Homework 05 Functions and Permutation Tests

Due by 11:59pm, Friday, February 21, 2025, 11:59pm

S&DS 230/530/ENV 757

Questions 1 and 2 use data from both 2017 and 2018 New Haven Road Races - in particular, we look at 5k run times. You can get data for 2018 HERE and for 2017 HERE.

1) Function for Data Cleaning *(20 points)*

(1.1) *(5 pts)* Load in both .csv files into objects called `nh2017` and `nh2018`. 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.

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

The two datasets do have the same variable names and format.

(1.2) *(15 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` to replace each of these datasets with the cleaned up version of themselves.

```
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[data$Div == "",]$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$Pace)
  data <- data[data$Name != "", ]

  return(data)
}

#run cleanNHData on nh2018 and nh2017 and replace these with the cleaned up
# versions of themselves
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, 10)
```

```
##     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
## 7  3963      Ryan Pearl       Hamden M20-29 16:12 5:13   16:11      M  20-29
## 8  4307  Chris Chisholm   Farmington M55-59 16:17 5:15   16:16      M  55-59
## 9  5131  Dillon Selfors Old Saybrook M13-19 16:18 5:15   16:17      M  13-19
## 10 5740 Tim Milenkevich      Ansonia M30-39 16:18 5:15   16:18      M  30-39
##    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
## 7     16.18333 16.20000 5.216667
## 8     16.26667 16.28333 5.250000
## 9     16.28333 16.30000 5.250000
## 10    16.30000 16.30000 5.250000
```

```
head(nh2018, 10)
```

```
##     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
## 7  4800 Matthew Chaston             M50-54 16:33 5:20   16:32      M  50-54
## 8  3710 Brendan Mellitt    Cheshire M13-19 16:35 5:21   16:33      M  13-19
## 9  4618     Mark Hixson    Simsbury M50-54 16:35 5:21   16:35      M  50-54
## 10 3142    Trey Chometa     Bethany M13-19 16:38 5:22   16:38      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
## 7     16.53333 16.55000 5.333333
## 8     16.55000 16.58333 5.350000
## 9     16.58333 16.58333 5.350000
## 10    16.63333 16.63333 5.366667
```

2) Repeat Runners Dataset *(35 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
```

Based on the difference in the number of rows between the original dataframes and the new ones, 7 observations from 2017 and 45 observations from 2018 were eliminated.

(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
```

986 people ran in both years.

(2.3) *(15 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.

```
# creates a boolean vector that is "TRUE" if the observation name is in
# repeatrunners and "FALSE" otherwise
w <- nh2018Unq$Name %in% repeatrunners

# creates a new dataframe from the 2018 dataset containing only the names,
# genders, and net times of repeat runners (entries where Name is in w).
nhcombined <- data.frame(Name = nh2018Unq$Name[w],
                         Gender = nh2018Unq$Gender[w],
                         Nettime_2018 = nh2018Unq$Nettime_min[w])

# merges this new dataframe with the 2017 dataset. Since the only shared
# variable between the two dataframes is "Name", this merges by Name, which
# associates the 2018 net times with the right 2017 net times. since nhcombined
# only contains 2018 entries from repeat runners, non-repeat runners from 2017
# are dropped
nhcombined <- merge(nhcombined, nh2017Unq[, c("Name", "Nettime_min")])

# drops the rows with unspecified gender
nhcombined <- nhcombined[!is.na(nhcombined$Gender),]

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

# prints the dimensions of the new dataframe
dim(nhcombined)
```

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

```r
# prints the first few rows of the new dataframe
head(nhcombined)
```

