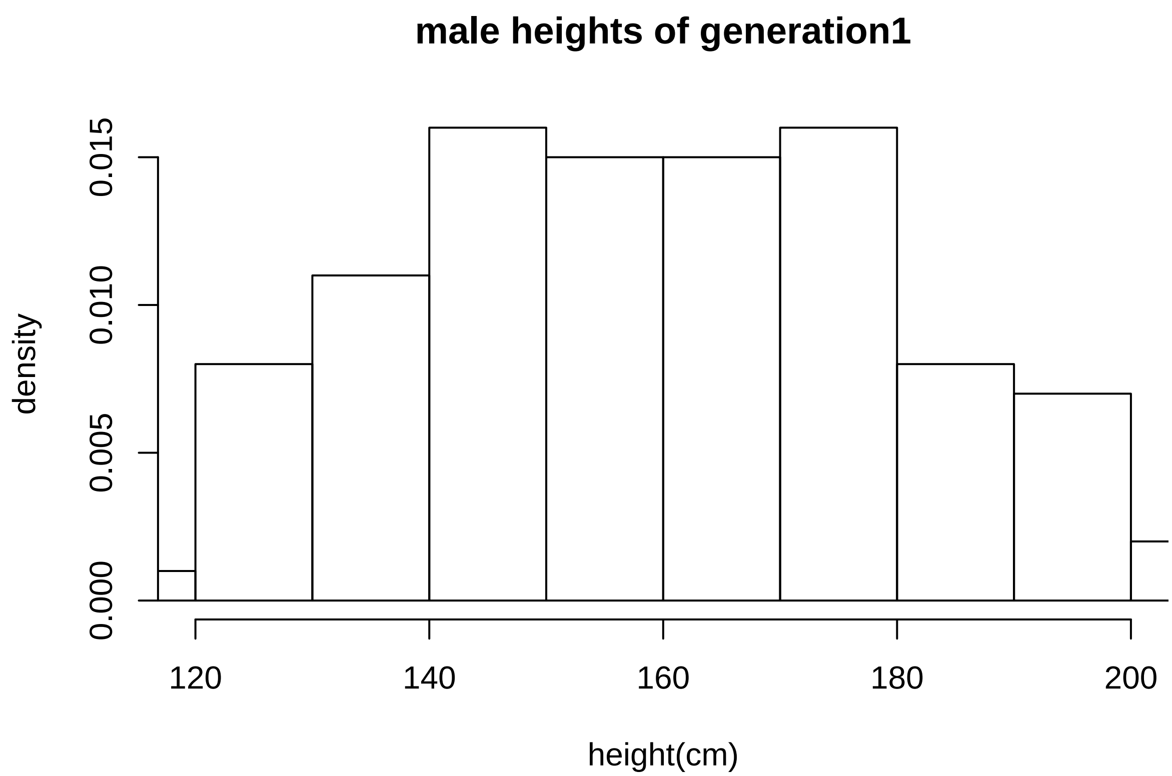
Bios 6301: Assignment 6

Wooyeol Lee

Question 1

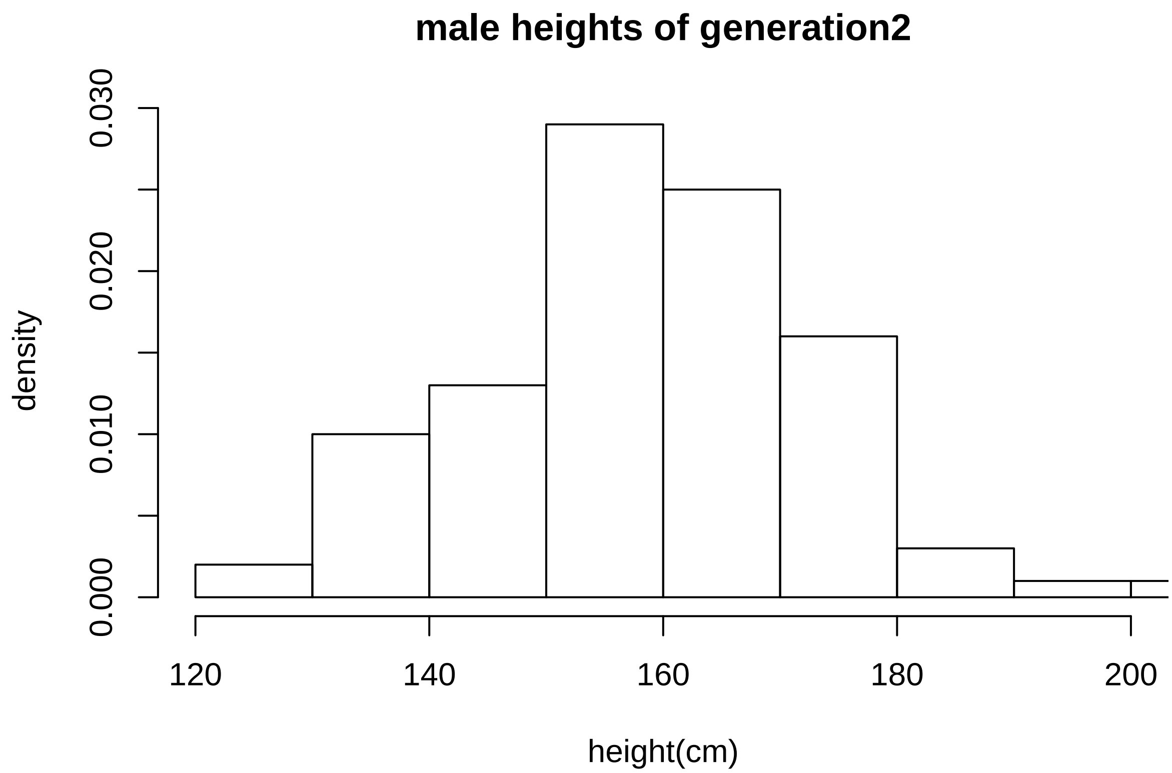
Use the function `next_gen` to generate nine generations (you already have the first), then use the function `hist` to plot the distribution of male heights in each generation (this will require multiple calls to `hist`). The phenomenon you see is called regression to the mean. Provide (at least) minimal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

```
###### generation 1
pop1 <- data.frame(m = rnorm(100, 160, 20), f = rnorm(100, 160, 20))
hist(pop1$m, freq=F,xlim=c(120,200), xlab="height(cm)", ylab="density", main="male heights of generatio
```

