Bios 6301: Assignment 6

Brooklyn Stanley

Due Thursday, 3 December, 1:00 PM

$5^{n=day}$ points taken off for each day late.

50 points total.

Submit a single knitr file (named `homework6.rmd`), along with a valid PDF output file. Inside the file, clearly indicate which parts of your responses go with which problems (you may use the original homework document as a template). Add your name as `author` to the file's metadata section. Raw R code/output or word processor files are not acceptable.

Failure to name file `homework6.rmd` or include author name may result in 5 points taken off.

Question 1

15 points

Consider the following very simple genetic model (*very* simple – don't worry if you're not a geneticist!). A population consists of equal numbers of two sexes: male and female. At each generation men and women are paired at random, and each pair produces exactly two offspring, one male and one female. We are interested in the distribution of height from one generation to the next. Suppose that the height of both children is just the average of the height of their parents, how will the distribution of height change across generations?

Represent the heights of the current generation as a dataframe with two variables, m and f, for the two sexes. We can use `rnorm` to randomly generate the population at generation 1:

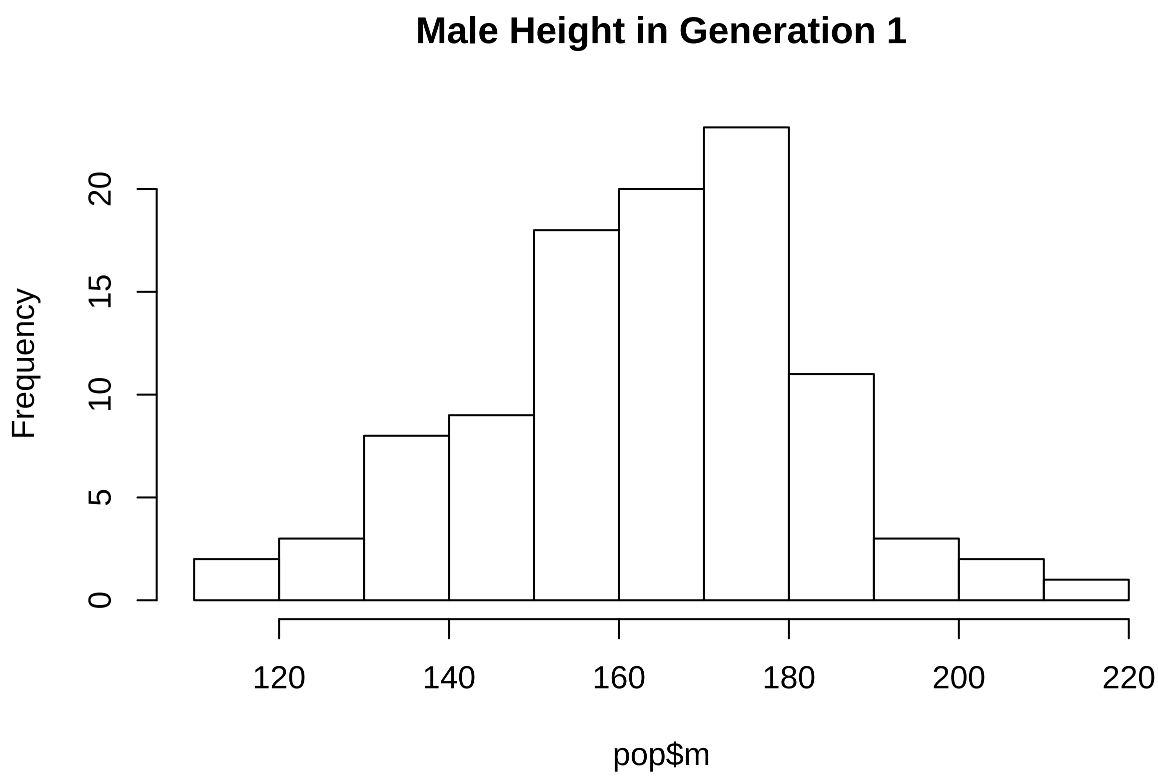
```
pop <- data.frame(m = rnorm(100, 160, 20), f = rnorm(100, 160, 20), gen=1)
```

