Homework 6

Molly Olson

November 23, 2015

```
library(graphics)
library(ggplot2)
```

Question 1

Consider the following very simple genetic model (very simple – don't worry if youre 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 the height change across generations?

Represent the heights of the the current generation of 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))
```

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 dataframe with the same structure, given the heights of the next generation.*

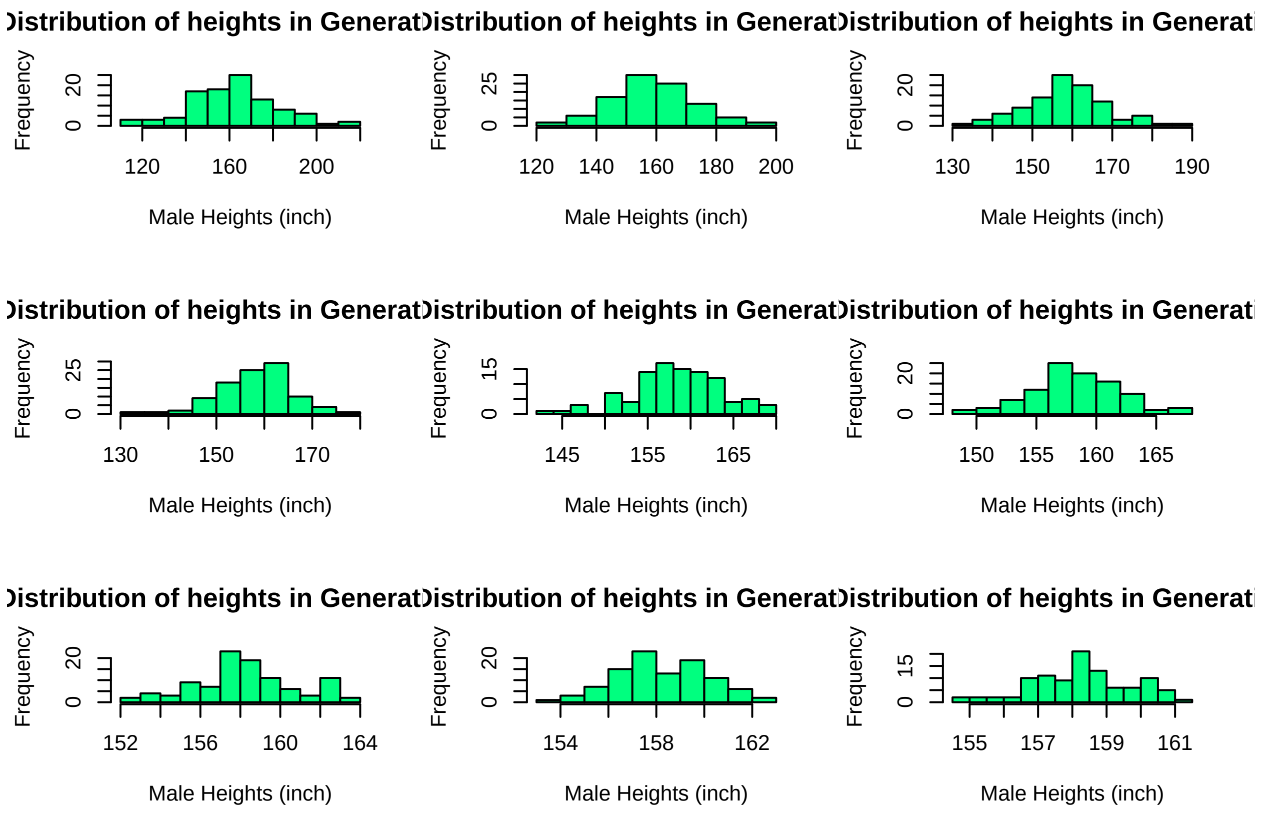
```
next_gen <- function(pop) {
  pop$m <- sample(pop$m)
  pop$m <- rowMeans(pop)
  pop$f <- pop$m
  pop
}
```

Using 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 phoenomenon you see is called regression to the mean. Provide (at least) minimal decorations such as title and x-axis labels.*