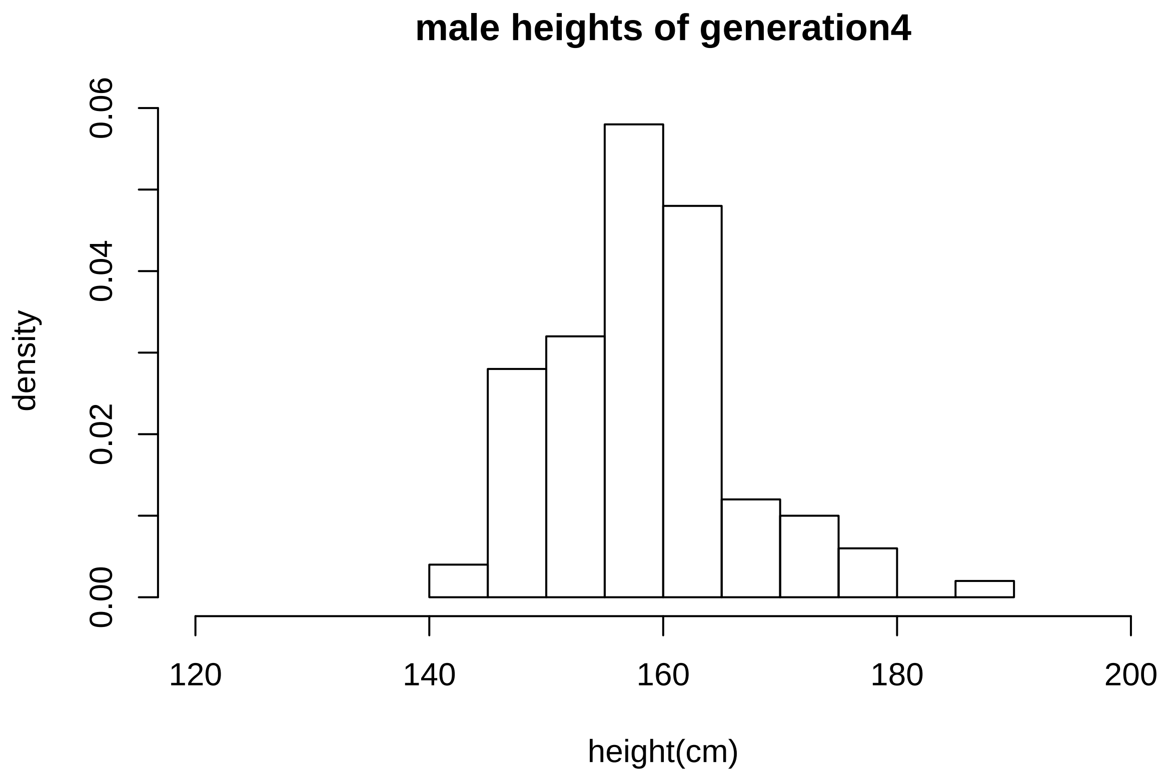


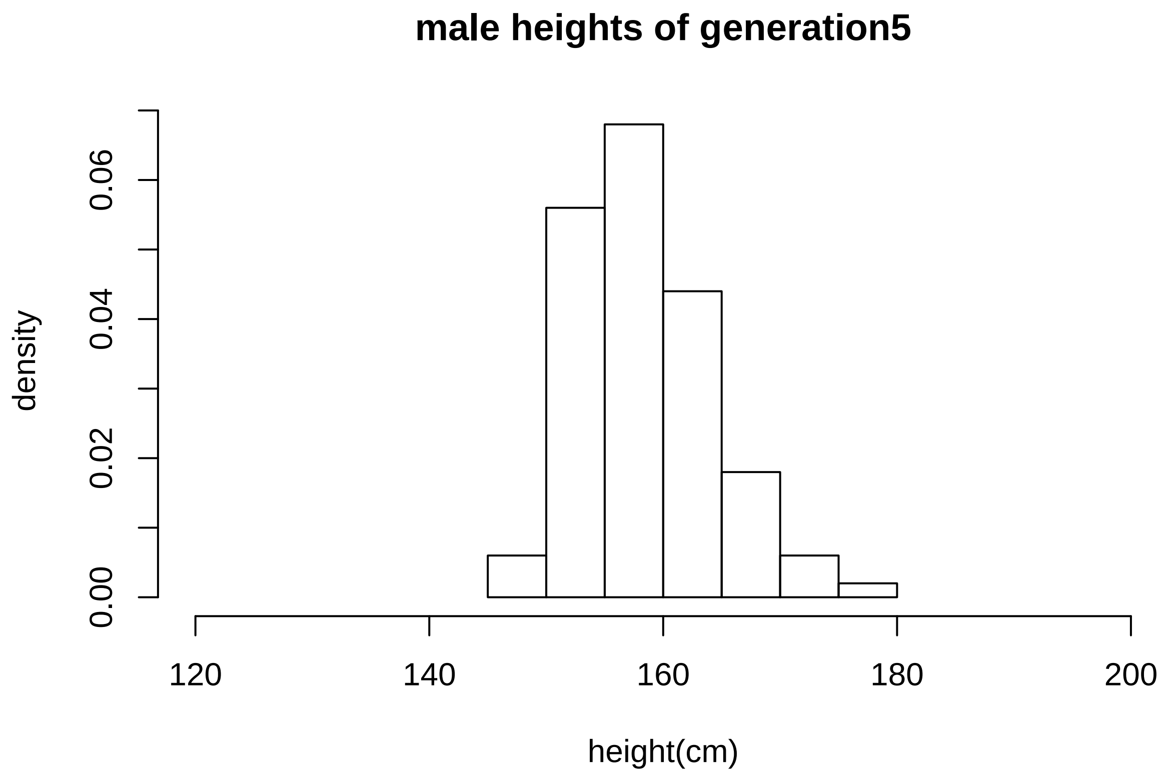

```
####### create a function
next_gen <- function(pop) {
  pop$m <- sample(pop$m)
  pop$m <- rowMeans(pop)
  pop$f <- pop$m
  pop
}
####### loop from gen2 to gen9
for(i in 2:9) {
```

