Bios 6301: Assignment 6

Valerie Welty

Due Thursday, 1 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:

```r
# edited some of the following code to set up the data for question 2
pop <- data.frame(m = rnorm(100, 160, 20), f = rnorm(100, 160, 20))
head(pop)
```

```
##          m        f
## 1 138.4887 155.1750
## 2 172.0815 137.4757
## 3 183.1020 145.7720
## 4 146.0997 160.0064
## 5 166.1805 171.3179
## 6 162.0637 173.9365
```

```r
mean(pop$m)
```

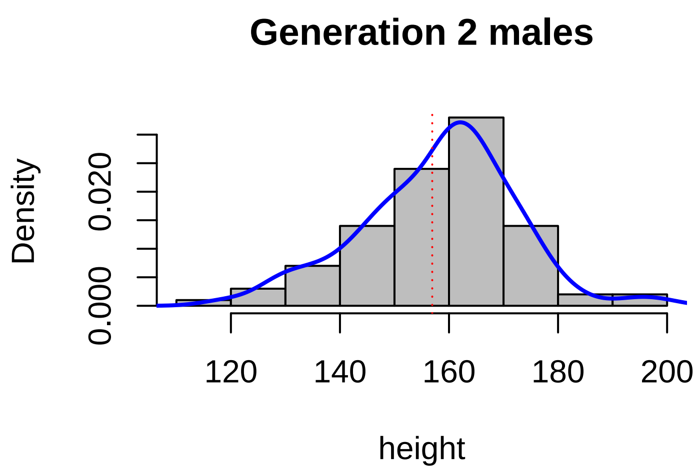
```
## [1] 156.9241
```

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.

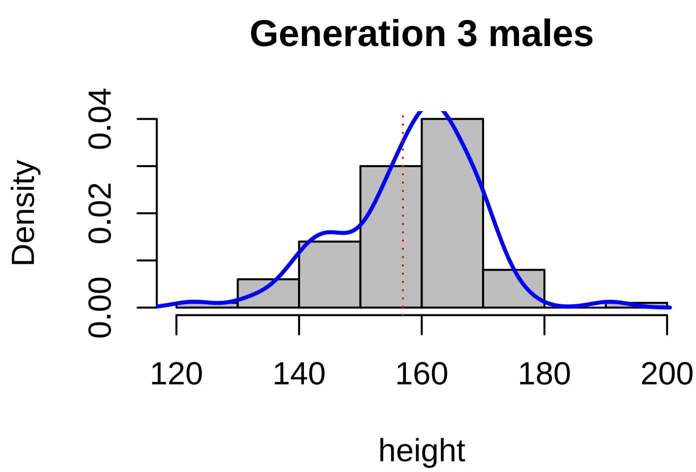
```
next_gen <- function(pop) {

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

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.

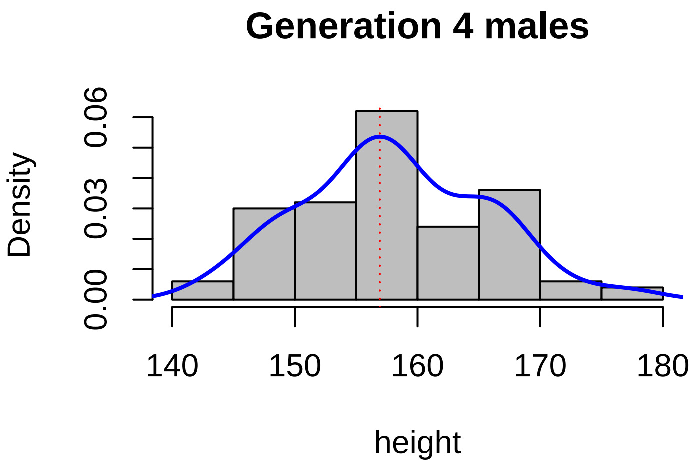
```
pop2 <- next_gen(pop)
hist(pop2$m, prob=TRUE, main='Generation 2 males', xlab='height', col='grey')
lines(density(pop2$m), col='blue', lwd=2)
abline(v=mean(pop$m),col='red',lty='dotted')
```



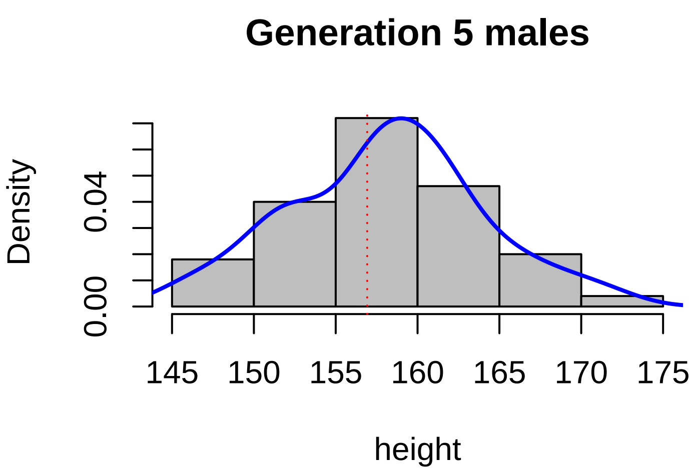