```
gen1 <- pop
gen2 <- next_gen(gen1)
gen3 <- next_gen(gen2)
gen4 <- next_gen(gen3)
gen5 <- next_gen(gen4)
gen6 <- next_gen(gen5)
gen7 <- next_gen(gen6)
gen8 <- next_gen(gen7)
gen9 <- next_gen(gen8)
```

```r
dist.pop <- function(generation, title){
  hist(generation$m, breaks = 10, xlab = "Male Heights (inch)", main = title, col='springgreen')
}
```

```r
par(mfrow=c(3,3))
dist.pop(gen1, "Distribution of heights in Generation 1")
dist.pop(gen2, "Distribution of heights in Generation 2")
dist.pop(gen3, "Distribution of heights in Generation 3")
dist.pop(gen4, "Distribution of heights in Generation 4")
dist.pop(gen5, "Distribution of heights in Generation 5")
dist.pop(gen6, "Distribution of heights in Generation 6")
dist.pop(gen7, "Distribution of heights in Generation 7")
dist.pop(gen8, "Distribution of heights in Generation 8")
dist.pop(gen9, "Distribution of heights in Generation 9")
```



Question 2

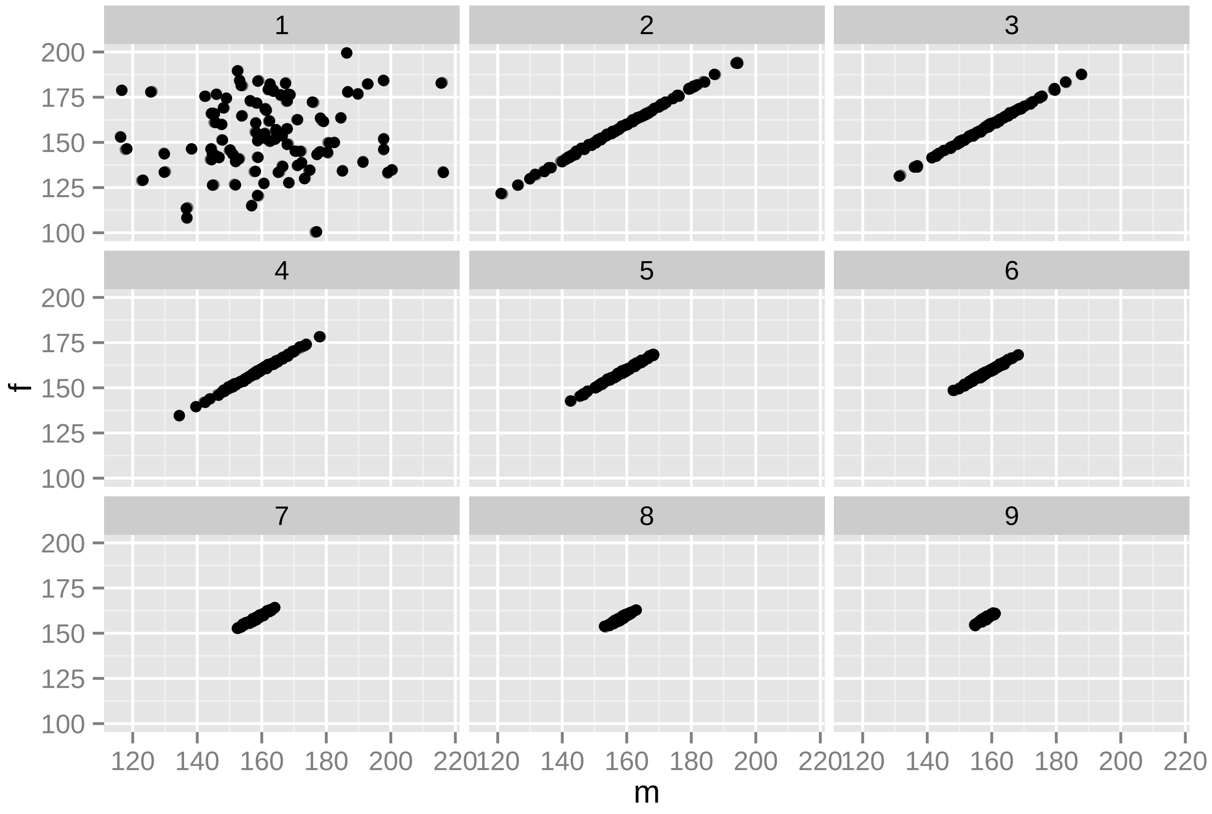
Use the simulated results from question 1 to reproduce (as closely as possible) the following plot in ggplot2. You would have all the values plotted in the same columns with another column indicating which generation they belong to then facet on that column