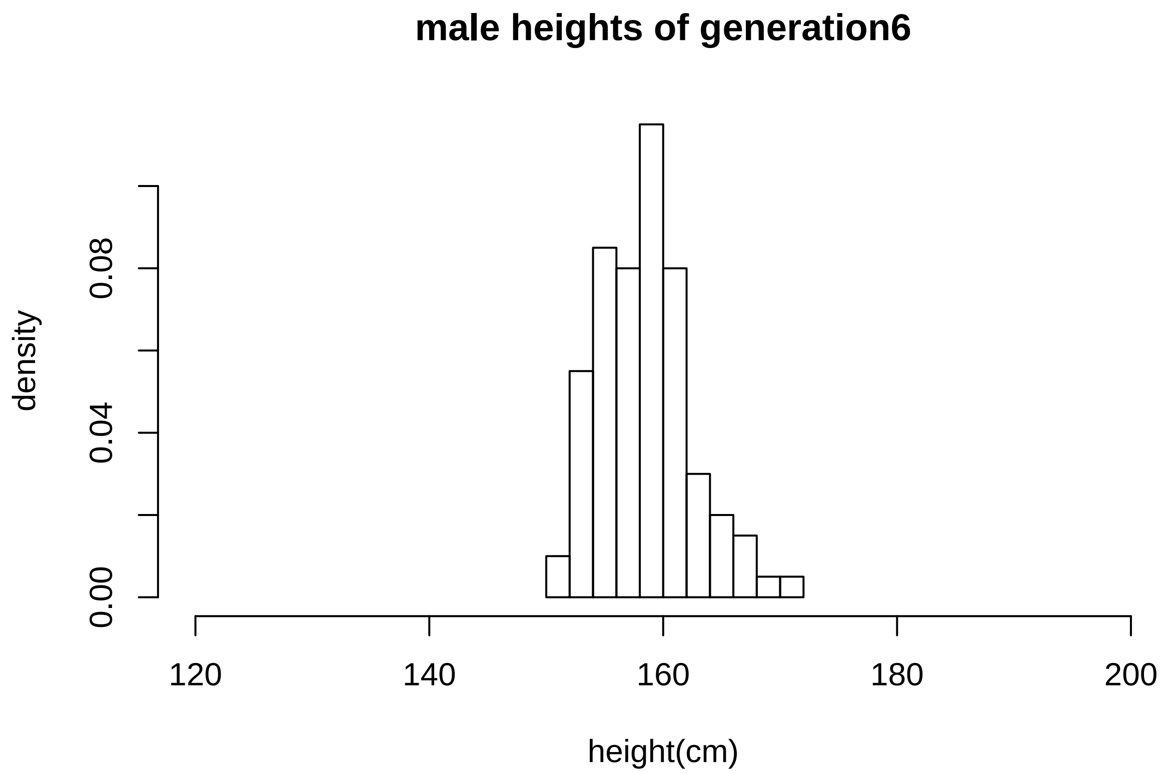
```
  vname <- paste("pop",i, sep="")         ###variable name
  argname <- paste("pop",i-1, sep="")    ###argument name
  assign(vname, next_gen(get(argname)))  ###data generation
  xlim=c(120,200)
  xlab="height(cm)"
  ylab="density"
  main=paste("male heights of generation",i, sep="")
  hist(get(vname)$m, freq=F,xlim=xlim, xlab=xlab, ylab=ylab, main=main)  ###histogram
}
```

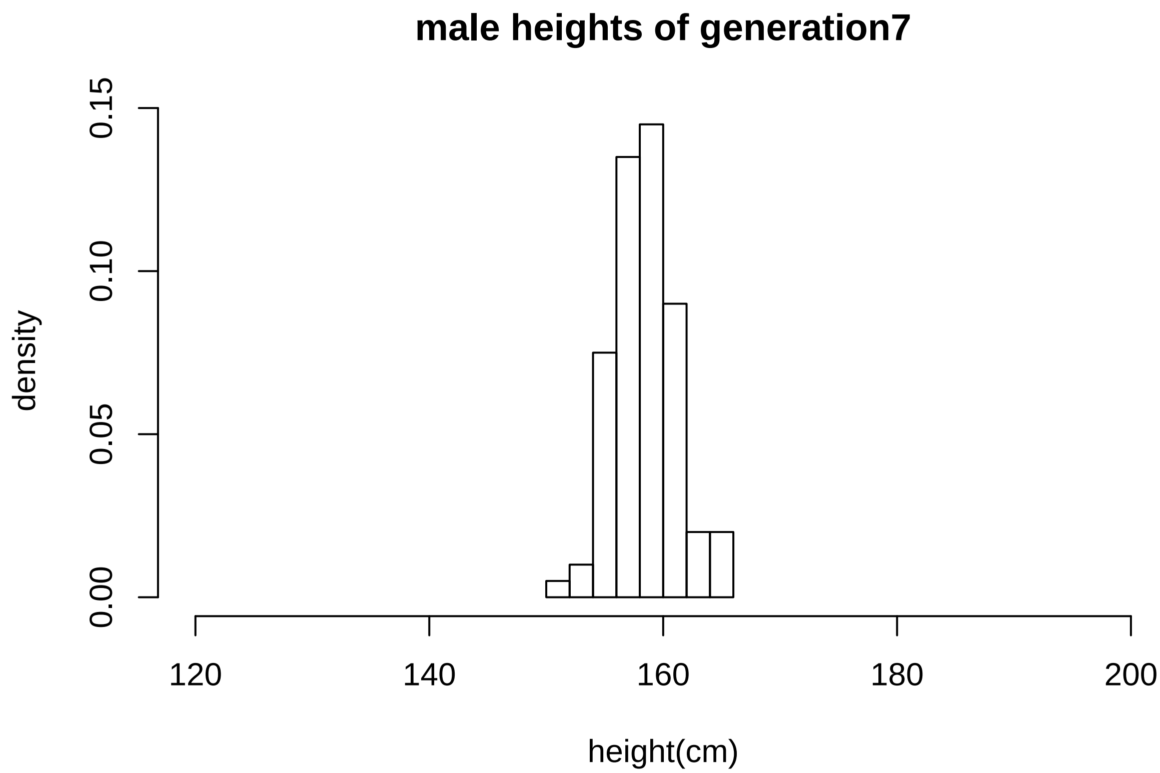

male heights of generation2
density
0.000
0.010
0.020
0.030
120
140
160
180
200
height(cm)

