Homework #2

Carson Crenshaw (cgc8gdt)

```
library(car)
```

```
## Loading required package: carData
```

Problem 1

Part a

```
commutes <- matrix(c(10,11,13,10,9,16,13,11,9,13), nrow=5, ncol=2, byrow=FALSE)
commutes
```

```
##      [,1] [,2]
## [1,]   10   16
## [2,]   11   13
## [3,]   13   11
## [4,]   10    9
## [5,]    9   13
```

Part b

```
colnames(commutes) <- c("Week1", "Week2")
rownames(commutes) <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday")
commutes
```

```
##           Week1 Week2
## Monday       10    16
## Tuesday      11    13
## Wednesday    13    11
## Thursday     10     9
## Friday        9    13
```

Part c

For Monday, Tuesday, and Friday, Professor V arrived faster leaving at 7:15am than at 7:30am.

```
commutes[,"Week1"] < commutes[,"Week2"]
```

```
##    Monday   Tuesday Wednesday  Thursday    Friday
##      TRUE      TRUE     FALSE     FALSE      TRUE
```

Part d

```
apply(commutes, 1, mean)
```

```
##    Monday   Tuesday Wednesday  Thursday    Friday
##      13.0      12.0      12.0       9.5      11.0
```

Part e

```
diff<-15-commutes
diff
```

```
##           Week1 Week2
## Monday        5    -1
## Tuesday       4     2
## Wednesday     2     4
## Thursday      5     6
## Friday        6     2
```

Part f

The average difference over Week1 was 4.4. The average distance for Week2 was 2.6

```
apply(diff, 2, mean)
```

```
## Week1 Week2
##   4.4   2.6
```

Part g

On the earliest day for both weeks Professor V was early by 6 minutes.

```
apply(diff, 2, max)
```

```
## Week1 Week2
##     6     6
```

Part h

The days of the second week on which Professor V arrived to work within 12 minutes are Wednesday and Thursday.

```
rownames(commutes[which(commutes[,2]<=12),])
```

```
## [1] "Wednesday" "Thursday"
```

Part i

For Week1, Professor V arrived within her budged time 5/5 days. For Week2, Professor V arrived within her budged time 4/5 days.

```
apply(commutes<=15,2,sum)
```

```
## Week1 Week2
##     5     4
```

Part j

Professor V arrived the fastest in the first week on Friday.

```
names(which(commutes[,1] == min(commutes[,1])))
```

```
## [1] "Friday"
```

```
#Using the subset function is also another way to result in the correct answer
#rownames(subset(commutes, commutes[,1] == min(commutes[,1])))
```

Problem 2

Part a

```
weight.metric <- Davis[,c(2,4)]
head(weight.metric)
```

```
##   weight repwt
## 1     77    77
## 2     58    51
## 3     53    54
## 4     68    70
## 5     59    59
## 6     76    76
```

Part b

```
weight.imp <- weight.metric * 2.2
head(weight.imp)
```

```
##   weight repwt
## 1  169.4 169.4
## 2  127.6 112.2
## 3  116.6 118.8
## 4  149.6 154.0
## 5  129.8 129.8
## 6  167.2 167.2
```

Part c

```
height.metric <- Davis[,c(3,5)]
head(height.metric)
```

```
##   height repht
## 1    182   180
## 2    161   159
## 3    161   158
## 4    177   175
## 5    157   155
## 6    170   165
```

Part d

```
height.imp <- height.metric/2.54
head(height.imp)
```

```
##     height    repht
## 1 71.65354 70.86614
## 2 63.38583 62.59843
## 3 63.38583 62.20472
## 4 69.68504 68.89764
## 5 61.81102 61.02362
## 6 66.92913 64.96063
```

Part e

```
Davis.imp <- data.frame(sex=Davis$sex, rec.weight=weight.imp$weight,
                        rep.weight=weight.imp$repwt, rec.height=height.imp$height,
                        rep.height=height.imp$repht)
head(Davis.imp)
```

```
##   sex rec.weight rep.weight rec.height rep.height
## 1   M      169.4      169.4   71.65354   70.86614
## 2   F      127.6      112.2   63.38583   62.59843
## 3   F      116.6      118.8   63.38583   62.20472
## 4   M      149.6      154.0   69.68504   68.89764
## 5   F      129.8      129.8   61.81102   61.02362
## 6   M      167.2      167.2   66.92913   64.96063
```

Part f

There are 34 total missing values across the Davis.imp data frame.

```
sapply(Davis.imp, function(x) sum(is.na(x)))
```

```
##        sex rec.weight rep.weight rec.height rep.height
##          0          0         17          0         17
```

Part g

There are 19 rows in the Davis.imp data frame that have missing values.

```
sum(apply(is.na(Davis.imp), 1, sum)>0)
```

```
## [1] 19
```

Part h

```
#Subsetting the sex assigned at birth in a data frame format of the subjects
#whose weight was not reported.
Davis.imp[is.na(Davis.imp$rep.weight)==TRUE,]["sex"]
```

```
##     sex
## 47    M
## 48    F
## 55    M
## 76    F
## 100   F
## 125   M
## 127   F
## 138   F
## 154   F
## 158   F
## 159   F
## 172   F
## 174   M
## 177   F
## 182   F
## 183   M
## 198   M
```

```
#A subset vector can also be created using the following code:
#Davis.imp$sex[is.na(Davis.imp$rep.weight)]
```

Problem 3

Part a

```
nym2019 <- read.table("nym2019.txt", header=TRUE)
head(nym2019)
```

```
##   Sex Age Place DivPlace    DIV DivAge   Time BostonQualifier
## 1   M  47   979       69 M45-49  45-49 175.00               Y
## 2   M  26   504      118 M25-29  25-29 167.70               Y
## 3   M  44 18719     2314 M40-44  40-44 248.27               N
## 4   M  45 10766     1269 M45-49  45-49 227.40               N
## 5   M  44  7623     1065 M40-44  40-44 216.27               N
## 6   M  32 15447     1795 M30-34  30-34 239.25               Y
##   HomeStateOrCountry
## 1                SWE
## 2                 NY
## 3                 NC
## 4                AUS
## 5                ESP
## 6                AUS
```

Part b

There are 400 finishers' times that are contained in this dataset.