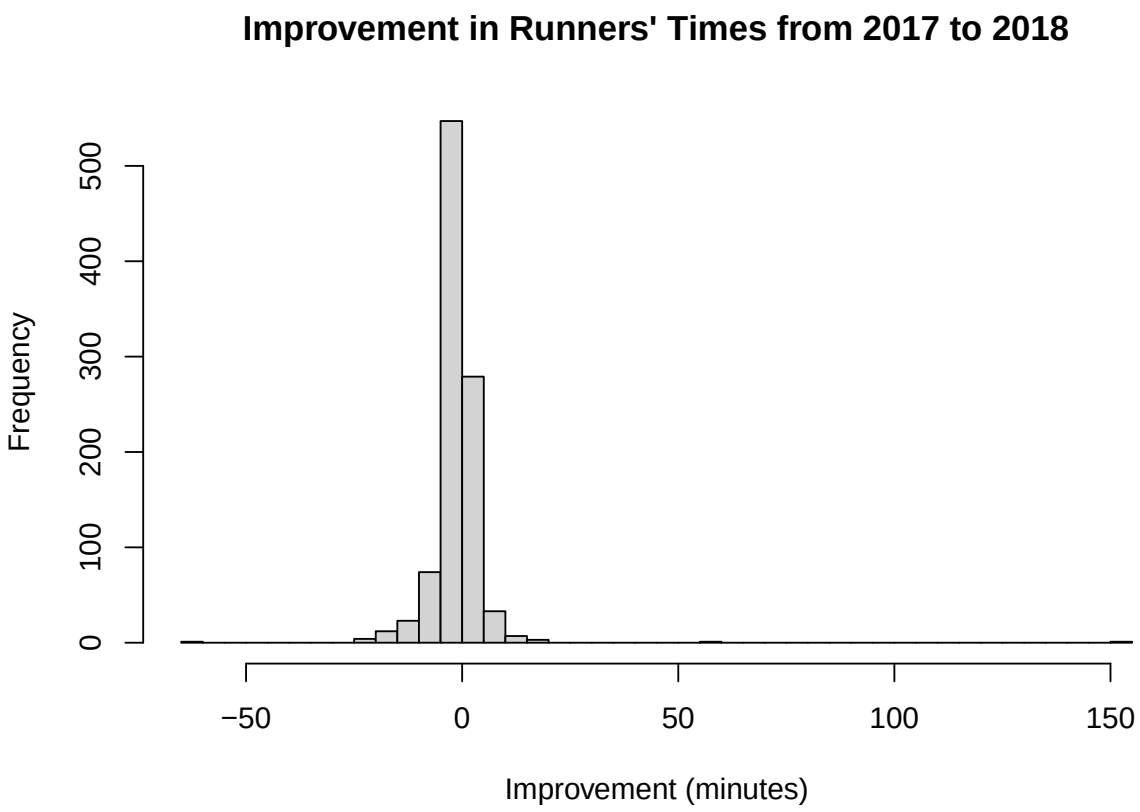
```
##              Name Gender Nettime_2018 Nettime_2017
## 1      Abbey Shaw      F     39.25000     40.25000
## 2     Abby Dziura      F     39.03333     35.63333
## 3      Abby Ganun      F     40.08333     44.65000
## 4     Abi Hawkins      F     35.86667     27.56667
## 5  Abigail Murphy      F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

(2.4) *(5 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_2017 - nhcombined$Nettime_2018
summary(nhcombined$improvement)
```

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

```r
hist(nhcombined$improvement,
     breaks=50,
     xlab="Improvement (minutes)",
     main="Improvement in Runners' Times from 2017 to 2018")
```