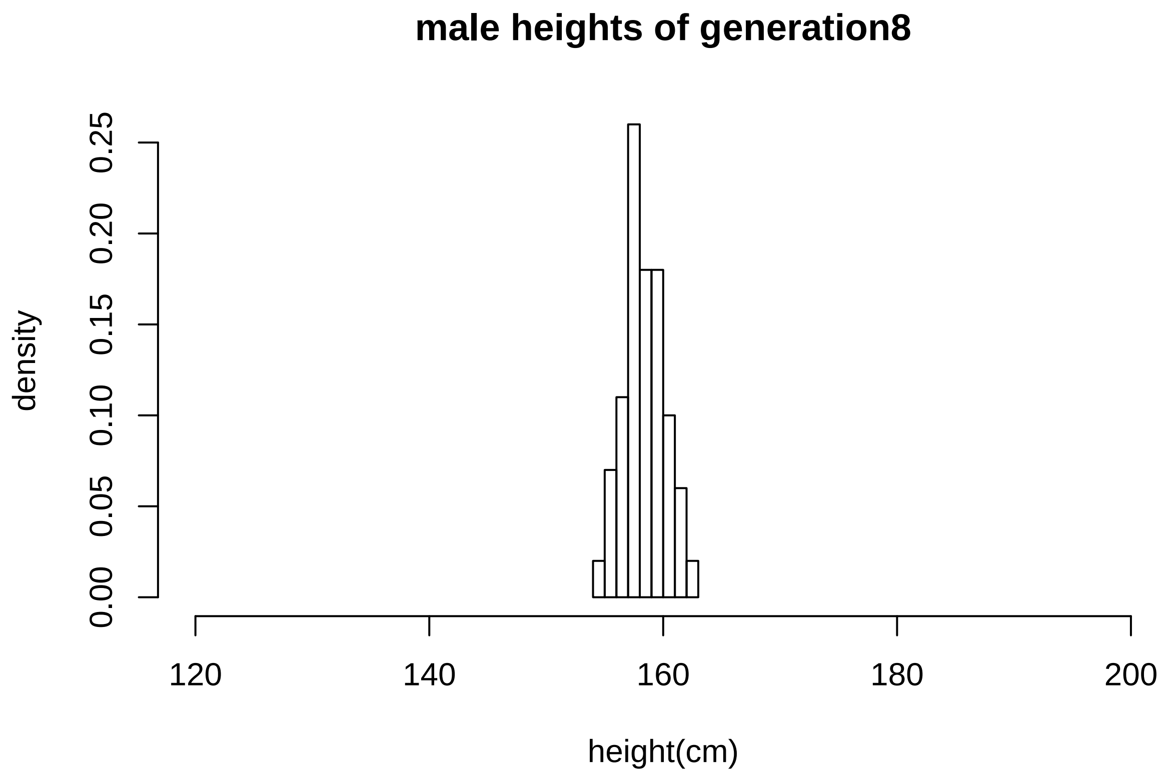

male heights of generation3
density
0.000
0.010
0.020
0.030
120
140
160
180
200
height(cm)


male heights of generation4
density
0.00
0.02
0.04
0.06
120
140
160
180
200
height(cm)

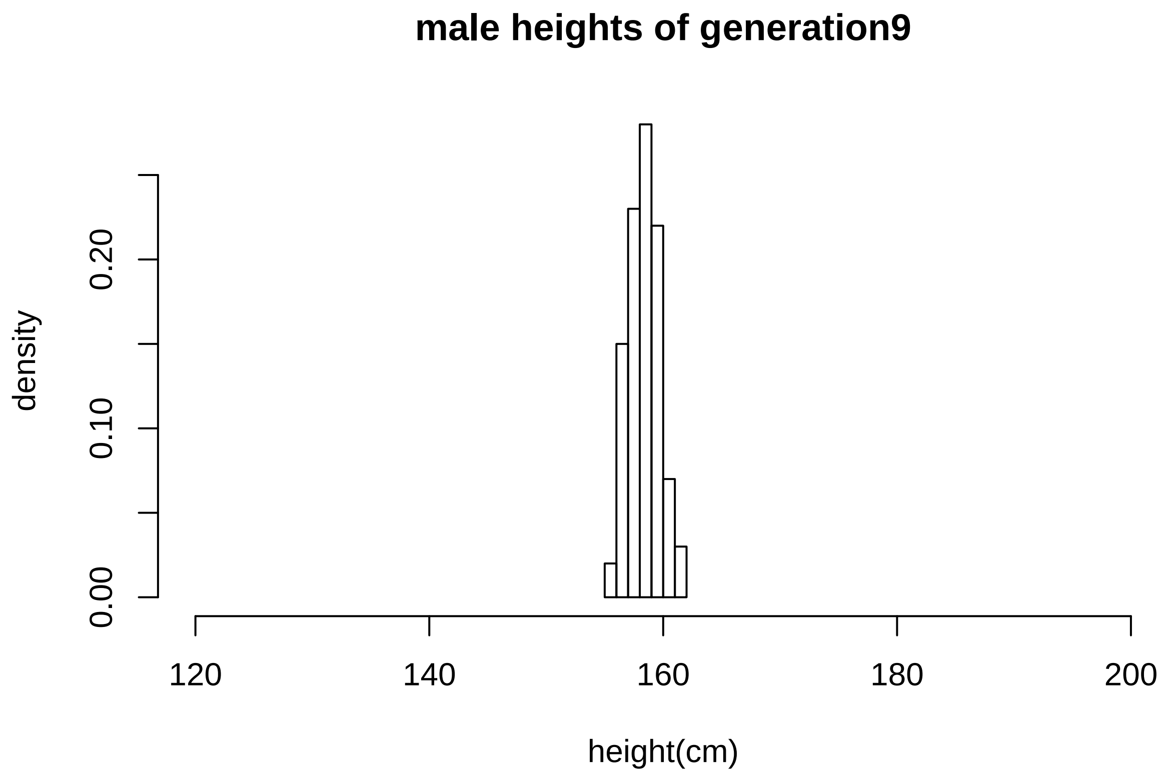

male heights of generation5
density
0.00
0.02
0.04
0.06
120
140
160
180
200
height(cm)


male heights of generation6
density
0.00
0.04
0.08
120
140
160
180
200
height(cm)


male heights of generation7
density
0.00
0.05
0.10
0.15
120
140
160
180
200
height(cm)



