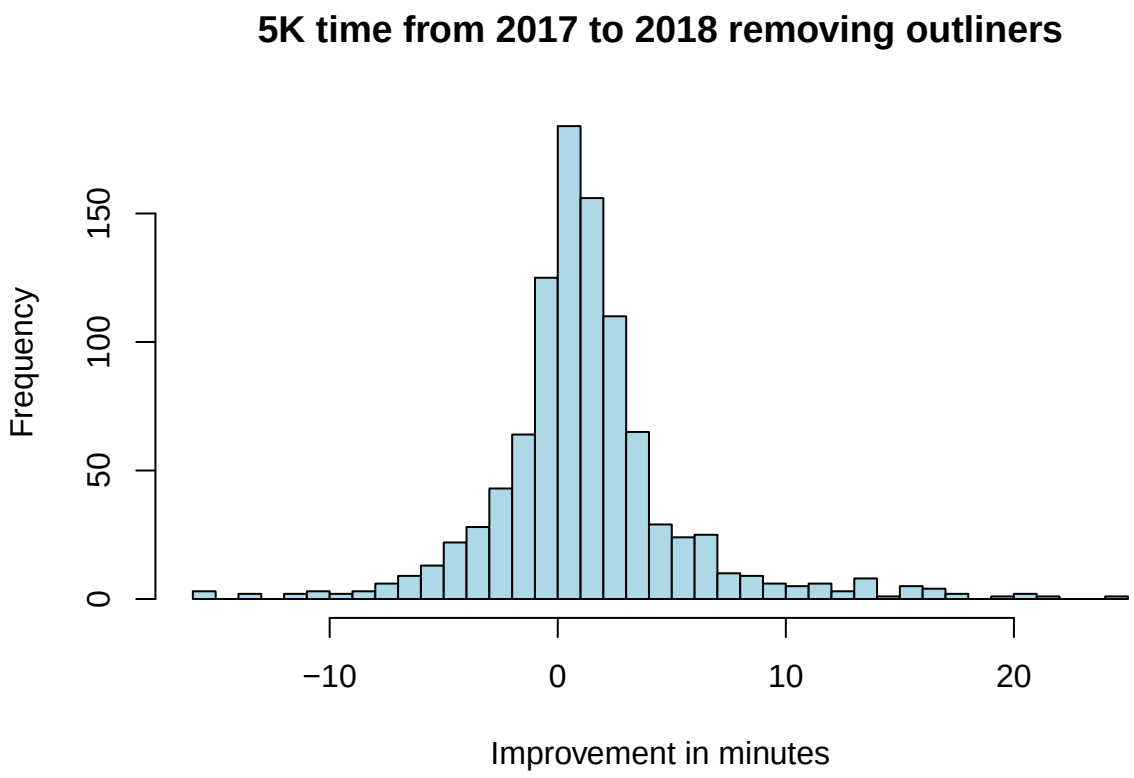
```
##              Name Gender Nettime_2018 Nettime_2017 improvement
## 483  Julius Bloom      M     30.28333     87.41667   -57.13333
## 594   Lina Alpert      F    109.51667     45.00000    64.51667
## 706 Mike Trumbley      M     37.81667    188.08333  -150.26667
```

```
nhcombined <- nhcombined[!(abs(nhcombined$improvement) > 50),]
summary(nhcombined)
```

```
##      Name              Gender           Nettime_2018     Nettime_2017
##  Length:982         Length:982         Min.   : 15.63   Min.   : 15.30
##  Class :character   Class :character   1st Qu.: 26.09   1st Qu.: 25.42
##  Mode  :character   Mode  :character   Median : 30.59   Median : 29.32
##                                        Mean   : 31.96   Mean   : 30.70
##                                        3rd Qu.: 36.25   3rd Qu.: 34.23
##                                        Max.   :132.28   Max.   :130.75
##   improvement
##  Min.   :-15.5000
##  1st Qu.: -0.5333
##  Median :  0.9333
```

```
##  Mean   :  1.2645
##  3rd Qu.:  2.5917
##  Max.   : 24.6167
```