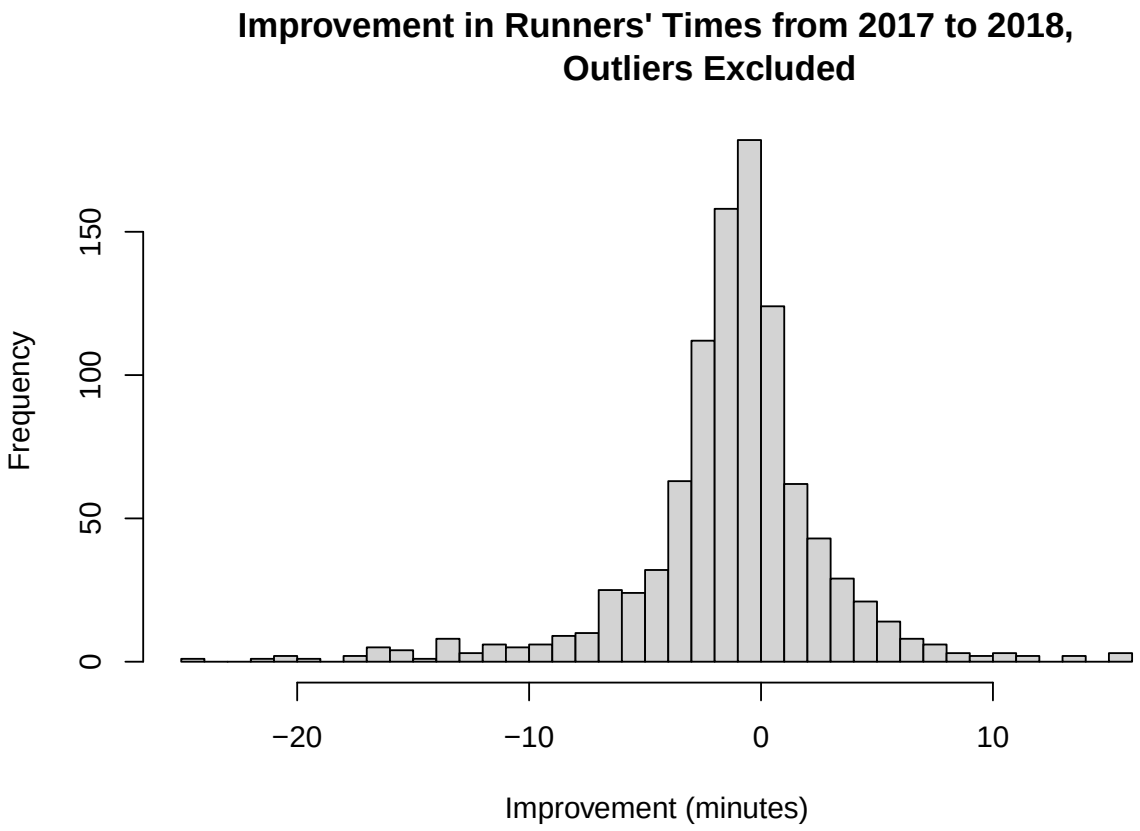
Both the median and mean are negative, which means that in general, the runners got worse from 2017 to 2018. It's hard to see the specifics in the histogram though since the outliers cause the bins to be rather large.

(2.5) *(5 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.

```
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$improvement) <= 50,]
hist(nhcombined$improvement,
     breaks=50,
     xlab="Improvement (minutes)",
     main="Improvement in Runners' Times from 2017 to 2018,
           Outliers Excluded")
```



3) Changes in Measles Vaccination Rates in the Past 8 Years *(45 pts)*

Question 3 uses data from from the 2016 and 2024 World Bank datasets. You can get data for 2016 HERE and for 2024 HERE.

(3.1) *(10 pts)* Read in the datasets. Get and show the names on each dataset. Confirm that the dimensions are the same between datasets. Then, modify each dataset so that it only contains "Country", "Measles", and "GNI".

Following the example in question 2, combine these datasets together. You'll need to rename "Measles" and "GNI" in each dataset before combining based on Country (something like "Measles_24", "Measles_16", etc.) In the combined dataset, remove any observations that are missing for either Measles variable or for GNI in 2024.

Create a new factor variable that identifies countries as having GNI in 2024 greater than 8000 or less than or equal to 8000.

Finally, calculate a variable that is the change in Measles vaccination rates (2024 minus 2016) per country.