```r
gen1name <- data.frame(rep('1',length(gen1)))
colnames(gen1name) <- 'generation'

gen2name <- data.frame(rep('2',length(gen1)))
```

```r
colnames(gen2name) <- 'generation'

gen3name <- data.frame(rep('3',length(gen3)))
colnames(gen3name) <- 'generation'

gen4name <- data.frame(rep('4',length(gen4)))
colnames(gen4name) <- 'generation'

gen5name <- data.frame(rep('5',length(gen5)))
colnames(gen5name) <- 'generation'

gen6name <- data.frame(rep('6',length(gen6)))
colnames(gen6name) <- 'generation'

gen7name <- data.frame(rep('7',length(gen7)))
colnames(gen7name) <- 'generation'

gen8name <- data.frame(rep('8',length(gen8)))
colnames(gen8name) <- 'generation'

gen9name <- data.frame(rep('9',length(gen9)))
colnames(gen9name) <- 'generation'



gen1dat <- cbind(gen1, gen1name)
colnames(gen1dat) <- c('male','female','generation')
gen2dat <- cbind(gen2, gen2name)
colnames(gen2dat) <- c('male','female','generation')
gen3dat <- cbind(gen3, gen3name)
colnames(gen3dat) <- c('male','female','generation')
gen4dat <- cbind(gen4, gen4name)
colnames(gen4dat) <- c('male','female','generation')
gen5dat <- cbind(gen5, gen5name)
colnames(gen5dat) <- c('male','female','generation')
gen6dat <- cbind(gen6, gen6name)
colnames(gen6dat) <- c('male','female','generation')
gen7dat <- cbind(gen7, gen7name)
colnames(gen7dat) <- c('male','female','generation')
gen8dat <- cbind(gen8, gen8name)
colnames(gen8dat) <- c('male','female','generation')
gen9dat <- cbind(gen9, gen9name)
colnames(gen9dat) <- c('male','female','generation')
```

```r
newdat <- rbind(gen1dat,gen2dat,gen3dat,gen4dat,gen5dat,gen6dat,gen7dat,gen8dat,gen9dat)
colnames(newdat) <- c('m','f','generation')
newdat[,1] <- as.numeric(newdat[,1])
newdat[,2] <- as.numeric(newdat[,2])
newdat[,3] <- as.integer(newdat[,3])
```

```
ggplot(newdat, aes(x = m, y = f)) + geom_point(alpha = 0.5) + geom_point(position=position_jitter(w=0.5
```



Question 3

You calculated the power of a study design in question 2 of assignment 3. the study has two variables, treatment group and outcome. There are two treatment groups (0,1) and they should be assigned randomly with equal probability. The outcome should be a random normal variable with a mean of 60 and a standard deviation of 20. If a patient is in the treatment group, add 5 to the outcome.

```
#the function takes the number of patients. It is one procedure through and will be used to repeat the
treatment <- function(sampleSize){

  #treatment groups
  treatmentGroups <- c(0,1)
  # assign treatments for n patients, we can use sample because
  #there is equal probability of being in either treatment group
  treatmentPerPatient <- sample(treatmentGroups, sampleSize, replace=TRUE)
  #treatments

  #outcome should be normal with mu=60 and Sigma=20
  outcome <- rnorm(sampleSize, mean = 60, sd = 20)