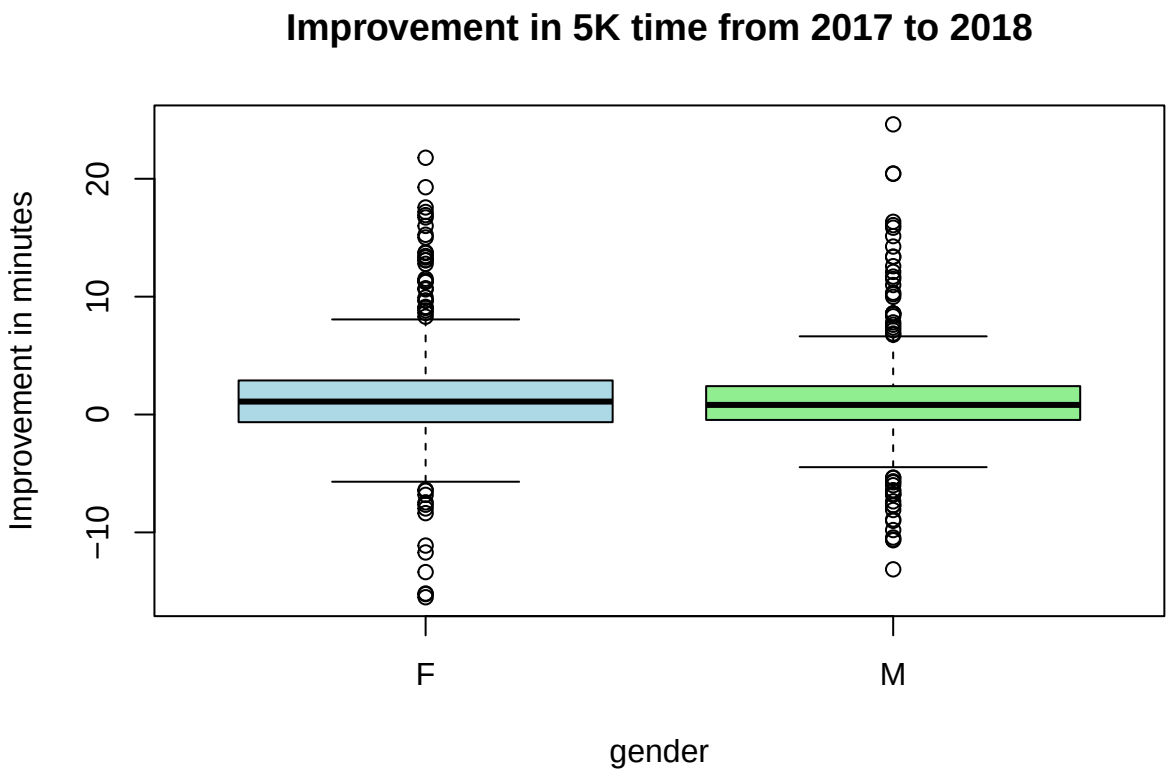
```
hist(nhcombined$improvement,
breaks = 50,
     main = "5K time from 2017 to 2018 removing outliners",
     xlab = "Improvement in minutes",
      ylab = "Frequency",
      col = "lightblue"
     )
```



3) Run Time Improvements *(37 pts)*

(3.1) *(6 pts)* Make a side-by-side boxplot to see differences between improvements between Females and Males. Does there appear to be any difference between groups? Comment both on center and spread.

```
boxplot(nhcombined$improvement ~ nhcombined$Gender,
        main = "Improvement in 5K time from 2017 to 2018",
        xlab = "gender",
        ylab = "Improvement in minutes",
        col = c("lightblue", "lightgreen")
        )
```



The boxlot shows that both genders have similar centers and spread. The median improvement times appear to be very similar, and the spread of the data is also similar, however the females had a slighly wider range. The key distinction I observe is that the upper outliers are more pronounced for males compared to females, while the lower outliers are more extreme for females than males. (3.2) *(16 pts)* Using a 95% bootstrap confidence interval, what can you say about the average improvement among the population of all female repeat 5K runners? Do the same for male repeat 5K runners. You don't need to make any histograms of your bootstrap results, and you don't need to use the `t.test()` function. You also are not comparing the means of these two groups - you're getting seperate intervals for each gender group.

```
# To make grading easier, please leave the following line of code in your assignment
set.seed(230)
n <- 10000
deltaf <- rep(NA, n)
deltam <- rep(NA, n)

for (i in 1:n) {
  deltaf[i] <- mean(sample(nhcombined$improvement[nhcombined$Gender == "F"],
                           sum(nhcombined$Gender == "F"),
                           replace = TRUE))
  deltam[i] <- mean(sample(nhcombined$improvement[nhcombined$Gender == "M"],
                           sum(nhcombined$Gender == "M"),
                           replace = TRUE))
}
(quantile(deltaf, c(0.025, 0.975)))
```

```
##     2.5%    97.5%
## 0.983298 1.767311
```

```r
(quantile(deltam, c(0.025, 0.975)))
```