Show the first 10 rows of the final dataset.

```
data2016 <- read.csv("http://reuningscherer.net/S&DS230/data/WB.2016.csv")
data2024 <- read.csv("http://reuningscherer.net/S&DS230/data/WB_2024.csv")
dim(data2016)
```

```
## [1] 217  29
```

```r
dim(data2024)
```

```
## [1] 217  17
```

```r
names(data2016)
```

```
##  [1] "Country"       "Code"          "Population"    "Rural"
##  [5] "GNI"           "IncomeTop10"   "Imports"       "Exports"
##  [9] "Military"      "Cell"          "Fertility66"   "Fertility16"
## [13] "Measles"       "InfMort"       "LifeExp"       "PM2.5"
## [17] "Diesel"        "CO2"           "EnergyUse"     "FossilPct"
## [21] "Forest94"      "Forest14"      "Deforestation" "GunTotal"
## [25] "GunHomicide"   "GunSuicide"    "GunUnint"      "GunUndet"
## [29] "GunsPer100"
```

```r
names(data2024)
```

```
##  [1] "Country"    "Population" "Rural"      "GNI"        "Imports"
##  [6] "Exports"    "Military"   "Cell"       "Fertility"  "Measles"
## [11] "InfMort"    "LifeExp"    "PM2.5"      "CO2"        "EnergyUse"
## [16] "Renewable"  "Debt"
```

```r
data2016 <- data2016[,c("Country", "Measles", "GNI")]
data2024 <- data2024[,c("Country", "Measles", "GNI")]

w <- data2016$Country %in% intersect(data2016$Country, data2024$Country)
datacombined <- data.frame(Country = data2016$Country[w],
                           Measles_2016 = data2016$Measles[w],
                           GNI_2016 = data2016$GNI[w])
datacombined <- merge(datacombined, data2024)
colnames(datacombined)[4] = "Measles_2024"
colnames(datacombined)[5] = "GNI_2024"
datacombined <- datacombined[!(is.na(datacombined$Measles_2016)
                               | is.na(datacombined$Measles_2024)
                               | is.na(datacombined$GNI_2024)),]
datacombined$GNI_over_8000 <- factor(ifelse(datacombined$GNI_2024 > 8000,
                                            "yes",
                                            "no"))
datacombined$Measles_Diff <- (datacombined$Measles_2024
                              - datacombined$Measles_2016)
head(datacombined, 10)
```

```
##                Country Measles_2016 GNI_2016 Measles_2024 GNI_2024
## 1          Afghanistan           62      580           68      360
## 2              Albania           96     4320           86     6770
## 3              Algeria           94     4360           79     4490
## 5              Andorra           97       NA           98    50080
## 6               Angola           49     3450           37     1870
## 7  Antigua and Barbuda           98    13560           99    18710
## 8            Argentina           90    11940           83    11590
## 9              Armenia           97     3770           95     5960
```

```
## 11           Australia           95    54130           96    60820
## 12             Austria           95    45850           95    55720
##    GNI_over_8000 Measles_Diff
## 1             no            6
## 2             no          -10
## 3             no          -15
## 5            yes            1
## 6             no          -12
## 7            yes            1
## 8            yes           -7
## 9             no           -2
## 11           yes            1
## 12           yes            0
```