  #if the patient is in the treatment group, i=1, then add 5 to it, otherwise do nothing
  treatmentOutcome <- ifelse(treatmentPerPatient==1, outcome + 5, outcome)
  #treatmentOutcome
treatgroup <- treatmentOutcome[treatmentPerPatient == 1]
placebogroup <- treatmentOutcome[treatmentPerPatient ==0]

bmeansTreat <- replicate(1000, mean(sample(treatgroup, length(treatgroup), replace=T)))
pTreat <- quantile(bmeansTreat, c(0.025,0.975))

bmeansPlacebo <- replicate(1000, mean(sample(placebogroup, length(placebogroup), replace=T)))
pPlacebo <- quantile(bmeansPlacebo, c(0.025,0.975))

bind <- cbind(mean(bmeansTreat),pTreat[1],pTreat[2],mean(bmeansPlacebo),pPlacebo[1],pPlacebo[2])
rownames(bind) <- NULL
  return(bind)
}
```

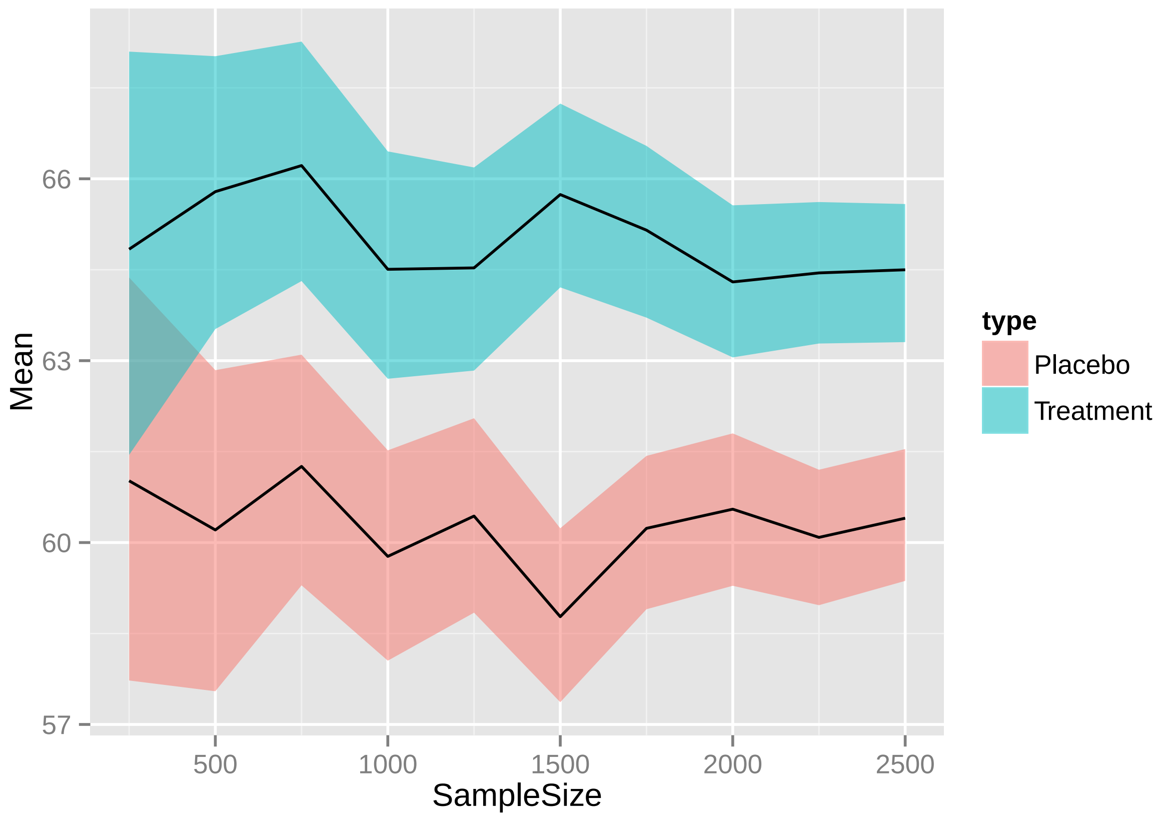
Starting with a sample size of 250, create a 95% bootstrap percentile interval for the mean of each group. Then create a new bootstrap interval by increasing the sample size by 250 until the sample is 2500. Thus you will create a total of 10 boostrap intervals. Each boostrap should create 1000 bootstrap samples.

