Homework 05 Functions and Permutation Tests

Due by 11:59pm, Saturday, February 25, 2023

S&DS 230/530/ENV 757

```
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

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

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")

nh2018 <- read.csv("http://reuningscherer.net/s&ds230/data/NHRR2018.csv")
```

(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(nh2017)
```

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

```
names(nh2018)
```

```
## [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" ...
```

Both the dataset have equal number of variables with same names. All the variables in the dataset have same format and stored in same variable type.However the number of observation is different in both the dataset. Year 2017 dataset has 2736 obs. of 7 variables while year 2018 dataset has 2685 obs. of 7 variables.

(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) {
  data$Div[data$Div == ""] <- NA
  data$Gender <- substr(data$Div, 1, 1)
  data$AgeGrp <- substr(data$Div, 2, nchar(data$Div))
  data$Nettime_min <- convertTimes(data$Nettime)
  data$Time_min <- convertTimes(data$Time)
  data$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
nh2018 <- cleanNHData(nh2018)
```

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

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

```
failed to
## parse, or all strings are NAs

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

```
nh2017 <- cleanNHData(nh2017)
```

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

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

2) Repeat Runners Dataset *(38 points)*

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

(1.1) *(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?

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

dim(nh2018)
```

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

```
dim(nh2018Unq)
```

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

```
dim(nh2017)

## [1] 2727   12

dim(nh2017Unq)

## [1] 2720   12
```

We can see that from the year 2018 ; 45 observations were eliminated. From the year 2017 ; 7 observations were eliminated.

(1.2) *(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(nh2018$Name, nh2017$Name)
length(repeatrunners)

## [1] 986
```

A total of 986 runners ran in both the years.

(1.3) *(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.

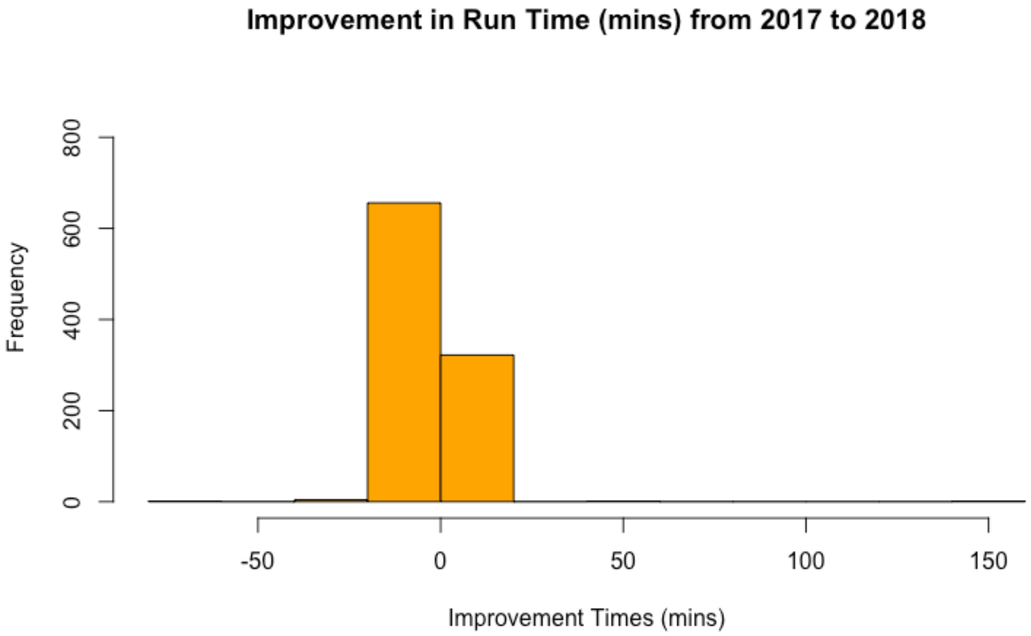
```
# This code creates a vector w with logical value (True or False) of the
names in nh2018Unq dataset who are present in the repeatrunners data i.e who
ran in both the years.
w <- nh2018Unq$Name %in% repeatrunners

# This code creates a dataset having 986 variables and 3 variables. It
includes the the name, gender and nettime_2018 in min for those individuals
who were present in nh2018Unq dataset and people who ran in both years.
nhcombined <- data.frame(Name = nh2018Unq$Name[w],
                         Gender = nh2018Unq$Gender[w],
                         Nettime_2018 = nh2018Unq$Nettime_min[w])

# The merge function merges two data frames by common columns or row names.
This code merges two dataset : nhcomined and nh2017Unq by default with common
name. The new created nhcombined dataset has 986 observations and 4
variables. Additionally, "Name" and ""Nettime_min" are the two suffixes
used.In the dataset we can see that one additional column "Nettime_min" was
also included which is the nettime in the year 2017.
```