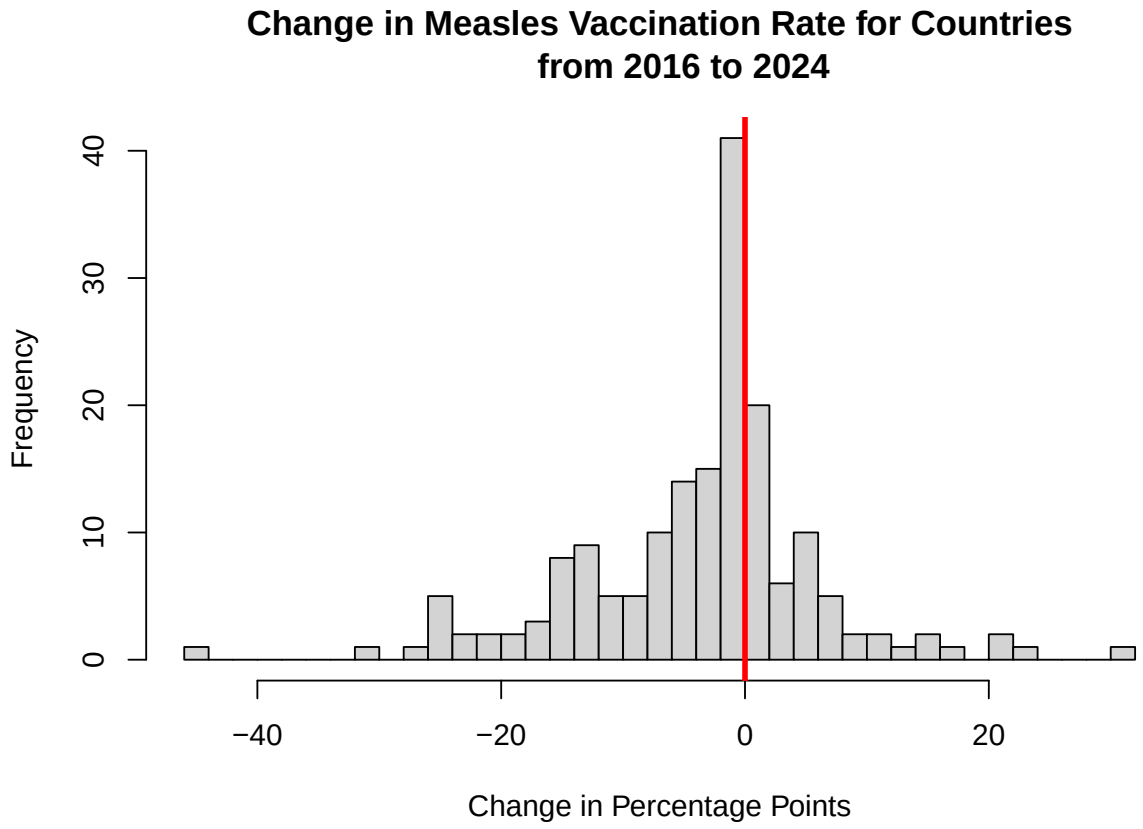
(3.2) *(10 pts)* Calculate and display summary statistics for the change in Measles vaccination rates overall. Make a histogram of these changes and add a vertical line at the value which indicates no change. Discuss what you observe in a few sentences.

Make a side-by-side boxplot to see differences between the change in Measles vaccination rates between Countries with GNI < 8000 and countries with GNI > 8000. Does there appear to be any difference between groups? Comment both on center and spread.

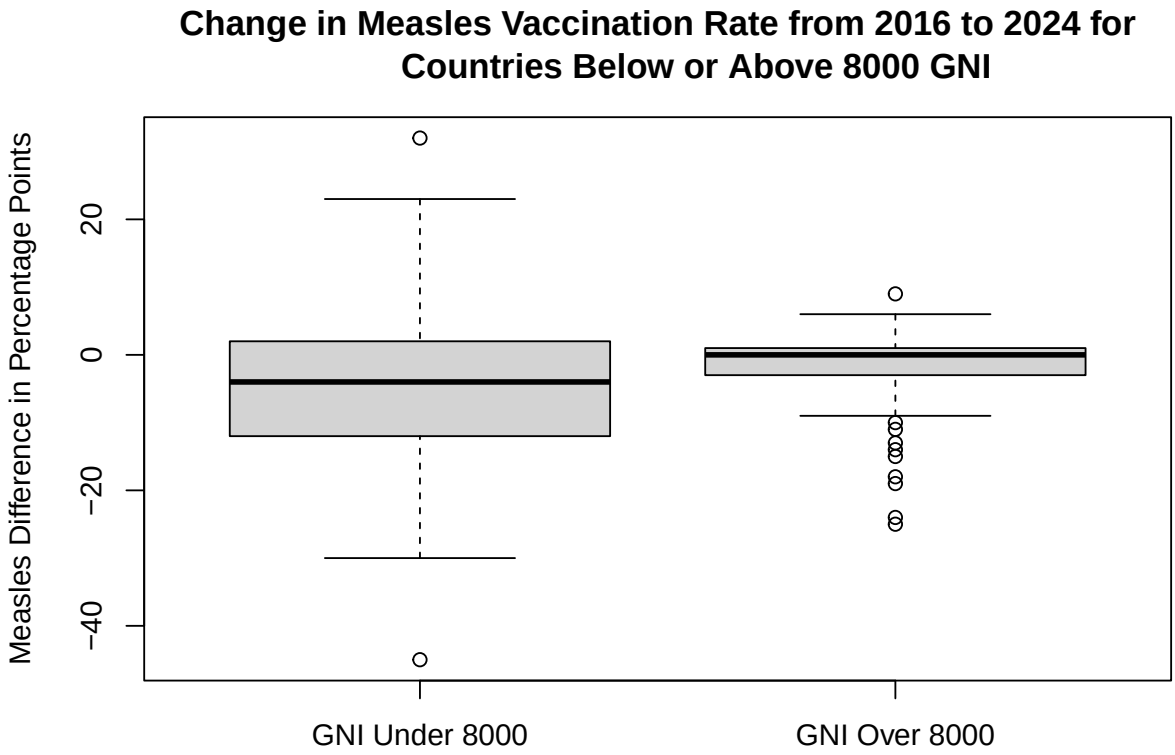
```
summary(datacombined$Measles_Diff)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -45.000  -7.000  -1.000  -3.153   1.000  32.000
```

```
hist(datacombined$Measles_Diff,
     breaks = 30,
     xlab = "Change in Percentage Points",
     main = "Change in Measles Vaccination Rate for Countries
     from 2016 to 2024")
abline(v=0, lwd = 3, col = "red")
```




```
boxplot(datacombined$Measles_Diff ~ datacombined$GNI_over_8000,
        xlab="",
        ylab="Measles Difference in Percentage Points",
        names = c("GNI Under 8000", "GNI Over 8000"),
        main = "Change in Measles Vaccination Rate from 2016 to 2024 for
        Countries Below or Above 8000 GNI")
```