```
bootstrapInts <- data.frame(matrix(NA, nrow=10,ncol=7))
colnames(bootstrapInts) <- c('TreatmentMean','t2.5%','t97.5%','PlaceboMean','p2.5%','p97.5%','SampleSiz
for(i in 1:10){
 #print(250*i)
 #treatment(250*i)
 #print(treatment(250*i))
 bootstrapInts[i,] <- c(treatment(250*i),250*i)
}
```

```
df1 <- bootstrapInts[,c(1,2,3,7)]
df2 <- bootstrapInts[,c(4,5,6,7)]

df1$type <- "Treatment"
df2$type <- "Placebo"
names(df1) <- c("Mean","2.5%", "97.5%","SampleSize", "type")
names(df2) <- c("Mean","2.5%", "97.5%","SampleSize", "type")
df <- rbind(df1,df2)
```

Produce a line chart that includes the bootstrapped mean and lower and upper percentile intervals for appropritae labels and a legend You may use base graphics or ggplot2, it should look similar to plot provided

```
ggplot(df, aes(x=SampleSize)) +
  geom_ribbon(aes(ymin=`2.5%`, ymax=`97.5%`, fill = type), alpha=0.5) +
  geom_line(aes(y = Mean, group = type))
```



Question 4

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 = name, gender = gender, date_of_birth = dob, date_of_admission = doa, pulse = pulse, tempe
}
```

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.

```r
set.seed(8)
p <- makePatient()
class(p) <- 'medicalRecord'
p
```

```
## $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
## [1] 0.62 0.93 0.18 0.39 0.34
##
## attr(,"class")
## [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.

```r
mean.medicalRecord <- function(patient){
  pulseAve = mean(patient$pulse)
  tempAve = mean(patient$temperature)
  fluidAve = mean(patient$fluid)
  return(list(pulseAve = pulseAve,tempAve = tempAve,fluidAve =fluidAve))
}
```

```r
print.medicalRecord <- function(patient){
  #made a data frame that contained doa and 3 measurements
  #can subset dataframe using order of data admission col
  cat("name:", patient$name, "\n")
  cat("gender: ", as.character(patient$gender), "\n")
  cat(sprintf("Born on: %s", patient$date_of_birth), "\n")
  df <- data.frame(DateOfAdmission = patient$date_of_admission, Pulse = patient$pulse, Temperature = pa
  df <- df[order(df$DateOfAdmission),]
  print(df)
}
```

```r
mean.medicalRecord(p)
```

```
## $pulseAve
## [1] 79.6
##
## $tempAve
## [1] 98.53
##
## $fluidAve
## [1] 0.492
```

```r
print.medicalRecord(p)
```

```
## name: Mev
## gender:  male
## Born on: 1976-08-09
##   DateOfAdmission Pulse Temperature Fluid
## 1      2011-03-14    67       98.33  0.62
## 5      2011-11-16    81       98.67  0.34
## 4      2012-08-23    74       98.49  0.39
## 3      2013-02-27    95       99.00  0.18
## 2      2013-10-30    81       98.16  0.93
```

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`.

```r
set.seed(8)
group <- replicate(10, list(makePatient()))
#cohort
```

```r
mean.cohort <- function(cohort){
  counter <- 0
  for(i in group){
    counter <- counter + 1
    cat(sprintf("Patient #: %s. Name: %s", counter, i$name), "\n")
    print(mean.medicalRecord(i))
  }
}
```

```r
mean.cohort(group)
```

```
## Patient #: 1. Name: Mev
## $pulseAve
## [1] 79.6
##
## $tempAve
## [1] 98.53
##
## $fluidAve
## [1] 0.492
##
## Patient #: 2. Name: Yul
## $pulseAve
## [1] 78
##
## $tempAve
## [1] 98.495
##
## $fluidAve
## [1] 0.245
##
## Patient #: 3. Name: Zet
## $pulseAve
## [1] 81.5
##
## $tempAve
## [1] 98.44
##
## $fluidAve
## [1] 0.4033333
##
## Patient #: 4. Name: Qih
## $pulseAve
## [1] 78
##
## $tempAve
## [1] 98.6
##
## $fluidAve
## [1] 0.65
##
## Patient #: 5. Name: Wut
## $pulseAve
## [1] 88.33333
##
## $tempAve
## [1] 98.05
##
## $fluidAve
## [1] 0.5866667
##
## Patient #: 6. Name: Juy
## $pulseAve
## [1] 83.5
##
## $tempAve
## [1] 98.45
##
## $fluidAve
## [1] 0.4525
##
## Patient #: 7. Name: God
## $pulseAve
## [1] 83
##
## $tempAve
## [1] 98.01
##
## $fluidAve
## [1] 0.97
##
## Patient #: 8. Name: Fut
## $pulseAve
## [1] 77.5
##
## $tempAve
## [1] 98.14833
##
## $fluidAve
## [1] 0.3366667
##
## Patient #: 9. Name: Pet
## $pulseAve
## [1] 77
##
## $tempAve
## [1] 98.83
##
## $fluidAve
## [1] 0.445
##
## Patient #: 10. Name: Yed
## $pulseAve
## [1] 79.33333
##
## $tempAve
## [1] 98.3
##
## $fluidAve
## [1] 0.6583333
```