The following function takes the data frame `pop` and randomly permutes the ordering of the men. Men and women are then paired according to rows, and heights for the next generation are calculated by taking the mean of each row. The function returns a data frame with the same structure, giving the heights of the next generation.

```
next_gen <- function(pop,gen) {
    temp <- pop[((gen-1)*100-99):((gen-1)*100),]
    temp$m <- sample(temp$m)
    temp$m <- rowMeans(temp[-3])
    temp$f <- temp$m
    temp$gen <- gen
    temp
}
```

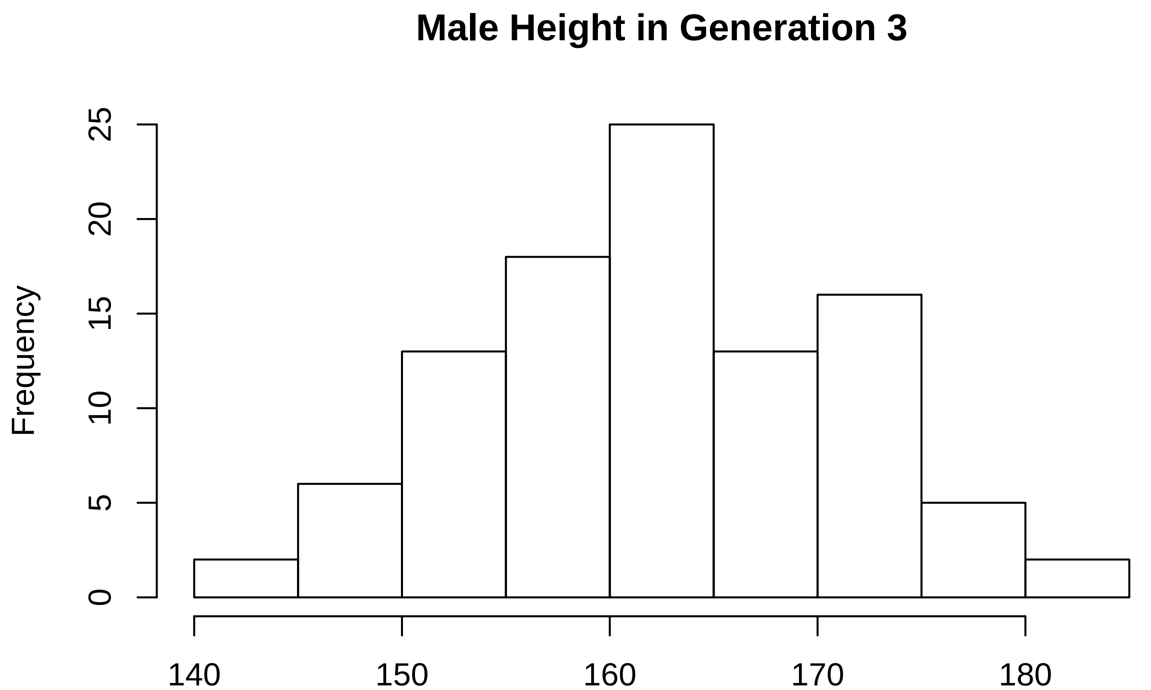
Use the function `next_gen` to generate nine generations (you already have the first), then use the function `hist` to plot the distribution of male heights in each generation (this will require multiple calls to `hist`). The phenomenon you see is called regression to the mean. Provide (at least) minimal decorations such as title and x-axis labels.

```
hist(pop$m, main = "Male Height in Generation 1")
```




```
for (i in 2:9) {
    poptemp <- next_gen(pop, i)
    hist(poptemp$m, main = paste("Male Height in Generation", i), breaks = 9,
        xlab = " ")
    cat("\n", "\n")
    pop <- rbind(pop, poptemp)
}
```

Male Height in Generation 2
Frequency
0
5
10
15
130
140
150
160
170
180
190



Male Height in Generation 3
Frequency
0
5
10
15
20
25
140
150
160
170
180

Male Height in Generation 4
Frequency
0
5
10
15
20
25
30
150
155
160
165
170
175
180

Male Height in Generation 5
Frequency
0
5
10
15
155
160
165
170


Male Height in Generation 6
Frequency
0
5
10
15
20
25
156
158
160
162
164
166
168
170