There does seem to be a difference between the two groups of countries. The median difference is slightly higher for the countries with GNI over 8000. The spread is much greater for the countries with GNI under 8000 compared to the countries with GNI over 8000.

(3.3) *(10 pts)* Create a 95% bootstrap confidence interval for the mean change in measles vaccination rates among countries with GNI < 8000. Do the same for countries with a GNI > 8000. 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 separate intervals for each GNI group. Display the intervals and discuss what you observe.

```
# To make grading easier, please leave the following line of code in your assignment
set.seed(230)
n_bootstrap <- 10000
measles_bootstrap1 <- c()
measles_bootstrap2 <- c()
for (i in 1:n_bootstrap) {
  measles_bootstrap1[i] <- mean(
    sample(
      datacombined$Measles_Diff[datacombined$GNI_over_8000 == "no"],
      size=length(datacombined$Measles_Diff[datacombined$GNI_over_8000 == "no"]),
      replace=TRUE))
  measles_bootstrap2[i] <- mean(
    sample(
      datacombined$Measles_Diff[datacombined$GNI_over_8000 == "yes"],
      size=length(datacombined$Measles_Diff[datacombined$GNI_over_8000 == "yes"]),
      replace=TRUE))
```

```
}
quantile(measles_bootstrap1, c(0.025, 0.975))
```

```
##      2.5%     97.5%
## -6.237871 -1.653218
```

```
quantile(measles_bootstrap2, c(0.025, 0.975))
```

```
##      2.5%     97.5%
## -3.592105 -0.750000
```

The 95% confidence interval is lower for the countries with GNI under 8000, though the confidence intervals are large enough that there is overlap between the two confidence intervals.

(3.4) *(15 pts)* Using a permutation test, examine whether there a significant difference in the **MEDIAN** change in vaccination rates between high GNI countries and low GNI countries (calculate as high - low). Use a significance level of 0.01. Be sure to state (in words is fine) the null and alternative hypotheses, and discuss 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).