```
nhcombined <- merge(nhcombined, nh2017Unq[, c("Name", "Nettime_min")])


# This code gives a dataset of 985 observations and 4 variables. The
observations having missing gender values were removed from the dataset / the
observations where gender was not NA were included in the dataset.
nhcombined <- nhcombined[!is.na(nhcombined$Gender),]

# This code is changing the column name of the fourth column from
'Nettime_min' to 'Nettime_2017'
colnames(nhcombined)[4] <- "Nettime_2017"

# This code give the dimensions (dim) of the nhcombined dataset
dim(nhcombined)

## [1] 985   4

# This code gives the top 6 rows of the nhcombined dataset.
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
```

(1.4) *(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.

```
improvement <- nhcombined$Nettime_2017 - nhcombined$Nettime_2018

nhcombined <- cbind(nhcombined, improvement)
head(nhcombined)

##                Name Gender Nettime_2018 Nettime_2017 improvement
## 1      Abbey Shaw      F     39.25000     40.25000    1.000000
## 2     Abby Dziura      F     39.03333     35.63333   -3.400000
## 3      Abby Ganun      F     40.08333     44.65000    4.566667
## 4     Abi Hawkins      F     35.86667     27.56667   -8.300000
## 5  Abigail Murphy      F     32.88333     34.06667    1.183333
## 6 Abraham Cordero      M     29.63333     31.83333    2.200000

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.   :-64.5167