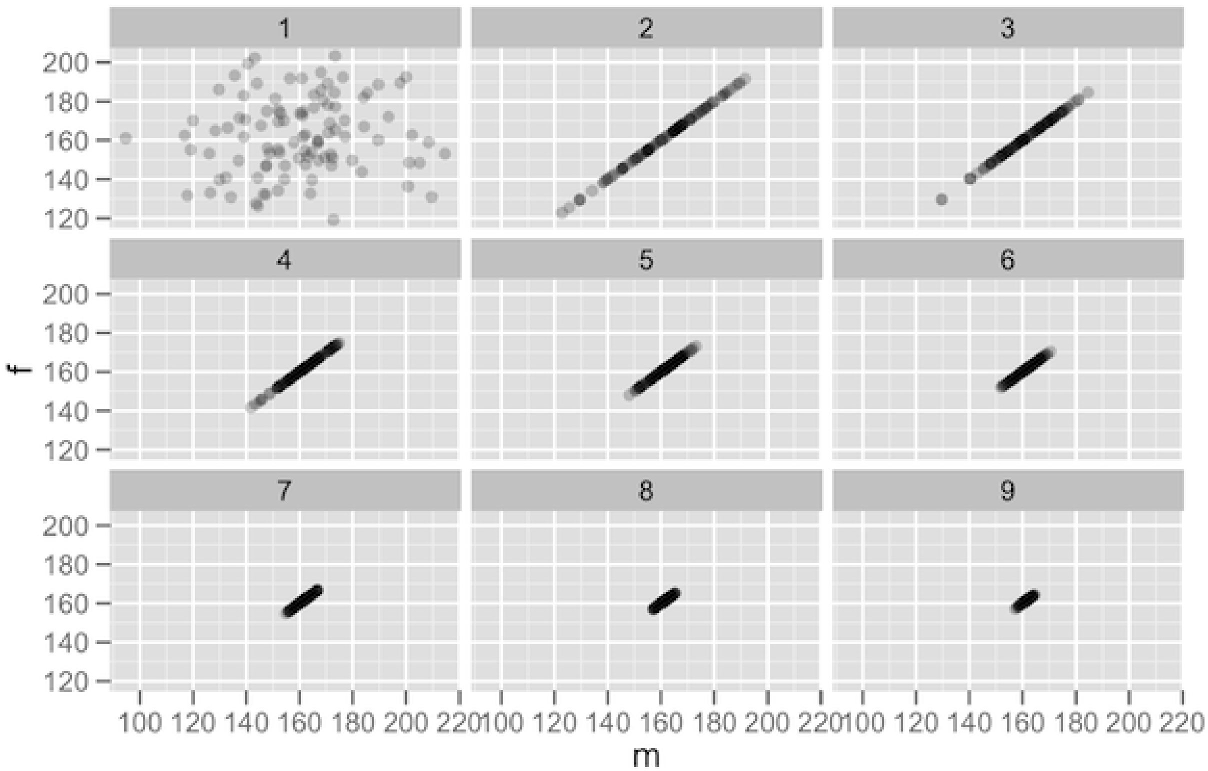

Male Height in Generation 7
Frequency
0
5
10
15
158
160
162
164
166

Question 2

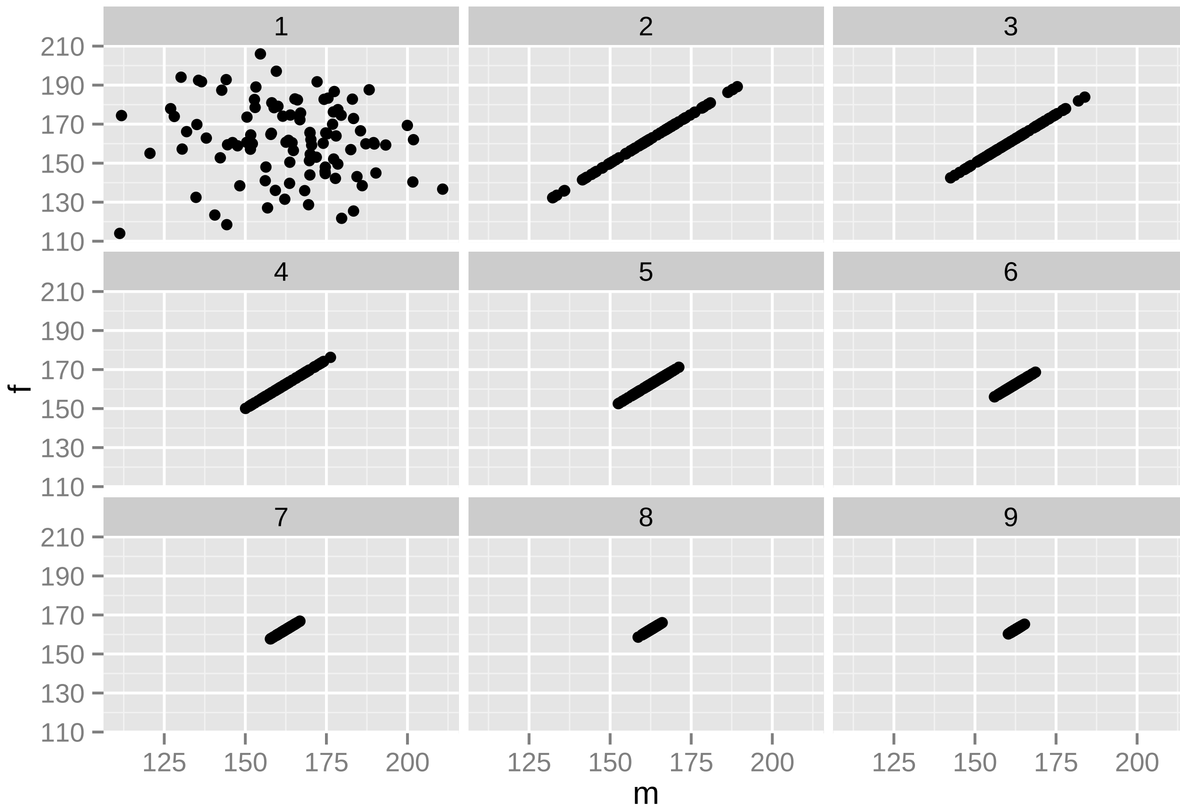
10 points

```r
library(ggplot2)
```

Use the simulated results from question 1 to reproduce (as closely as possible) the following plot in ggplot2.




```r
qplot(data=pop, x=m, y=f, facets = ~gen)
```



Question 3

10 points

We know that the *U(-1,1)* random variable has mean 0. Use a sample of size 100 to estimate the mean and give a 95% confidence interval. Does the confidence interval contain 0? Repeat the above a large number of times (say, 1000) and set the RNG seed to `1000`. What percentage of time does the confidence interval contain 0? Write your code so that it produces output similar to the following (to save space, only output the first ten trials):

```
std <- 2/sqrt(12)
trials <- data.frame(SampleMean = rep(0,1000), LowerBound = rep(0,1000),
                     UpperBound = rep(0,1000), ContainsMean = rep(0,1000))
for (i in 1:1000){
  dat <- runif(100, -1, 1)
  trials$SampleMean[i] <- mean(dat)
  confInt <- trials$SampleMean[i] + qnorm(c(0.025,0.975))*std/10
  trials$LowerBound[i] <- confInt[1]
  trials$UpperBound[i] <- confInt[2]
  if (trials$SampleMean[i] >= confInt[1] & trials$SampleMean[i] <= confInt[2]) {
    trials$ContainsMean[i] <- 1
  }
}

trials[1:10,]
```

```
##      SampleMean  LowerBound UpperBound ContainsMean
```

```
## 1  -0.01977702 -0.13293559 0.09338156            1
## 2   0.06061192 -0.05254665 0.17377050            1
## 3   0.01057505 -0.10258352 0.12373363            1
## 4  -0.02313193 -0.13629050 0.09002665            1
## 5  -0.07954319 -0.19270176 0.03361539            1
## 6   0.07160101 -0.04155757 0.18475958            1
## 7   0.02037453 -0.09278404 0.13353311            1
## 8   0.02014529 -0.09301329 0.13330386            1
## 9   0.06544732 -0.04771125 0.17860590            1
## 10  0.04649401 -0.06666456 0.15965258            1
```

The 95% confidence intervals contain the mean 0% of the time.