```
pop3 <- next_gen(pop2)
hist(pop3$m, prob=TRUE, main='Generation 3 males', xlab='height', col='grey')
lines(density(pop3$m), col='blue', lwd=2)
abline(v=mean(pop$m),col='red',lty='dotted')
```



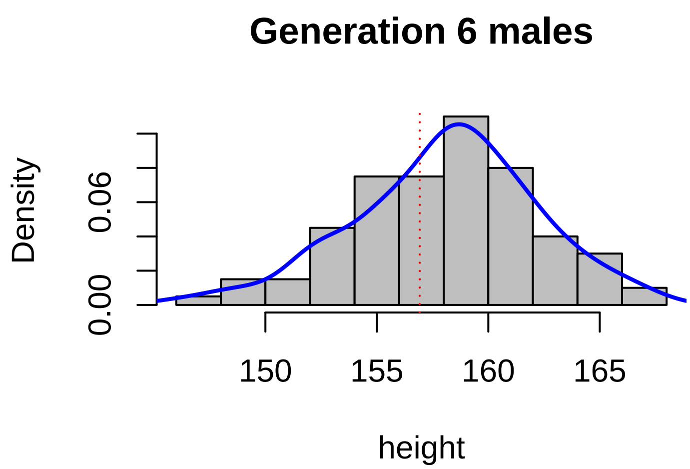
```
pop4 <- next_gen(pop3)
hist(pop4$m, prob=TRUE, main='Generation 4 males', xlab='height', col='grey')
lines(density(pop4$m), col='blue', lwd=2)
abline(v=mean(pop$m),col='red',lty='dotted')
```



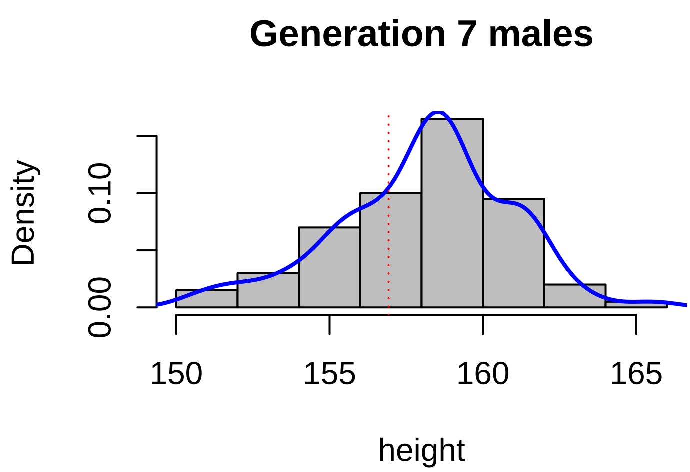