##  1st Qu.: -2.6000
##  Median : -0.9333
##  Mean   : -1.1156
##  3rd Qu.:  0.5333
##  Max.   :150.2667
```

```
hist(nhcombined$improvement,
     col = "orange",
     main = "Improvement in Run Time (mins) from 2017 to 2018",
     xlab = "Improvement Times (mins)",
     ylim = c(0,900))
```

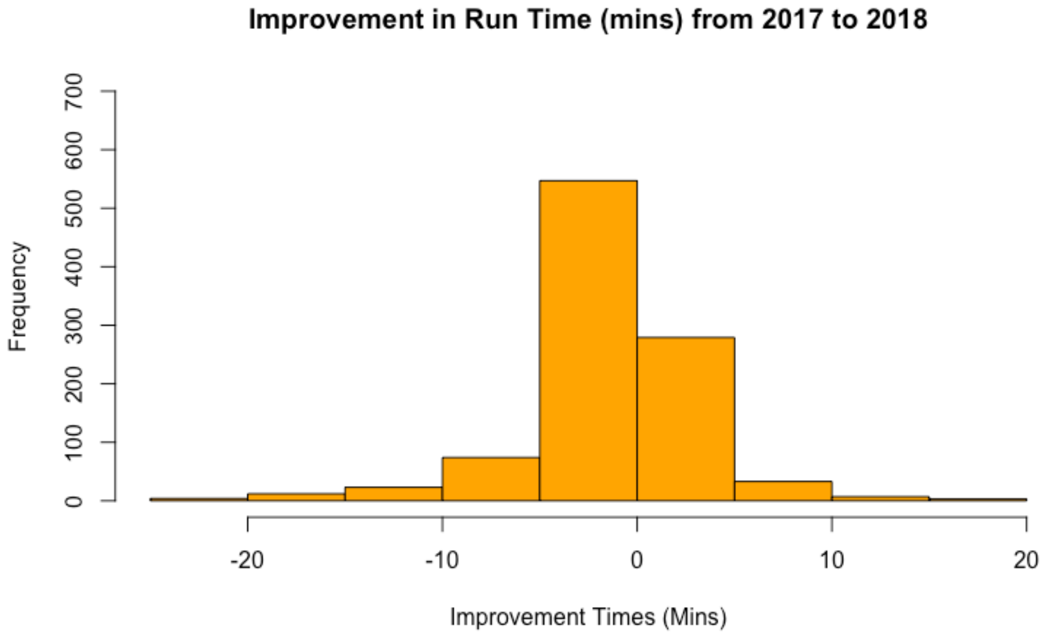


Through the summary statistics we can see that there are 985 observations, with mean nettime in year 2018 to be 32 minutes with a range of 15mins to 132 mins. In the year 2017; the average nettime was 30mins, which approximately equal to the nettime in the year 2018. The range is 15mins to 188 mins. The mean improvement time is -1.11 min which means that the runners did worse in the year 2018 on average.

Through the histogram we can see the same that the runners did worse in the year 2018 on average compared to 2017. About 650 runners had an improvement score of approx -1.0 (worse) and only 300 runners had an improvement score of approx +1.0 (better). We can also *see some outliers who had a score of -50 or +50 (i.e. people got amazingly better or worse).The distribution is right skewed.*

(1.5) *(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.

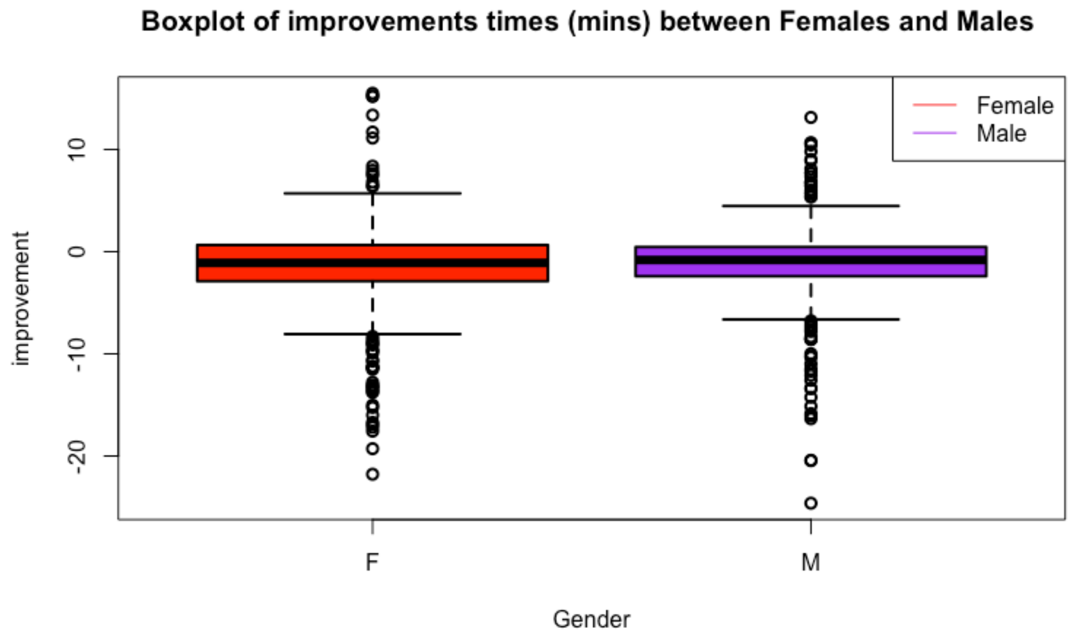
```
(subset <- subset(nhcombined, abs(improvement) > 50))