```
print.cohort <- function(cohort){
  for(i in group)
    print.medicalRecord(i)
}
```

```
print.cohort(group)
```

```
## name: Mev
## gender:  male
## Born on: 1976-08-09
##   DateOfAdmission Pulse Temperature Fluid
## 1      2011-03-14    67       98.33  0.62
## 5      2011-11-16    81       98.67  0.34
## 4      2012-08-23    74       98.49  0.39
## 3      2013-02-27    95       99.00  0.18
## 2      2013-10-30    81       98.16  0.93
## name: Yul
## gender:  male
## Born on: 1988-06-28
##   DateOfAdmission Pulse Temperature Fluid
## 1      2012-01-16    76       98.92  0.14
## 2      2013-08-07    80       98.07  0.35
## name: Zet
## gender:  female
## Born on: 1970-06-13
##   DateOfAdmission Pulse Temperature Fluid
## 6      2010-03-21    79       98.58  0.22
## 5      2010-04-01    73       98.32  0.61
## 4      2012-08-29    88       98.47  0.59
## 3      2013-06-01    84       98.22  0.25
## 1      2013-11-03    72       98.54  0.03
## 2      2014-02-05    93       98.51  0.72
## name: Qih
## gender:  female
## Born on: 1987-08-30
##   DateOfAdmission Pulse Temperature Fluid
## 1      2011-06-22    78        98.6  0.65
## name: Wut
## gender:  male
## Born on: 1974-06-28
##   DateOfAdmission Pulse Temperature Fluid
## 3      2010-04-12    76       98.05  0.65
## 1      2011-02-16    93       98.26  0.97
## 2      2012-04-12    96       97.84  0.14
## name: Juy
## gender:  male
## Born on: 1983-06-09
##   DateOfAdmission Pulse Temperature Fluid
## 4      2010-03-10    81       99.11  0.66
## 1      2010-03-25    90       98.58  0.26
## 3      2010-04-18    75       98.58  0.60
## 2      2010-06-10    88       97.53  0.29
## name: God
## gender:  female
## Born on: 1990-02-12
##   DateOfAdmission Pulse Temperature Fluid
## 1      2010-03-12    83       98.01  0.97
## name: Fut
## gender:  male
## Born on: 1970-01-11
##   DateOfAdmission Pulse Temperature Fluid
## 5      2011-04-07    80       97.87  0.36
## 4      2011-04-14    83       97.91  0.00
## 2      2011-08-16    66       98.49  0.13
## 1      2013-03-15    74       98.38  0.31
## 6      2013-06-20    74       98.41  0.49
## 3      2013-11-12    88       97.83  0.73
## name: Pet
## gender:  male
## Born on: 1979-01-01
##   DateOfAdmission Pulse Temperature Fluid
## 1      2010-10-30    85       98.84  0.60
## 2      2012-05-10    69       98.82  0.29
## name: Yed
## gender:  male
## Born on: 1977-11-11
##   DateOfAdmission Pulse Temperature Fluid
## 4      2010-01-28    63       97.95  0.94
## 3      2010-03-06    81       98.45  0.67
## 1      2010-07-10    98       98.65  0.79
## 6      2010-08-27    66       97.68  0.36
## 5      2011-06-18    83       98.00  0.69
## 2      2013-01-06    85       99.07  0.50
```