```
pop5 <- next_gen(pop4)
hist(pop5$m, prob=TRUE, main='Generation 5 males', xlab='height', col='grey')
lines(density(pop5$m), col='blue', lwd=2)
abline(v=mean(pop$m),col='red',lty='dotted')
```



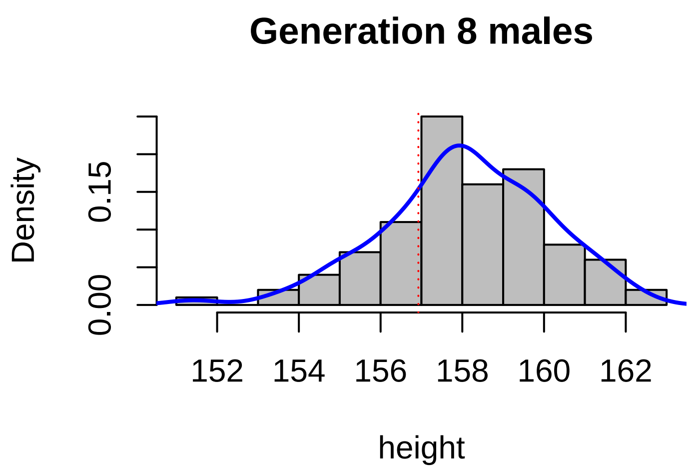

```
pop6 <- next_gen(pop5)
hist(pop6$m, prob=TRUE, main='Generation 6 males', xlab='height', col='grey')
lines(density(pop6$m), col='blue', lwd=2)
abline(v=mean(pop$m),col='red',lty='dotted')
```



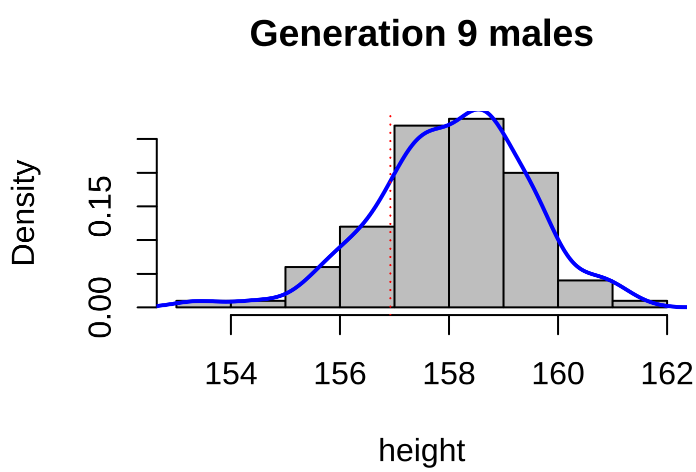

```
pop7 <- next_gen(pop6)
hist(pop7$m, prob=TRUE, main='Generation 7 males', xlab='height', col='grey')
lines(density(pop7$m), col='blue', lwd=2)
abline(v=mean(pop$m),col='red',lty='dotted')
```




```
pop8 <- next_gen(pop7)
hist(pop8$m, prob=TRUE, main='Generation 8 males', xlab='height', col='grey')
lines(density(pop8$m), col='blue', lwd=2)
abline(v=mean(pop$m),col='red',lty='dotted')
```




```r
pop9 <- next_gen(pop8)
hist(pop9$m, prob=TRUE, main='Generation 9 males', xlab='height', col='grey')
lines(density(pop9$m), col='blue', lwd=2)
abline(v=mean(pop$m),col='red',lty='dotted')
```



Question 2

10 points

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