##                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 <- anti_join(nhcombined, subset)

## Joining with `by = join_by(Name, Gender, Nettime_2018, Nettime_2017,
## improvement)`

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.   :-24.6167
##  1st Qu.: -2.5917
##  Median : -0.9333
##  Mean   : -1.2645
##  3rd Qu.:  0.5333
##  Max.   : 15.5000

dim(nhcombined)

## [1] 982   5

hist(nhcombined$improvement,
     col = "orange",
     main = "Improvement in Run Time (mins) from 2017 to 2018",
     xlab = "Improvement Times (Mins)",
     ylim = c(0,700))
```



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(improvement ~ Gender,
        data = nhcombined,
        lwd = 2,
        main = "Boxplot of improvements times (mins) between Females and
Males ",
        col = c("red","purple"))

legend("topright", legend=c("Female", "Male"),
       col=c("red","purple"), lty=1, cex=1.0)
```



The center in both the boxplots seems equal i.e. around -1.0 while the spread seems to be more among females than the males.The outliers also seems to be visually equal in number.

(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.

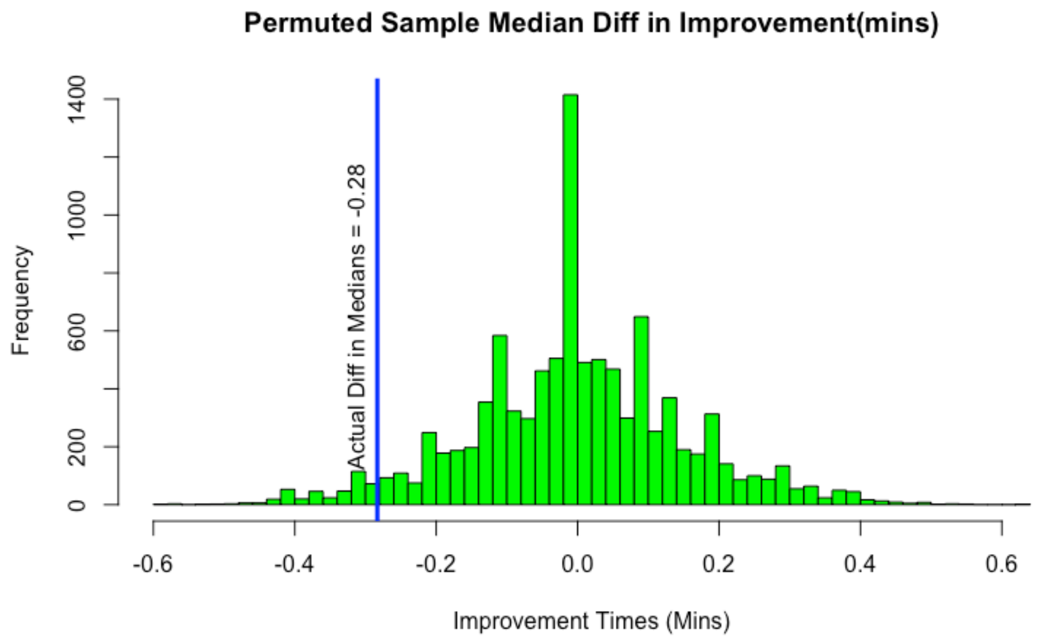
```
# To make grading easier, please leave the following line of code in your
assignment
set.seed(230)

# FILL IN REMAINING CODE


N <-  10000   #number of samples

Fmeanimpvt <-  rep(NA, N)
Mmeanimpvt <- rep(NA, N)

for(i in 1:N){
  SF <- sample(nhcombined$improvement[nhcombined$Gender == "F"],
               sum(nhcombined$Gender == "F"), replace = TRUE)
  Fmeanimpvt[i] <- mean(SF)
}

for(i in 1:N){
  SM <- sample(nhcombined$improvement[nhcombined$Gender == "M"],
              sum(nhcombined$Gender == "M"), replace = TRUE)
  Mmeanimpvt[i] <- mean(SM)
}

ciF <-  quantile(Fmeanimpvt, c(.025, .975))
ciM <-  quantile(Mmeanimpvt, c(.025, .975))

round(ciF,2)

##  2.5% 97.5%
## -1.77 -0.99

round(ciM,2)

##  2.5% 97.5%
## -1.51 -0.81
```

95% Bootstrap confidence interval of improvement among females was (-1.77,-0.99).The negative scores states that the runners got worse in the year 2018 compared to 2017. The confidence interval did not include a zero which suggests that we reject the null hypothesis. There was a difference in the average improvement among the population of all female repeat 5K runners from year 2017 to year 2018.

95% Bootstrap confidence interval of improvement among males was (-1.51,-0.81).The negative scores states that the runners got worse in the year 2018 compared to 2017. The confidence interval did not include a zero which suggests that we reject the null hypothesis. There was a difference in the average improvement among the population of all male repeat 5K runners from year 2017 to year 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).

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

# FILL IN REMAINING CODE

# actual difference between the medians

(actualdiff <- by(nhcombined$improvement, nhcombined$Gender, median))

## nhcombined$Gender: F
## [1] -1.1
## ------------------------------------------------------------
## nhcombined$Gender: M
## [1] -0.8166667

actualdiffimpvt <- median(nhcombined$improvement[nhcombined$Gender == "F"] -
median(nhcombined$improvement[nhcombined$Gender == "M"]))

N <- 10000

diffvals <- rep(NA, N)
for (i in 1:N) {
  fakeGender <- sample(nhcombined$Gender)
  diffvals[i] <- median(nhcombined$improvement[fakeGender == "F"]) -
median(nhcombined$improvement[fakeGender == "M"])
}


#Make histogram of permuted MEDIAN differences
hist(diffvals, col = "green", main = "Permuted Sample Median Diff in
Improvement(mins)", xlab = "Improvement Times (Mins)", breaks = 50)
abline(v = actualdiffimpvt, col="blue", lwd=3)
text(actualdiffimpvt - 0.03, 650, paste("Actual Diff in Medians =",
round(actualdiffimpvt,2)), srt = 90)
```



```
#Two-sided p-value for difference of medians
mean(abs(diffvals) >= abs(actualdiffimpvt))

## [1] 0.0822
```

Null Hypothesis : There is no actual difference in the median time improvement between males and females i.e Median (F) = Median (M) Alternative Hypothesis : There is a difference in the median time improvement between males and females i.e Median (F) is not equal Median(M). In the example since the p value is 0.08 which is greater than 0.05 ; we fail to reject the null hypothesis. Stating that there is no statistically significant difference in the median time improvement between males and females. The histogram seems to be normally distributed and the center is around zero (no difference).