Homework Example

Gretchen Martinet

Problem 1

a) Save the parameter values as objects.

```
Population_Mean<- c(29.3)
Standard_Deviation <-c(9.9)
Significant_Level<- c(0.05)
```

Part b

```
mytest<-function(x){
  samp<-rnorm(x,29.3,9.9)
  ts<-(mean(samp)-29.3)/(9.9/sqrt(x))
  pval<- 2*pnorm(-abs(ts))
  pval<= Significant_Level
}
mytest(26)
```

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

Part c. What proportion of tests reject the null hypothesis?

```
Replicate<-replicate(10000, mytest(26))
sum(Replicate)/10000
```

```
## [1] 0.047
```

##d. Theoretically, what should the proportion resulting from part (c) be?

```
# It should be equal to the significant level
```

##e. Write a function that will output the #proportion described in part (c) for a given sample size #(again, without using any loops). Execute your function #for the sample sizes 7, 26,and 50.

```r
prop<-function(x){
  Replicate1<-replicate(10000, mytest(x))
  sum(Replicate1)/10000
}
prop(7)
```

```
## [1] 0.0497
```

```r
prop(26)
```

```
## [1] 0.0498
```

```r
prop(50)
```

```
## [1] 0.0517
```

(f) Without using any loops,

#execute the function written in part (e) #for every sample size from 3 through 55.

```r
sapply(3:55,prop)
```

```
##  [1] 0.0485 0.0485 0.0474 0.0452 0.0468 0.0525 0.0464 0.0479 0.0512 0.0517
## [11] 0.0484 0.0495 0.0486 0.0519 0.0544 0.0506 0.0486 0.0467 0.0532 0.0511
## [21] 0.0516 0.0503 0.0498 0.0540 0.0505 0.0506 0.0475 0.0504 0.0459 0.0505
## [31] 0.0515 0.0522 0.0502 0.0482 0.0511 0.0482 0.0524 0.0496 0.0538 0.0499
## [41] 0.0502 0.0481 0.0443 0.0497 0.0497 0.0506 0.0501 0.0506 0.0464 0.0524
## [51] 0.0481 0.0489 0.0501
```

#(g) What do you notice about the general trend of the resulting proportions? Does sample size appear to have any effect on the results?

```r
# The general trend that it is
#close to the significant level which is 0.05.
#Even the sample size is different,
#the prop is almost 0.05
```

Problem 2 #a. Read in the file and save the data

#in a data frame called nym2019.

```
nym2019 <-
read.table("/Users/mailuu/Desktop/STAT3080/nym2019.txt", header=TRUE,
stringsAsFactors =FALSE,
fileEncoding = "latin1")
head(nym2019)
```

```
##   Sex Age Place DivPlace    DIV   Time BostonQualifier HomeStateOrCountry
## 1   M  26  7660      751 M25-29 216.40               N                 FL
## 2   M  29   768      157 M25-29 172.02               Y                 NY
## 3   M  52  6028      494 M50-54 209.52               N                FRA
## 4   M  42  9247     1252 M40-44 222.30               N                GER
## 5   M  40  5819      861 M40-44 208.77               N                FRA
## 6   M  50   859       31 M50-54 173.45               Y                 IL
```

2b. Determine the number of finishers' times that are contained in this data set.

```
length(nym2019$Time)
```

```
## [1] 250
```

2c. Determine the number of finishers in the data whose home country is the U.S.

```
 sum(nchar(as.character(nym2019$HomeStateOrCountry))==2)
```

```
## [1] 121
```

2d. Determine the number of finishers representing each country.

```
home<-as.character((nym2019$HomeStateOrCountry))
home[nchar(home)==2]<-"USA"
table(home)
```

```
## home
## ARG ARU AUS BEL BRA CAN CHI CHN CRC CZE ESP ETH FIN FRA GBR GER INA IRL ISR ITA
##   1   1   6   1   4   4   3   7   2   1  10   1   1  21   6  10   1   2   1   8
## JPN KEN MEX NED NOR NZL PAN PAR POL POR RSA RUS SIN SLO SUI SWE TPE UKR URU USA
##   5   1   4   8   2   1   1   1   1   1   1   2   1   1   3   1   1   1   1 121
## VEN
##   1
```

2e.Determine the number of countries represented in the data.

```
nrow(table(home))
```

```
## [1] 41
```

f. Determine the age of the youngest and

#oldest finishers given in the data

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------- tidyverse 1.3.0 --

## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
Youngest_Oldest<-group_by(nym2019) %>%
  summarize(YoungestAge=min(Age),OldestAge=max(Age))
Youngest_Oldest
```

```
## # A tibble: 1 x 2
##   YoungestAge OldestAge
##         <int>     <int>
## 1          23        66
```

(g) Determine the age of the fastest and slowest finishers given in the data.

```
nym2019[nym2019$Time == min(nym2019$Time),"Age"]
```

```
## [1] 25
```

```
nym2019[nym2019$Time == max(nym2019$Time),"Age"]
```

```
## [1] 37
```

h. #9. number who qualifies. and their information

```
Top10_Div<-nym2019[nym2019$DivPlace<= 10,]
nrow(Top10_Div)
```

```
## [1] 9
```

```
Top10_Div
```

```
##     Sex Age Place DivPlace    DIV   Time BostonQualifier HomeStateOrCountry
## 36    F  35   919       10 F35-39 174.15               Y                 NY
## 37    M  36    95        5 M35-39 153.43               Y                 NJ
## 66    F  29   748        6 F25-29 171.68               Y                 NC
## 71    F  30    50        2 F30-34 147.12               Y                AUS
## 86    F  24   265        1 F20-24 162.35               Y                ETH
## 152   F  43    92        4 F40-44 153.07               Y                 FL
## 214   F  42    43        2 F40-44 146.38               Y                AUS
## 225   M  66  5854        6 M65-69 208.88               Y                 OH
## 236   M  25     2        2 M25-29 128.60               Y                KEN
```

#i. Determine the divisions of the finishers who finished in the Top 10 of their division.

```
Divisions_Top_Ten<-Top10_Div[,5]
sort(unique(Divisions_Top_Ten))
```

```
## [1] "F20-24" "F25-29" "F30-34" "F35-39" "F40-44" "M25-29" "M35-39" "M65-69"
```

(j) Display all information for finishers who finished in the Top 5 of their division.

```
Top_5_Div<-nym2019[nym2019$DivPlace<= 5,]
Top_5_Div
```

```
##     Sex Age Place DivPlace    DIV   Time BostonQualifier HomeStateOrCountry
## 37    M  36    95        5 M35-39 153.43               Y                 NJ
## 71    F  30    50        2 F30-34 147.12               Y                AUS
## 86    F  24   265        1 F20-24 162.35               Y                ETH
## 152   F  43    92        4 F40-44 153.07               Y                 FL
## 214   F  42    43        2 F40-44 146.38               Y                AUS
## 236   M  25     2        2 M25-29 128.60               Y                KEN
```

##(k) Determine the average age of finishers who did and who did not qualify for the Boston Marathon.

```
tapply(nym2019$Age, nym2019$BostonQualifier, mean)
```

```
##        N        Y
## 39.86364 36.72034
```

References

1. https://www.mathsisfun.com/numbers/addition.html
2. https://www.mathsisfun.com/numbers/subtraction.html