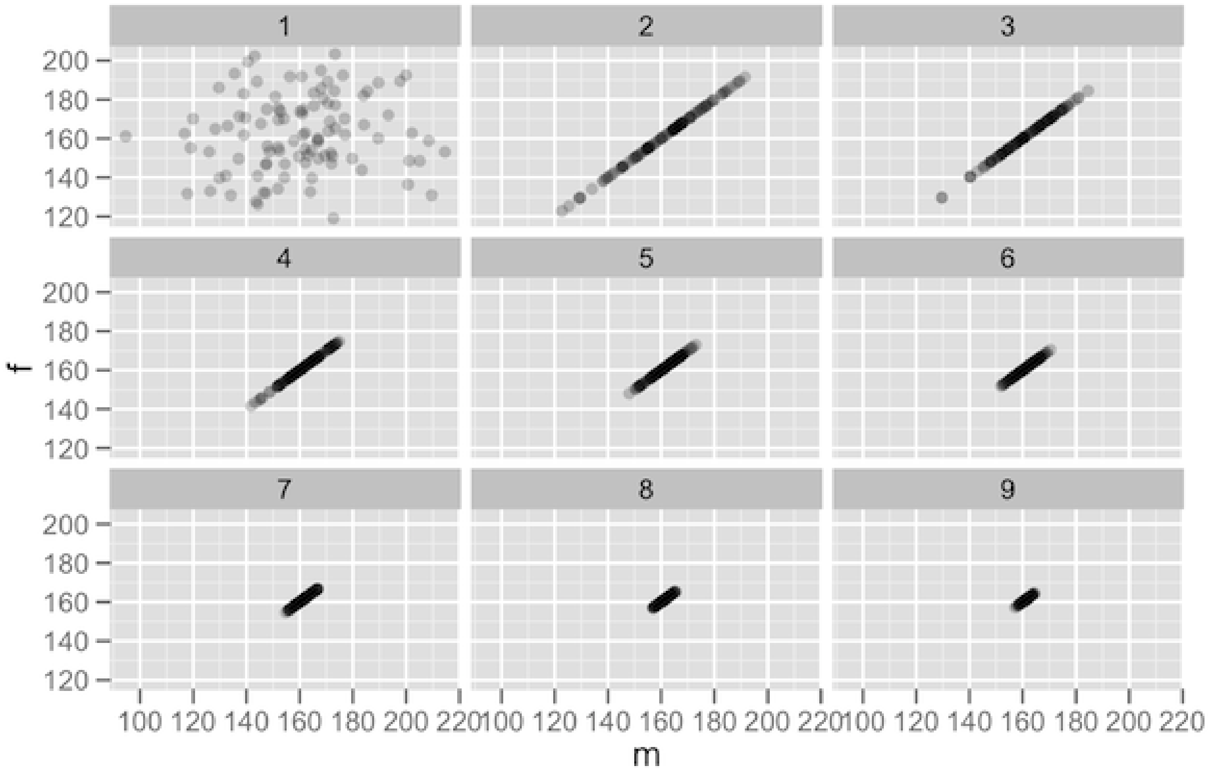
male heights of generation8
density
0.00
0.05
0.10
0.15
0.20
0.25
120
140
160
180
200
height(cm)



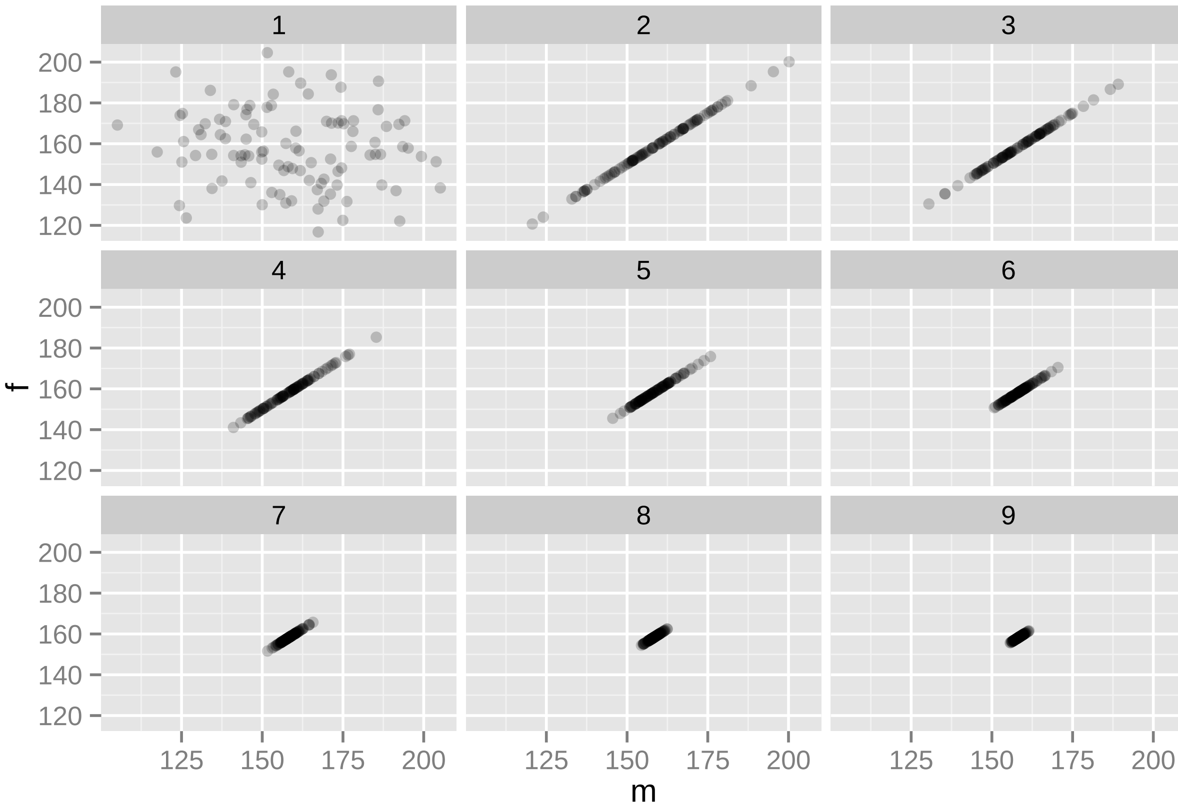
Question 2

10 points

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




```
##question 2
library(ggplot2)
data<- rbind(pop1,pop2,pop3,pop4,pop5,pop6,pop7,pop8,pop9)
gen<-c(rep(1:9, each=100))
data<- cbind(data, gen)
#ggplot
p <- ggplot(data, aes(m, f))
p + geom_point(alpha = 1/5) +facet_wrap(~gen)
```



