Homework 3

Noah McIntire

Problem 1

a)

```
mean1<-99.5
std1<-4.8
sig1<-.05
```

b)

```
zfunc <- function(x){
samp <- rnorm(x, mean=mean1, sd=std1)
smean <- mean(samp)
zTest <- (smean - mean1)/(std1/sqrt(x))
if (abs(zTest)>1.96) {
  return(TRUE)
} else {
  return(FALSE)
}
}
zfunc(27)
```

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

c)

```
repvect <- replicate(10000,zfunc(27))
length(repvect[repvect == T])/10000
```

```
## [1] 0.0458
```

d)

Theoretically, the proportion resulting from part c should be 1 as the sample is defined as a perfect normal distribution with a mean of 99.5. This means that a z-test of a random sample of values from the normal distribution should always find 99.5 to be the true mean of the population, however with a smaller sample size it can be hard to prove with a 0.05 significance level.

e)

```
prop <- function(x){
vectprop<- replicate(10000,zfunc(x))
length(vectprop[vectprop == T])/10000
}
prop(9)
```

```
## [1] 0.0515
```

```
prop(27)
```

```
## [1] 0.0476
```

```
prop(51)
```

```
## [1] 0.049
```

f)

```
sapply(3:51, prop)
```

```
##  [1] 0.0512 0.0470 0.0489 0.0471 0.0507 0.0467 0.0508 0.0517 0.0479 0.0510
## [11] 0.0474 0.0504 0.0494 0.0520 0.0510 0.0499 0.0484 0.0485 0.0502 0.0510
## [21] 0.0476 0.0518 0.0508 0.0468 0.0486 0.0510 0.0501 0.0503 0.0529 0.0510
## [31] 0.0499 0.0491 0.0506 0.0476 0.0473 0.0496 0.0470 0.0473 0.0503 0.0500
## [41] 0.0513 0.0499 0.0511 0.0445 0.0538 0.0512 0.0496 0.0515 0.0494
```

g)

Sample size does not appear to have an effect on the results of each sample size, though it should as the sample size is used within the zscore equation.

Problem 2

a)

```
setwd("/Users/noahmcintire/Desktop/STAT 3080")
nym2019<-read.table("nym2019.txt", header=TRUE)
head(nym2019)
```

```
##   Sex Age Place DivPlace    DIV DivAge   Time BostonQualifier
## 1   M  38  5824      947 M35-39  35-39 208.80               N
## 2   M  44 18719     2314 M40-44  40-44 248.27               N
## 3   M  56 14716      609 M55-59  55-59 237.72               Y
## 4   M  48 11240     1327 M45-49  45-49 228.72               N
## 5   M  44  1572      248 M40-44  40-44 180.15               N
## 6   M  28   245       64 M25-29  25-29 161.42               Y
##   HomeStateOrCountry
## 1                 NY
## 2                 NC
## 3                 NY
## 4                 NJ
## 5                ESP
## 6                 NY
```

b)

```
vect2 <- nym2019$Time[na.rm = T]
length(vect2)
```

```
## [1] 400
```

c)

```
vect3 <- nym2019$HomeStateOrCountry[nchar(nym2019$HomeStateOrCountry) == 2]
length(vect3)
```

```
## [1] 191
```

Since all us states and territories only have two character length acronyms, this allowed me to subset the data by using nchar.

d)

```
usrun <- length(vect3)
vect4<-nym2019$HomeStateOrCountry[nchar(nym2019$HomeStateOrCountry) == 3]
vect5<-replicate(usrun, "US")
vect4<-c(vect4, vect5)
table(vect4)
```

```
## vect4
## AND ARG AUS AUT BEL BRA CAN CHN COL CZE DEN ECU ESA ESP ETH FRA GBR GER GUA HKG
##   1   1  10   2   2   4  15   6   3   1   4   2   1  13   6  25  20  10   1   2
## HUN INA IRL ITA JPN KEN MEX NCA NED NOR NZL PER PHI POL POR RSA RUS SIN SRI SUI
##   1   1   5  17   4   2   6   1   9   3   1   2   1   4   2   1   1   1   1   5
## SWE THA TPE UGA UKR  US VEN
##   6   1   1   1   1 191   2
```

e)

```
length(unique(vect4))
```

```
## [1] 47
```

f)

```
age <- nym2019$Age
quantile(age)
```

```
##    0%   25%   50%   75%  100%
## 21.00 31.75 38.00 46.00 71.00
```

The Youngest finisher was 21 and the oldest finisher was 71.

g)

```
speed <- nym2019$Time
print(quantile(speed))
```

```
##      0%     25%     50%     75%    100%
## 130.650 172.565 209.045 233.205 251.280
```

```
frow<-nym2019[nym2019$Time== 130.650,]
srow<-nym2019[nym2019$Time==251.280,]
vect6<-c(frow$Age, srow$Age)
vect6
```

```
## [1] 23 41
```

The fastest finisher finished was 23 years old the the slowest finisher was 41 years old.

h)

```
place <- nym2019$DivPlace[nym2019$DivPlace <= 20]
length(place)
```

```
## [1] 31
```

i)

```
top20<-nym2019[nym2019$DivPlace <= 20,]
div<-sort(unique(top20$DIV))
div
```

```
##  [1] "F20-24" "F25-29" "F30-34" "F35-39" "F40-44" "M20-24" "M25-29" "M30-34"
##  [9] "M35-39" "M40-44" "M45-49" "M50-54" "M70-74"
```

j)

```
top5 <- nym2019[nym2019$DivPlace <= 5,]
top5
```

```
##     Sex Age Place DivPlace    DIV DivAge   Time BostonQualifier
## 13    M  70  6929        4 M70-74  70-74 213.37                Y
## 56    M  71  9278        5 M70-74  70-74 222.43                N
## 63    M  40    25        2 M40-44  40-44 139.68                N
## 126   M  38    11        1 M35-39  35-39 132.95                Y
## 137   F  41    74        3 F40-44  40-44 150.20                N
## 159   M  23     5        1 M20-24  20-24 130.65                Y
## 172   M  46    91        3 M45-49  45-49 153.05                N
## 281   F  24   265        1 F20-24  20-24 162.35                Y
## 389   F  25    39        2 F25-29  25-29 145.85                Y
##     HomeStateOrCountry
## 13                 CHN
## 56                  MI
## 63                 SWE
## 126                GER
## 137                 NJ
## 159                ETH
## 172                 NY
## 281                ETH
## 389                ETH
```

k)

```
notq <- nym2019[nym2019$BostonQualifier == "N",]
yesq <- nym2019[nym2019$BostonQualifier == "Y",]
mean(notq$Age)
```

```
## [1] 39.25234
```

```
mean(yesq$Age)
```

```
## [1] 38.95699
```

References

1.http://uc-r.github.io/na_exclude

2.https://rdrr.io/r/base/nchar.html

3.https://www.geeksforgeeks.org/sorting-of-arrays-in-r-programming/