Question 4

15 points

Programming with classes. The following function will generate random patient information.

```
makePatient <- function() {
  vowel <- grep("[aeiou]", letters)
  cons <- grep("[^aeiou]", letters)
  name <- paste(sample(LETTERS[cons], 1), sample(letters[vowel], 1),
                sample(letters[cons], 1), sep='')
  gender <- factor(sample(0:1, 1), levels=0:1, labels=c('female','male'))
  dob <- as.Date(sample(7500, 1), origin="1970-01-01")
  n <- sample(6, 1)
  doa <- as.Date(sample(1500, n), origin="2010-01-01")
  pulse <- round(rnorm(n, 80, 10))
  temp <- round(rnorm(n, 98.4, 0.3), 2)
  fluid <- round(runif(n), 2)
  list(name, gender, dob, doa, pulse, temp, fluid)
}
```

1. Create an S3 class `medicalRecord` for objects that are a list with the named elements `name`, `gender`, `date_of_birth`, `date_of_admission`, `pulse`, `temperature`, `fluid_intake`. Note that an individual patient may have multiple measurements for some measurements. Set the RNG seed to `8` and create a medical record by taking the output of `makePatient`. Print the medical record, and print the class of the medical record. (5 points)

```
medRec <- function(){
  patient <- makePatient()
  names(patient) <- c('name', 'gender', 'date_of_birth', 'date_of_admission', 'pulse',
                      'temperature', 'fluid_intake')
  class(patient) <- 'medicalRecord'
  return(patient)
}

set.seed(8)
patient <- medRec()
patient
```

```
## $name
## [1] "Mev"
##
## $gender
## [1] male
## Levels: female male
##
## $date_of_birth
## [1] "1976-08-09"
##
## $date_of_admission
## [1] "2011-03-14" "2013-10-30" "2013-02-27" "2012-08-23" "2011-11-16"
##
## $pulse
## [1] 67 81 95 74 81
##
## $temperature
## [1] 98.33 98.16 99.00 98.49 98.67
##
## $fluid_intake
## [1] 0.62 0.93 0.18 0.39 0.34
##
## attr(,"class")
## [1] "medicalRecord"
```

```r
class(patient)
```

```
## [1] "medicalRecord"
```

2. Write a `medicalRecord` method for the generic function `mean`, which returns averages for pulse, temperature and fluids. Also write a `medicalRecord` method for `print`, which employs some nice formatting, perhaps arranging measurements by date, and `plot`, that generates a composite plot of measurements over time. Call each function for the medical record created in part 1. (5 points)

```r
mean.medicalRecord <- function(patient) {
    pul <- mean(patient$pulse)
    tem <- mean(patient$temperature)
    flu <- mean(patient$fluid_intake)
    out <- list(`Mean Pulse` = pul, `Mean Temperature` = tem, `Mean Fluid Intake` = flu)
    return(out)
}

print.medicalRecord <- function(patient) {
    cat(sprintf("Name: %s\nGender: %s\nDate of Birth: %s", patient$name, patient$gender,
        patient$date_of_birth), "\n\n")
    for (i in 1:length(patient$date_of_admission)) {
        cat(sprintf("Date of Admission: %s\nFluid Intake: %s\nTemperature: %s\nPulse: %s",
            patient$date_of_admission[i], patient$fluid_intake[i], patient$temperature[i],
            patient$pulse[i]), "\n\n")
    }
}

# Since each measurement has such difference scales, I chose to plot them on
```

```r
# separate plots so the graphs could be the most informative.
plot.medicalRecord <- function(patient) {
    sorted <- sort(patient$date_of_admission)
    indices <- c()
    for (i in 1:length(sorted)) {
        indices <- c(indices, which(patient$date_of_admission == sorted[i]))
    }
    temp <- c()
    puls <- c()
    flui <- c()
    for (i in indices) {
        temp <- c(temp, patient$temperature[i])
        puls <- c(puls, patient$pulse[i])
        flui <- c(flui, patient$fluid_intake[i])
    }
    par(mfrow = c(1, 3))
    plot(sorted, temp, type = "b", xaxt = "n", col = "red", xlab = "Date of Admission",
        ylab = "Temperature")
    axis(1, at = sorted, labels = sorted)
    plot(sorted, puls, type = "b", xaxt = "n", col = "green", ylab = "Pulse",
        xlab = "Date of Admission")
    axis(1, at = sorted, labels = sorted)
    plot(sorted, flui, type = "b", xaxt = "n", col = "blue", ylab = "Fluid Intake",
        xlab = "Date of Admission")
    axis(1, at = sorted, labels = sorted)
    mtext(paste("Measurements for Patient: ", patient$name), outer = TRUE)
}
methods(class = "medicalRecord")
```

```
## [1] mean  plot  print
## see '?methods' for accessing help and source code
```

```r
mean(patient)
```