```r
library(ggplot2)
pop$gen <- 1
pop2$gen <- 2
pop3$gen <- 3
pop4$gen <- 4
pop5$gen <- 5
pop6$gen <- 6
pop7$gen <- 7
pop8$gen <- 8
pop9$gen <- 9
all_pop <- rbind(pop, pop2, pop3, pop4, pop5, pop6, pop7, pop8, pop9)
```

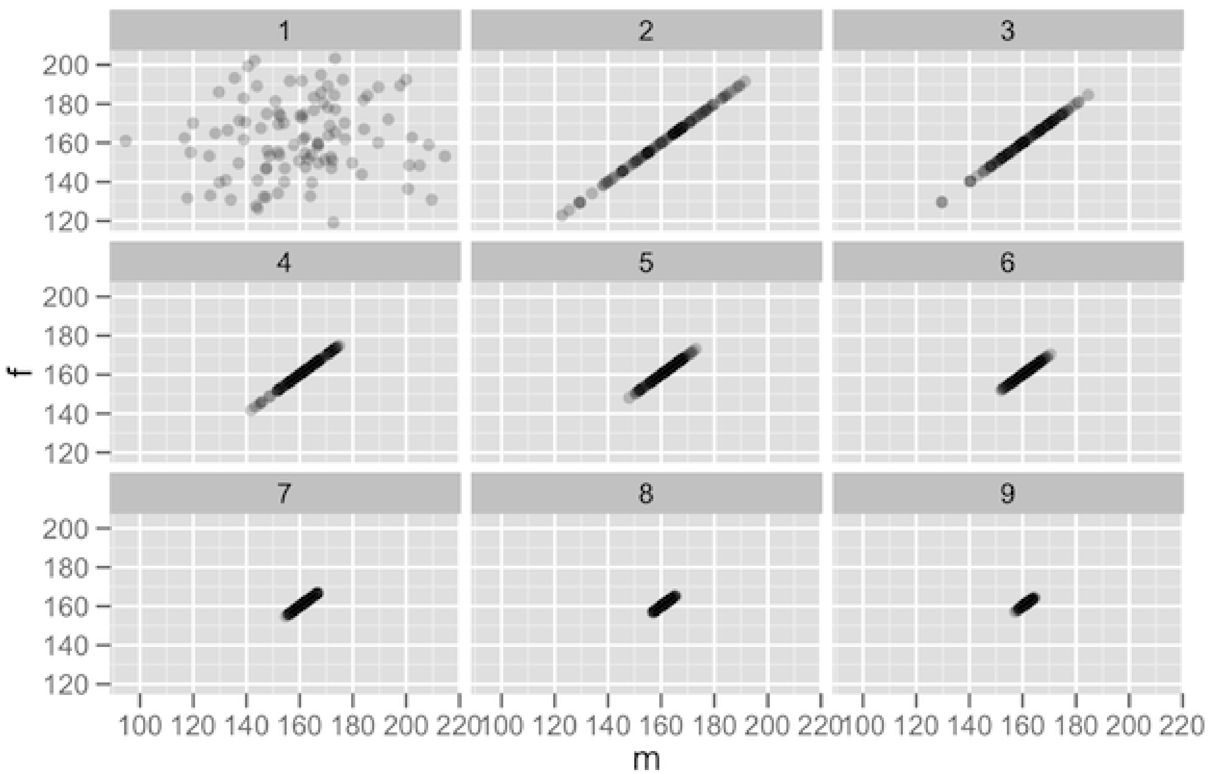
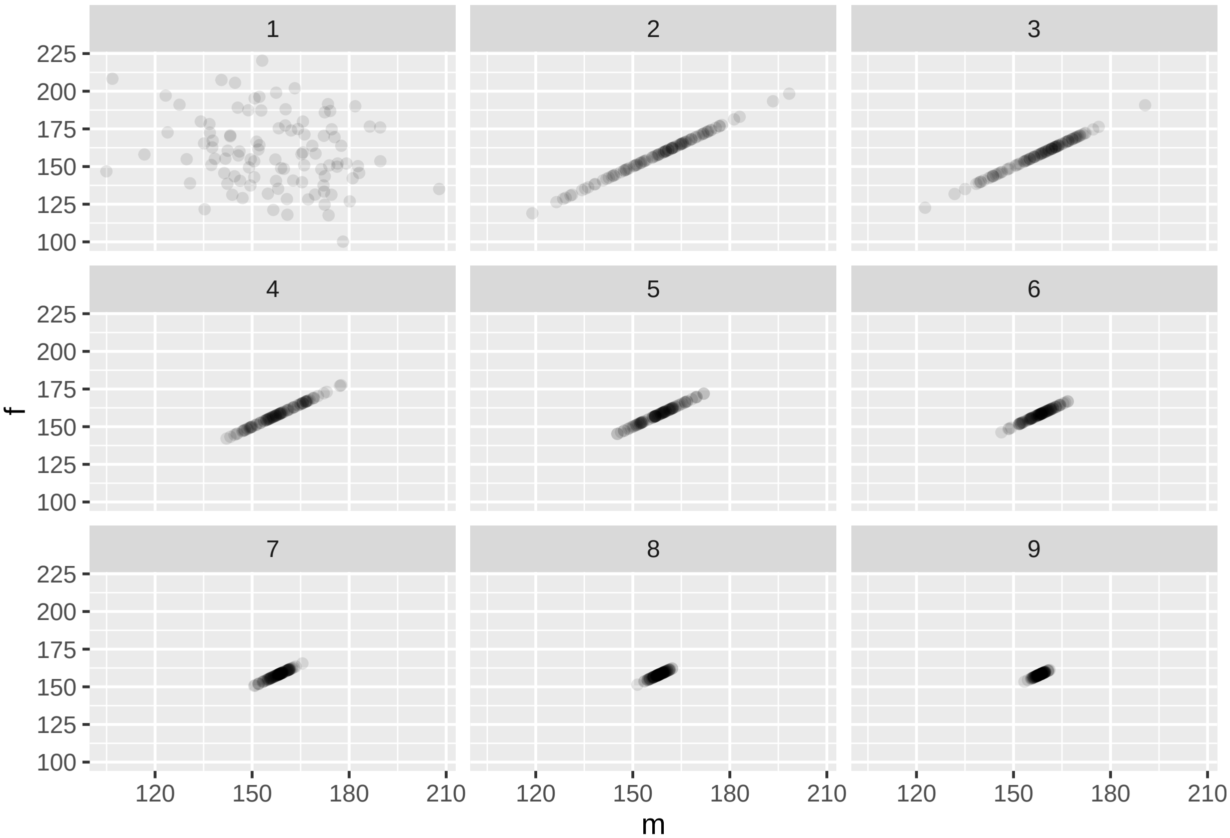


Figure 1: generations plot