Question 3

10 points

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 standard deviation of 20. If a patient is in the treatment group, add 5 to the outcome.

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 bootstrap intervals. Each bootstrap should create 1000 bootstrap samples. (4 points)

```
##make a data frame
dat <- data.frame(numeric(0),numeric(0),numeric(0),numeric(0),numeric(0),numeric(0),numeric(0))
## r loop: start with 250, incresing by 250
for(j in 1:10) {
    # 1. create population 250 to 2500
  n <- j*250
  grp <- sample(0:1, n, replace=TRUE)
  out <- rnorm(n, 60, 20)
  out[grp == 1] <- out[grp == 1] + 5
  pop<-data.frame(grp,out)

  # 2. create bootstrap sample, size1000
  ssize<-1000
```

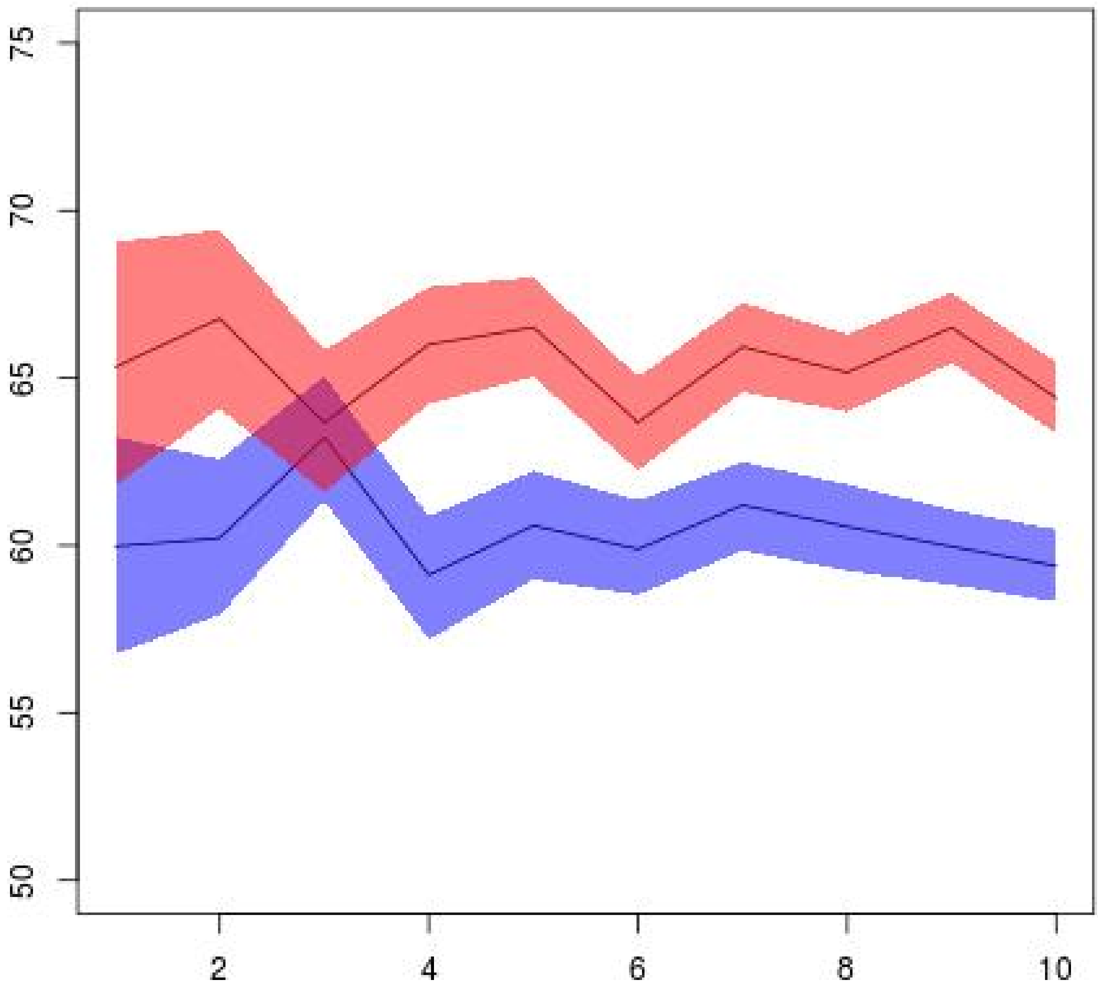
```
  boot<-data.frame(numeric(ssize),numeric(ssize))
  for(i in 1:ssize) {
    boot[i,1] <- mean(sample(pop$out[grp==1], n, replace=T))
    boot[i,2] <- mean(sample(pop$out[grp==0], n, replace=T))
  }
  g1<-quantile(boot[,1], probs=c(0.05,0.5,0.95))
  g0<-quantile(boot[,2], probs=c(0.05,0.5,0.95))

  values <- c(n, g1,g0)
  dat <- rbind(dat,values)
}
colnames(dat)<- c('n','L1','M1','U1','L0','M0','U0')
dat
```

```
##       n       L1       M1       U1       L0       M0       U0
## 1   250 61.18139 63.52202 65.70792 57.03220 58.98309 60.93161
## 2   500 64.58408 65.89520 67.23378 59.52481 61.02976 62.57395
## 3   750 65.06144 66.20493 67.49834 59.67068 60.86511 62.03918
## 4  1000 64.39408 65.39605 66.36316 57.43133 58.47563 59.48043
## 5  1250 64.13940 65.06242 66.03024 58.85749 59.74223 60.64908
## 6  1500 64.97466 65.78769 66.65674 58.56339 59.45299 60.32788
## 7  1750 63.91025 64.68256 65.48819 61.41003 62.18278 62.96823
## 8  2000 64.19267 64.90177 65.62760 59.02044 59.80075 60.52944
## 9  2250 65.03454 65.68345 66.31138 59.84073 60.52853 61.20385
## 10 2500 65.10645 65.80605 66.47775 59.78908 60.45211 61.04534
```