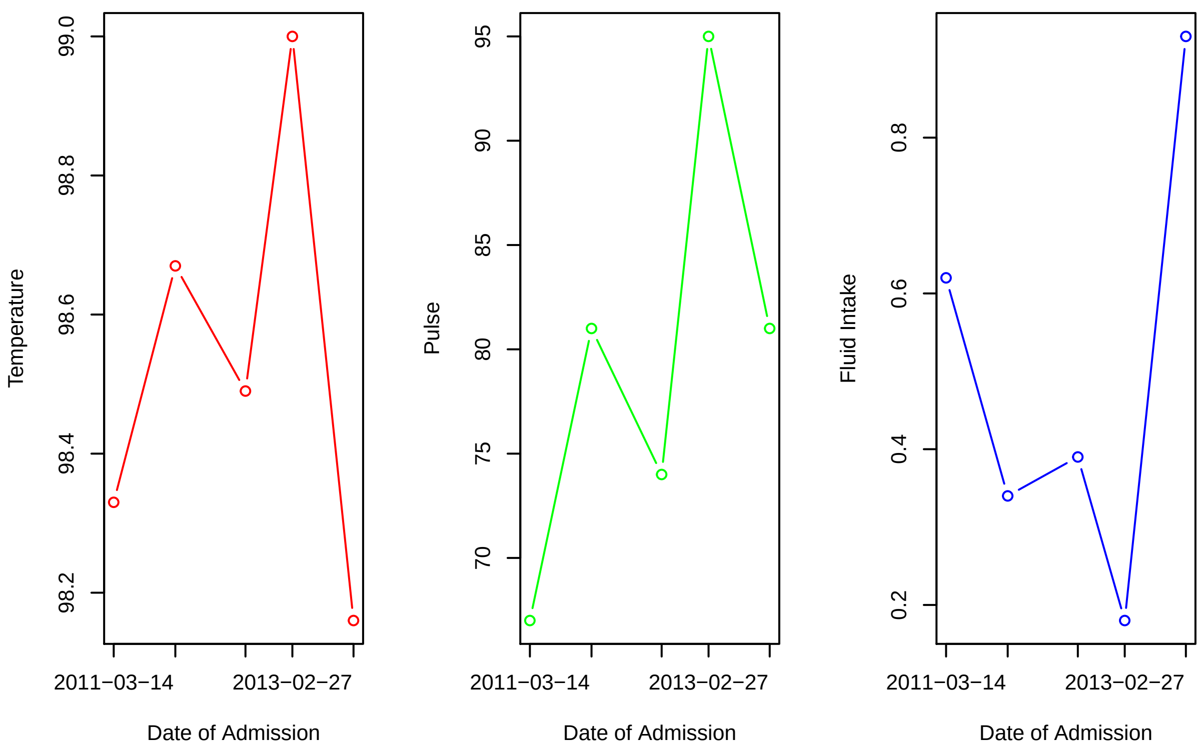
```
## $`Mean Pulse`
## [1] 79.6
##
## $`Mean Temperature`
## [1] 98.53
##
## $`Mean Fluid Intake`
## [1] 0.492
```

```r
print(patient)
```

```
## Name: Mev
## Gender: male
## Date of Birth: 1976-08-09
##
## Date of Admission: 2011-03-14
## Fluid Intake: 0.62
## Temperature: 98.33
```

```
## Pulse: 67
##
## Date of Admission: 2013-10-30
## Fluid Intake: 0.93
## Temperature: 98.16
## Pulse: 81
##
## Date of Admission: 2013-02-27
## Fluid Intake: 0.18
## Temperature: 99
## Pulse: 95
##
## Date of Admission: 2012-08-23
## Fluid Intake: 0.39
## Temperature: 98.49
## Pulse: 74
##
## Date of Admission: 2011-11-16
## Fluid Intake: 0.34
## Temperature: 98.67
## Pulse: 81
```

```
plot(patient)
```