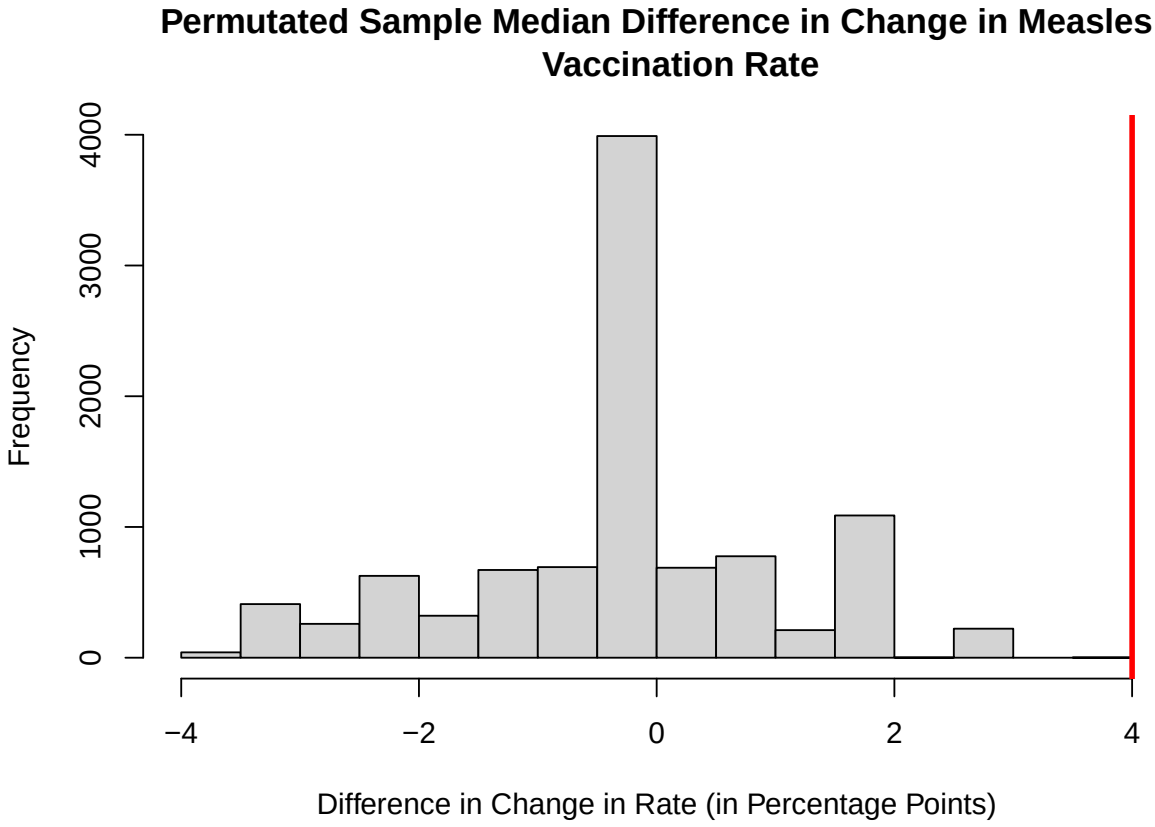
```
# To make grading easier, please leave the following line of code in your assignment
set.seed(230)
true_diff <- (median(datacombined$Measles_Diff
                     [datacombined$GNI_over_8000 == "yes"])
          - median(datacombined$Measles_Diff
                   [datacombined$GNI_over_8000 == "no"]))
true_diff
```

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

```
n_perm <- 10000
perm_diffs <- c()
for (i in 1:n_perm) {
  fakedata <- sample(datacombined$GNI_over_8000)
  perm_diffs[i] <- (median(datacombined$Measles_Diff[fakedata == "yes"])
                    - median(datacombined$Measles_Diff[fakedata == "no"]))
}
mean(abs(perm_diffs) >= abs(true_diff))
```

```
## [1] 0.0016
```

```
hist(perm_diffs,
     breaks=15,
     xlab="Difference in Change in Rate (in Percentage Points)",
     main="Permutated Sample Median Difference in Change in Measles
     Vaccination Rate")
abline(v = true_diff,
       lwd=3,
       col="red")
```



The null hypothesis is that the difference in the median change in vaccination rate of Measles between high GNI countries and low GNI countries is 0. The alternative hypothesis is that this difference of medians is not 0. From the data we found that the difference of medians was 4, which has a significance level of 0.0016. Thus, we reject the null hypothesis as the data supports the claim that the difference in medians of change in vaccination rate between high GNI countries and low GNI countries is not 0.