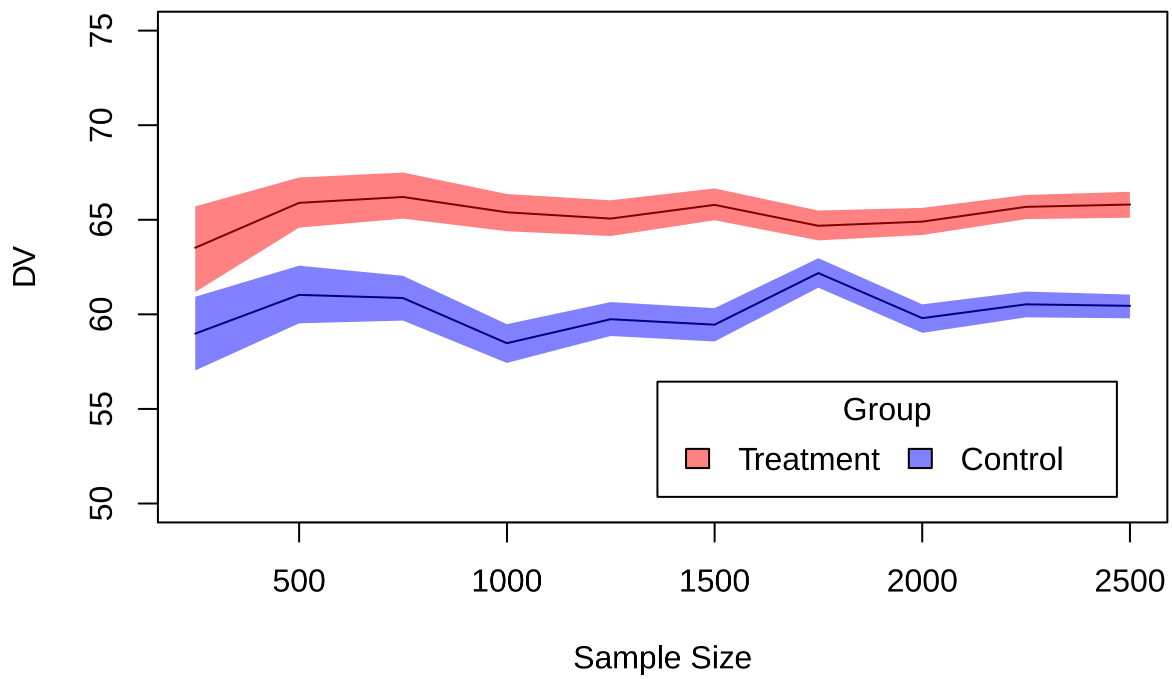
Produce a line chart that includes the bootstrapped mean and lower and upper percentile intervals for each group. Add appropriate labels and a legend. (6 points)

You may use base graphics or ggplot2. It should look similar to this (in base).




```
##draw a plot
makeTransparent = function(..., alpha=0.5) {
  if(alpha<0 | alpha>1) stop("alpha must be between 0 and 1")
  alpha = floor(255*alpha)
  newColor = col2rgb(col=unlist(list(...)), alpha=FALSE)
  .makeTransparent = function(col, alpha) {
    rgb(red=col[1], green=col[2], blue=col[3], alpha=alpha, maxColorValue=255)
  }
  newColor = apply(newColor, 2, .makeTransparent, alpha=alpha)
  return(newColor)
}
t.red <- makeTransparent('red',alpha=0.5)
t.blue<- makeTransparent('blue',alpha=0.5)
par(new=FALSE)
```

```r
plot(NULL,
     xlim=c(250, 2500),
     ylim=c(50,75),
     xlab="Sample Size",
     ylab="DV"
)
lines(dat$n, dat$M1)
lines(dat$n, dat$M0)
polygon(c(rev(dat$n), dat$n), c(rev(dat$U1),dat$L1),col=t.red, border = NA)
polygon(c(rev(dat$n), dat$n), c(rev(dat$U0),dat$L0),col=t.blue, border = NA)
## Add a legend.
legend("bottomright", inset=.05, title="Group",
       c("Treatment","Control"), fill=c(t.red, t.blue), horiz=TRUE)
```



Question 4

15 points

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

```r
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'))
```

```r
  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)

```r
set.seed(8)
j<-makePatient()
class(j) <-"medicalRecord"
names(j) <- c("name","gender","date_of_birth","date_of_admission","pulse","temp","fluid")
print(j)      ##Print the medical record
```

```
## $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
##
## $temp
## [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"
```

```r
attributes(j) ##print the class of the medical record
```

```
## $class
## [1] "medicalRecord"
##
## $names
```

```
## [1] "name"              "gender"            "date_of_birth"
## [4] "date_of_admission" "pulse"             "temp"
## [7] "fluid"
```

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(p) {
  mean.pulse <- mean(p$pulse)
  mean.temp <- mean(p$temp)
  mean.fluid <- mean(p$fluid)
x<-c(mean.pulse, mean.temp,mean.fluid)
names(x)<-c("mean.pulse", "mean.temp","mean.fluid")
return(x)
}

##which employs some nice formatting, perhaps arranging measurements by date
print.medicalRecord <- function(p) {
  cat("name:",p$name,"\n")
  cat(sprintf("gender: %s\n",p$gender))
  cat(sprintf("date of birth: %s\n",p$date_of_birth))
  t<-as.data.frame(p[4:7])
  t <- t[order(t[,1]),]   ### sorting data by date
  return(t)
}

##generates a composite plot of measurements over time
plot.medicalRecord <- function(p) {
  par(mfrow=c(1,3))
  plot(p$pulse, ylab="pulse", xlab="visit")
  plot(p$temp, ylab="temparature", xlab="visit")
  plot(p$fluid,ylab="fluid", xlab="visit")
}
mean.medicalRecord(j)
```