3. Create a further class for a cohort (group) of patients, and write methods for `mean` and `print` which, when applied to a cohort, apply mean or print to each patient contained in the cohort. Hint: think of this as a “container” for patients. Reset the RNG seed to 8 and create a cohort of ten patients, then show the output for `mean` and `print`. (5 points)

```
# Function of how many people you want in cohort
# Assumes there is at least 2 patients in the cohort
cohortmaker <- function(number){
  cohort <- list(medRec())
  for (i in 2:number){
    tempPatient <- medRec()
    cohort[[i]] <- tempPatient
  }
  class(cohort) <- "cohort"
  return(cohort)
}

set.seed(8)
cohort <- cohortmaker(10)

mean.cohort <- function(cohort){
  for (i in 1:length(cohort)){
    cat('--> ', cohort[[i]]$name, '\n')
    print(mean(cohort[[i]]))
  }
}

mean(cohort)
```

```
## -->  Mev
## $`Mean Pulse`
## [1] 79.6
##
## $`Mean Temperature`
## [1] 98.53
##
## $`Mean Fluid Intake`
## [1] 0.492
##
## -->  Yul
## $`Mean Pulse`
## [1] 78
##
## $`Mean Temperature`
## [1] 98.495
##
## $`Mean Fluid Intake`
## [1] 0.245
##
## -->  Zet
## $`Mean Pulse`
## [1] 81.5
##
## $`Mean Temperature`
## [1] 98.44
##
## $`Mean Fluid Intake`
## [1] 0.4033333
```

```
##
## -->  Qih
## $`Mean Pulse`
## [1] 78
##
## $`Mean Temperature`
## [1] 98.6
##
## $`Mean Fluid Intake`
## [1] 0.65
##
## -->  Wut
## $`Mean Pulse`
## [1] 88.33333
##
## $`Mean Temperature`
## [1] 98.05
##
## $`Mean Fluid Intake`
## [1] 0.5866667
##
## -->  Juy
## $`Mean Pulse`
## [1] 83.5
##
## $`Mean Temperature`
## [1] 98.45
##
## $`Mean Fluid Intake`
## [1] 0.4525
##
## -->  God
## $`Mean Pulse`
## [1] 83
##
## $`Mean Temperature`
## [1] 98.01
##
## $`Mean Fluid Intake`
## [1] 0.97
##
## -->  Fut
## $`Mean Pulse`
## [1] 77.5
##
## $`Mean Temperature`
## [1] 98.14833
##
## $`Mean Fluid Intake`
## [1] 0.3366667
##
## -->  Pet
## $`Mean Pulse`
## [1] 77
```

```
##
## $`Mean Temperature`
## [1] 98.83
##
## $`Mean Fluid Intake`
## [1] 0.445
##
## -->  Yed
## $`Mean Pulse`
## [1] 79.33333
##
## $`Mean Temperature`
## [1] 98.3
##
## $`Mean Fluid Intake`
## [1] 0.6583333
```

```
print.cohort <- function(cohort){
  for (i in 1:length(cohort)){
    print(cohort[[i]])
    cat('-----------','\n')
  }
}

print(cohort)
```

```
## Name: Mev
## Gender: male
## Date of Birth: 1976-08-09
##
## Date of Admission: 2011-03-14
## Fluid Intake: 0.62
## Temperature: 98.33
## Pulse: 67
##
## Date of Admission: 2013-10-30
## Fluid Intake: 0.93
## Temperature: 98.16
## Pulse: 81
##
## Date of Admission: 2013-02-27
## Fluid Intake: 0.18
## Temperature: 99
## Pulse: 95
##
## Date of Admission: 2012-08-23
## Fluid Intake: 0.39
## Temperature: 98.49
## Pulse: 74
##
## Date of Admission: 2011-11-16
## Fluid Intake: 0.34
## Temperature: 98.67
## Pulse: 81
```

```