```r
ggplot(data=all_pop)+ geom_point(mapping=aes(x=m, y=f), alpha = 1/10) + 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)

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).

```r
alpha <- 0.05

x <- as.data.frame(matrix(data = NA, nrow = 250, ncol = 2))
head <- c("treatment", "outcome")
names(x) = head
treat <- c(0, 1)
```

```r
x[, 1] = sample(treat, 250, replace = TRUE)
for (i in 1:250) {
    if (x[i, 1] == 0) {
        x[i, 2] = rnorm(1, 60, 20)
    } else if (x[i, 1] == 1) {
        x[i, 2] = rnorm(1, 60, 20) + 5
    }
}

aggregate(x[, 2], list(x$treatment), mean)
```

```
##   Group.1        x
## 1       0 61.25067
## 2       1 66.78229
```

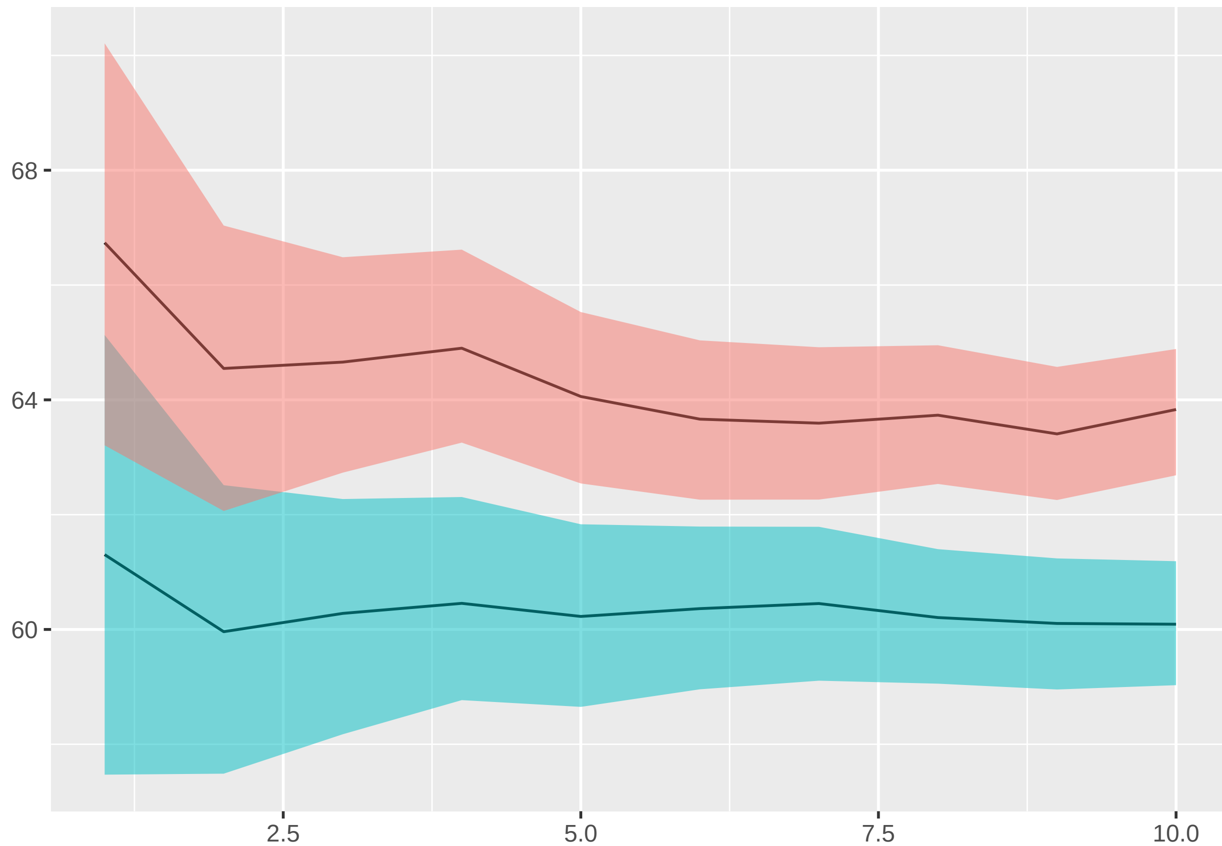
```r
library(dplyr)
results <- as.data.frame(matrix(data = NA, nrow = 20, ncol = 5))
head <- c("interval", "treat", "mean", "lb", "ub")
names(results) = head
results[, 1] <- rep(1:10, each = 2)


### bootstrapping
bootstrap <- function(df, B = 1000) {
    sample.n <- nrow(df)
    boot_means <- matrix(data = NA, nrow = B, ncol = 2)
    for (i in 1:B) {
        boot <- sample_n(df, sample.n, replace = TRUE)
        boot_means[i, ] <- as.numeric(unlist(aggregate(boot[, 2], list(boot[,
            1]), mean)[2]))
    }
    return(c(0, mean(boot_means[, 1]), quantile(boot_means[, 1], c(0.025, 0.975)),
        1, mean(boot_means[, 2]), quantile(boot_means[, 2], c(0.025, 0.975))))
}

l <- bootstrap(x)
results[1, 2:5] <- l[1:4]
results[2, 2:5] <- l[5:8]

# increase data set to 500

x[251:500, 1] = sample(treat, 250, replace = TRUE)
for (i in 251:500) {
    if (x[i, 1] == 0) {
        x[i, 2] = rnorm(1, 60, 20)
    } else if (x[i, 1] == 1) {
        x[i, 2] = rnorm(1, 60, 20) + 5
    }
}

l <- bootstrap(x)
results[3, 2:5] <- l[1:4]
```

```r
results[4, 2:5] <- l[5:8]

# increase data set to 750

x[501:750, 1] = sample(treat, 250, replace = TRUE)
for (i in 501:750) {
    if (x[i, 1] == 0) {
        x[i, 2] = rnorm(1, 60, 20)
    } else if (x[i, 1] == 1) {
        x[i, 2] = rnorm(1, 60, 20) + 5
    }
}

l <- bootstrap(x)
results[5, 2:5] <- l[1:4]
results[6, 2:5] <- l[5:8]

# increase data set to 1000

x[751:1000, 1] = sample(treat, 250, replace = TRUE)