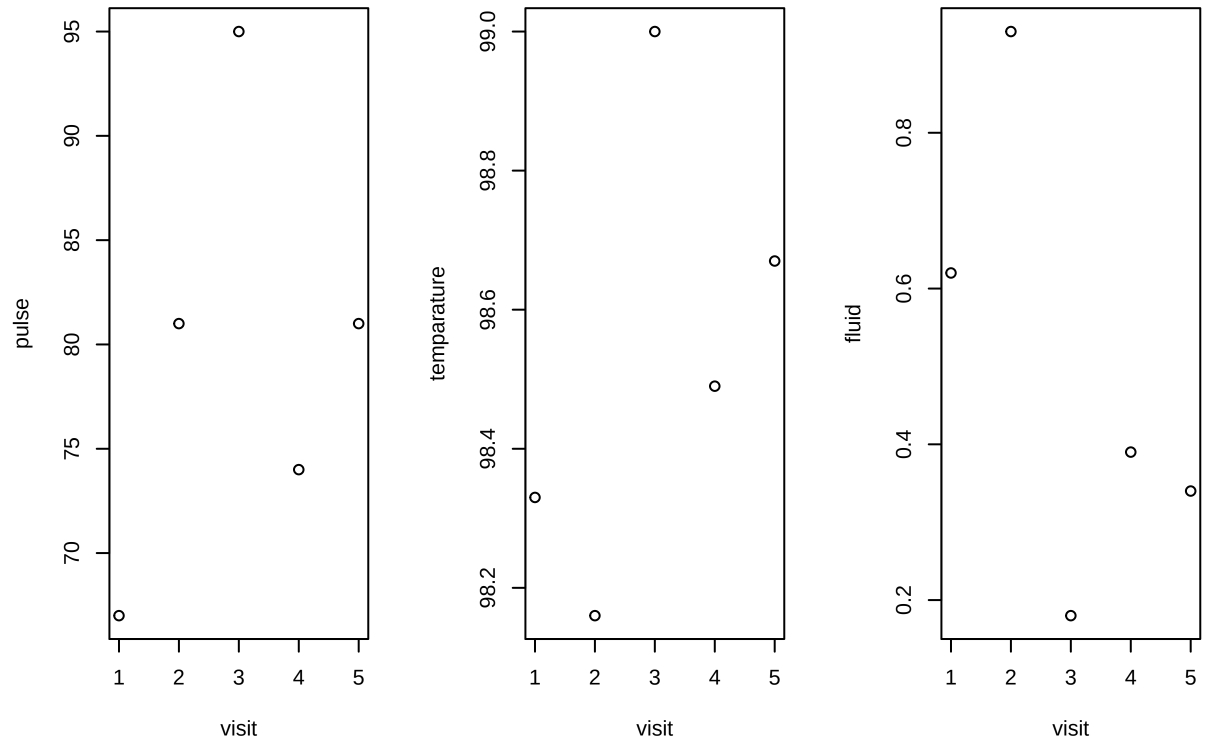
```
## mean.pulse  mean.temp mean.fluid
##     79.600     98.530      0.492
```

```r
print.medicalRecord(j)
```

```
## name: Mev
## gender: male
## date of birth: 1976-08-09

##   date_of_admission pulse  temp 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
```

```r
plot.medicalRecord(j)
```



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)

```r
###reset seed
set.seed(8)
## for loop, create 10 records
for(i in 1:10)
{
  name <- paste("c", i, sep = "")
  cohort<- makePatient()
  class(cohort)<-"medicalRecord"
  names(cohort)<-c("name","gender","date_of_birth","date_of_admission","pulse","temp","fluid")
  assign(name, cohort)
}
## show mean and print
for(i in 1:10){
  name <- paste("c", i, sep = "")
  cohort <- get(name)
  print(mean(cohort))
  print(print(cohort))
}
```

```
## mean.pulse  mean.temp mean.fluid
##     79.600     98.530      0.492
## name: Mev
## gender: male
## date of birth: 1976-08-09
##   date_of_admission pulse  temp 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
## mean.pulse  mean.temp mean.fluid
##     78.000     98.495      0.245
## name: Yul
## gender: male
## date of birth: 1988-06-28
##   date_of_admission pulse  temp fluid
## 1        2012-01-16    76 98.92  0.14
## 2        2013-08-07    80 98.07  0.35
## mean.pulse  mean.temp mean.fluid
## 81.5000000 98.4400000  0.4033333
## name: Zet
## gender: female
## date of birth: 1970-06-13
##   date_of_admission pulse  temp 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
## mean.pulse  mean.temp mean.fluid
##      78.00      98.60       0.65
## name: Qih
## gender: female
## date of birth: 1987-08-30
##   date_of_admission pulse temp fluid
## 1        2011-06-22    78 98.6  0.65
## mean.pulse  mean.temp mean.fluid
## 88.3333333 98.0500000  0.5866667
## name: Wut
## gender: male
## date of birth: 1974-06-28
##   date_of_admission pulse  temp 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
## mean.pulse  mean.temp mean.fluid
##    83.5000    98.4500     0.4525
## name: Juy
## gender: male
## date of birth: 1983-06-09
##   date_of_admission pulse  temp 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
## mean.pulse  mean.temp mean.fluid
##      83.00      98.01       0.97
## name: God
## gender: female
## date of birth: 1990-02-12
##   date_of_admission pulse  temp fluid
## 1        2010-03-12    83 98.01  0.97
## mean.pulse  mean.temp mean.fluid
## 77.5000000 98.1483333  0.3366667
## name: Fut
## gender: male
## date of birth: 1970-01-11
##   date_of_admission pulse  temp 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
## mean.pulse  mean.temp mean.fluid
##     77.000     98.830      0.445
## name: Pet
## gender: male
## date of birth: 1979-01-01
##   date_of_admission pulse  temp fluid
## 1        2010-10-30    85 98.84  0.60
## 2        2012-05-10    69 98.82  0.29
## mean.pulse  mean.temp mean.fluid
## 79.3333333 98.3000000  0.6583333
## name: Yed
## gender: male
## date of birth: 1977-11-11
##   date_of_admission pulse  temp 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
```