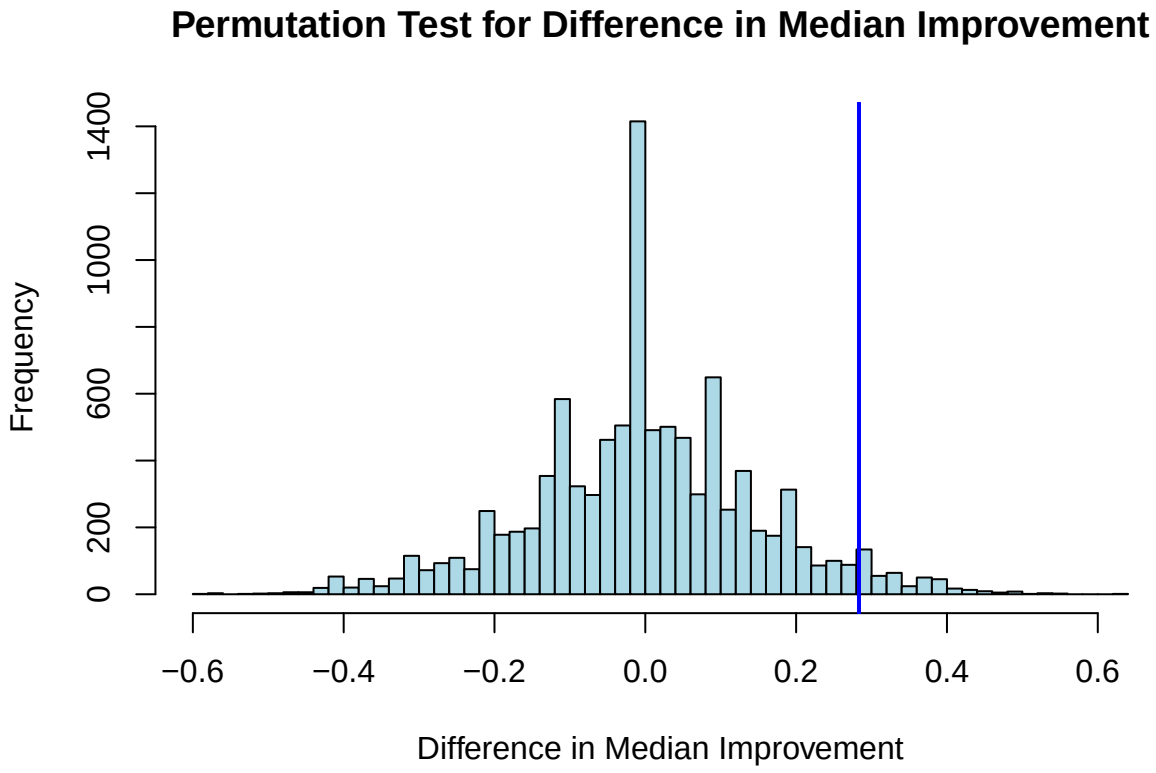
```
##      2.5%     97.5%
## 0.8006109 1.5091042
```

The 95% confidence interval for the average improvement in 5K time from 2017 to 2018 with 95% confidence is 0.983298, 1.767311 for females and 0.8006109, 1.5091042. Since the null hypothesis is an improvement of 0, we can determine that because 0 does not fall between these intervals, there is a statistically signifigant change between times in 2017 and 2018

(3.3) *(15 pts)* Using a permutation test, examine whether there a significant difference in the **MEDIAN** improvement between males and females. Use a significance level of 0.05. Be sure to state (in words is fine) the null and alternative hypotheses, and justify your conclusion. Be sure to include a histogram of results and add a vertical line that shows that observed difference in medians (see example in code from class).

```r
# To make grading easier, please leave the following line of code in your assignment
set.seed(230)

attach(nhcombined)
actualDiff <- by(improvement, Gender, median)
actualDiff <- actualDiff[1] - actualDiff[2]
n <- 10000
diffs <- rep(NA, n)
for(i in 1:n) {
fakeGender <- sample(nhcombined$Gender)
diffs[i] <- median(nhcombined$improvement[fakeGender == "M"]) - median(nhcombined $improvement[fakeGend
}
hist(diffs, breaks = 50, col = "light blue", main = "Permutation Test for Difference in Median Improveme
abline(v = actualDiff, col = "blue", lwd = 2)
```




```
mean(abs(diffs) >= abs(actualDiff))
```

```
## [1] 0.0822
```

The null hypothesis is that ther e is no significant difference in median improvement between males in females from 2017 to 2018. The alternative hypothesis is that there is a significant difference in median improvement between males and females from 2017 to 2018. Given a p value we got from the test is 0.0822 is greater than 0.0500, we fail to reject the null hypothesis. This means that there is no significant difference in median improvement of males and females from 2017 to 2018