##
## -----------
## Name: Yul
## Gender: male
## Date of Birth: 1988-06-28
##
## Date of Admission: 2012-01-16
## Fluid Intake: 0.14
## Temperature: 98.92
## Pulse: 76
##
## Date of Admission: 2013-08-07
## Fluid Intake: 0.35
## Temperature: 98.07
## Pulse: 80
##
## -----------
## Name: Zet
## Gender: female
## Date of Birth: 1970-06-13
##
## Date of Admission: 2013-11-03
## Fluid Intake: 0.03
## Temperature: 98.54
## Pulse: 72
##
## Date of Admission: 2014-02-05
## Fluid Intake: 0.72
## Temperature: 98.51
## Pulse: 93
##
## Date of Admission: 2013-06-01
## Fluid Intake: 0.25
## Temperature: 98.22
## Pulse: 84
##
## Date of Admission: 2012-08-29
## Fluid Intake: 0.59
## Temperature: 98.47
## Pulse: 88
##
## Date of Admission: 2010-04-01
## Fluid Intake: 0.61
## Temperature: 98.32
## Pulse: 73
##
## Date of Admission: 2010-03-21
## Fluid Intake: 0.22
## Temperature: 98.58
## Pulse: 79
##
## -----------
## Name: Qih
## Gender: female
```

```
## Date of Birth: 1987-08-30
##
## Date of Admission: 2011-06-22
## Fluid Intake: 0.65
## Temperature: 98.6
## Pulse: 78
##
## -----------
## Name: Wut
## Gender: male
## Date of Birth: 1974-06-28
##
## Date of Admission: 2011-02-16
## Fluid Intake: 0.97
## Temperature: 98.26
## Pulse: 93
##
## Date of Admission: 2012-04-12
## Fluid Intake: 0.14
## Temperature: 97.84
## Pulse: 96
##
## Date of Admission: 2010-04-12
## Fluid Intake: 0.65
## Temperature: 98.05
## Pulse: 76
##
## -----------
## Name: Juy
## Gender: male
## Date of Birth: 1983-06-09
##
## Date of Admission: 2010-03-25
## Fluid Intake: 0.26
## Temperature: 98.58
## Pulse: 90
##
## Date of Admission: 2010-06-10
## Fluid Intake: 0.29
## Temperature: 97.53
## Pulse: 88
##
## Date of Admission: 2010-04-18
## Fluid Intake: 0.6
## Temperature: 98.58
## Pulse: 75
##
## Date of Admission: 2010-03-10
## Fluid Intake: 0.66
## Temperature: 99.11
## Pulse: 81
##
## -----------
## Name: God
```

```
## Gender: female
## Date of Birth: 1990-02-12
##
## Date of Admission: 2010-03-12
## Fluid Intake: 0.97
## Temperature: 98.01
## Pulse: 83
##
## -----------
## Name: Fut
## Gender: male
## Date of Birth: 1970-01-11
##
## Date of Admission: 2013-03-15
## Fluid Intake: 0.31
## Temperature: 98.38
## Pulse: 74
##
## Date of Admission: 2011-08-16
## Fluid Intake: 0.13
## Temperature: 98.49
## Pulse: 66
##
## Date of Admission: 2013-11-12
## Fluid Intake: 0.73
## Temperature: 97.83
## Pulse: 88
##
## Date of Admission: 2011-04-14
## Fluid Intake: 0
## Temperature: 97.91
## Pulse: 83
##
## Date of Admission: 2011-04-07
## Fluid Intake: 0.36
## Temperature: 97.87
## Pulse: 80
##
## Date of Admission: 2013-06-20
## Fluid Intake: 0.49
## Temperature: 98.41
## Pulse: 74
##
## -----------
## Name: Pet
## Gender: male
## Date of Birth: 1979-01-01
##
## Date of Admission: 2010-10-30
## Fluid Intake: 0.6
## Temperature: 98.84
## Pulse: 85
##
## Date of Admission: 2012-05-10
```

```
## Fluid Intake: 0.29
## Temperature: 98.82
## Pulse: 69
##
## -----------
## Name: Yed
## Gender: male
## Date of Birth: 1977-11-11
##
## Date of Admission: 2010-07-10
## Fluid Intake: 0.79
## Temperature: 98.65
## Pulse: 98
##
## Date of Admission: 2013-01-06
## Fluid Intake: 0.5
## Temperature: 99.07
## Pulse: 85
##
## Date of Admission: 2010-03-06
## Fluid Intake: 0.67
## Temperature: 98.45
## Pulse: 81
##
## Date of Admission: 2010-01-28
## Fluid Intake: 0.94
## Temperature: 97.95
## Pulse: 63
##
## Date of Admission: 2011-06-18
## Fluid Intake: 0.69
## Temperature: 98
## Pulse: 83
##
## Date of Admission: 2010-08-27
## Fluid Intake: 0.36
## Temperature: 97.68
## Pulse: 66
##
## -----------
```

Question 5

-5 bonus points

Use the simulated results from question 1 to create a three-dimensional pie chart (actually, don't).