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

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

Part c

There are 191 finishers in the data whose home country is the U.S., including U.S. territories.

```
#US states/territories have a string length of two letters.
dim(nym2019[nchar(nym2019$HomeStateOrCountry)==2,])
```

```
## [1] 191   9
```

```
#The sum() function would also work here:
#sum(nchar(nym2019$HomeStateOrCountry)==2)
```

Part d

```
#A more complicated method could be undertaken in which a function is built
#so that an sapply() function could be used to count the row numbers of each
#unique country, but the table function alleviates this excessive effort.
#This was discussed during office hours.
countries <- nym2019$HomeStateOrCountry
countries[nchar(countries)==2] <- "USA"
table(countries)
```

```
## countries
## 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 USA VEN
##   6   1   1   1   1 191   2
```

Part e

There are 47 unique countries in the nym2019 dataset.

```
length(unique(countries))
```

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

Part f

The oldest finisher in the data is 71. The youngest is 21.

```
range(nym2019$Age)
```

```
## [1] 21 71
```

```
#The max and min functions confirm the range values.
#max(nym2019$Age)
#min(nym2019$Age)
```

Part g

The age of the fastest finisher is 23. The ages of the slowest finishers are 41 and 46.

```
nym2019$Age[nym2019$Time == max(nym2019$Time)] #Slowest
```

```
## [1] 41 46
```

```
nym2019$Age[nym2019$Time == min(nym2019$Time)] #Fastest
```

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

Part h

31 finishers finished in the Top 20 of their division.

```
sum(nym2019$DivPlace <= 20)
```

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

Part i

```
top20<-nym2019[nym2019$DivPlace <= 20,]
sort(unique(top20$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"
```

Part j

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

```
##     Sex Age Place DivPlace    DIV DivAge   Time BostonQualifier
## 17    M  38    11        1 M35-39  35-39 132.95               Y
## 22    F  25    39        2 F25-29  25-29 145.85               Y
## 56    F  24   265        1 F20-24  20-24 162.35               Y
## 82    M  70  6929        4 M70-74  70-74 213.37               Y
## 160   F  41    74        3 F40-44  40-44 150.20               N
```

```
## 162   M  46    91        3 M45-49  45-49 153.05                N
## 257   M  71  9278        5 M70-74  70-74 222.43                N
## 392   M  23     5        1 M20-24  20-24 130.65                Y
## 400   M  40    25        2 M40-44  40-44 139.68                N
##     HomeStateOrCountry
## 17                 GER
## 22                 ETH
## 56                 ETH
## 82                 CHN
## 160                 NJ
## 162                 NY
## 257                 MI
## 392                ETH
## 400                SWE
```

Part k

The average age of someone who did not qualify for the Boston Marathon is 39.25. The average age of someone who did qualify is 38.96.

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

```
##        N        Y
## 39.25234 38.95699
```

Problem 4

Part a

```
popmean <- 76.4
popsd <- 3.5
popsiglevel <- 0.05
```

Part b

The the null hypothesis is not rejected (FALSE).Therefore, there is not enough evidence to claim that the true population mean is different than 76.4, at the 0.05 significance level.

```
#Null Hypothesis = pop mean equals 76.4
#Alternative Hypothesis = pop mean does not equal 76.4

ztestfunc <- function(size){
  samp<-rnorm(size, mean=popmean, sd=popsd)
  zscore <- ((mean(samp) - popmean) / (popsd/sqrt(size)))
  pvalue <- pnorm(abs(zscore), lower.tail=FALSE)*2
  pvalue <= popsiglevel
}

ztestfunc(26)
```

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

Part c

Approximately 5% of tests reject the null hypothesis.

```
rejectnulltotal <- sum(replicate(10000,ztestfunc(26)))
rejectnulltotal/10000
```

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

Part d

Theoretically, the proportion resulting from part (c) should be 0.05 or 5% because it is equivalent to the significance level.

Part e

```
propc <- function(size){
  ztestfunc(size)
  rejectnullcount <- sum(replicate(10000,ztestfunc(size)))
  finalprop <- rejectnullcount/10000
  c(finalprop)
}

propc(8)
```

```
## [1] 0.0506
```

```
propc(26)
```

```
## [1] 0.0454
```

```
propc(53)
```

```
## [1] 0.0488
```

Part f

```
sapply(3:53,propc)
```

```
##  [1] 0.0492 0.0471 0.0462 0.0502 0.0507 0.0478 0.0485 0.0498 0.0507 0.0487
## [11] 0.0501 0.0483 0.0467 0.0507 0.0473 0.0481 0.0528 0.0510 0.0533 0.0491
## [21] 0.0508 0.0478 0.0482 0.0500 0.0513 0.0519 0.0476 0.0480 0.0509 0.0490
## [31] 0.0493 0.0534 0.0509 0.0510 0.0459 0.0506 0.0488 0.0491 0.0488 0.0531
## [41] 0.0464 0.0485 0.0489 0.0466 0.0497 0.0504 0.0491 0.0508 0.0486 0.0499
## [51] 0.0534
```

Part g

All of the proportions cluster around the theoretical value of 0.05. Sample size has little effect on the results because each test using a given sample size is run 10,000 times.