for (i in 751:1000) {
    if (x[i, 1] == 0) {
        x[i, 2] = rnorm(1, 60, 20)
    } else if (x[i, 1] == 1) {
        x[i, 2] = rnorm(1, 60, 20) + 5
    }
}

l <- bootstrap(x)
results[7, 2:5] <- l[1:4]
results[8, 2:5] <- l[5:8]


# increase data set to 1250

x[1001:1250, 1] = sample(treat, 250, replace = TRUE)
for (i in 1001:1250) {
    if (x[i, 1] == 0) {
        x[i, 2] = rnorm(1, 60, 20)
    } else if (x[i, 1] == 1) {
        x[i, 2] = rnorm(1, 60, 20) + 5
    }
}

l <- bootstrap(x)
results[9, 2:5] <- l[1:4]
results[10, 2:5] <- l[5:8]


# increase data set to 1500

x[1251:1500, 1] = sample(treat, 250, replace = TRUE)
for (i in 1251:1500) {
```

```
    if (x[i, 1] == 0) {
        x[i, 2] = rnorm(1, 60, 20)
    } else if (x[i, 1] == 1) {
        x[i, 2] = rnorm(1, 60, 20) + 5
    }
}

l <- bootstrap(x)
results[11, 2:5] <- l[1:4]
results[12, 2:5] <- l[5:8]


# increase data set to 1750

x[1501:1750, 1] = sample(treat, 250, replace = TRUE)
for (i in 1501:1750) {
    if (x[i, 1] == 0) {
        x[i, 2] = rnorm(1, 60, 20)
    } else if (x[i, 1] == 1) {
        x[i, 2] = rnorm(1, 60, 20) + 5
    }
}

l <- bootstrap(x)
results[13, 2:5] <- l[1:4]
results[14, 2:5] <- l[5:8]

# increase data set to 2000

x[1751:2000, 1] = sample(treat, 250, replace = TRUE)
for (i in 1751:2000) {
    if (x[i, 1] == 0) {
        x[i, 2] = rnorm(1, 60, 20)
    } else if (x[i, 1] == 1) {
        x[i, 2] = rnorm(1, 60, 20) + 5
    }
}

l <- bootstrap(x)
results[15, 2:5] <- l[1:4]
results[16, 2:5] <- l[5:8]

# increase data set to 2250

x[2001:2250, 1] = sample(treat, 250, replace = TRUE)
for (i in 2001:2250) {
    if (x[i, 1] == 0) {
        x[i, 2] = rnorm(1, 60, 20)
    } else if (x[i, 1] == 1) {
        x[i, 2] = rnorm(1, 60, 20) + 5
    }
}
```

```r
l <- bootstrap(x)
results[17, 2:5] <- l[1:4]
results[18, 2:5] <- l[5:8]

# increase data set to 2500

x[2251:2500, 1] = sample(treat, 250, replace = TRUE)
for (i in 2251:2500) {
    if (x[i, 1] == 0) {
        x[i, 2] = rnorm(1, 60, 20)
    } else if (x[i, 1] == 1) {
        x[i, 2] = rnorm(1, 60, 20) + 5
    }
}

l <- bootstrap(x)
results[19, 2:5] <- l[1:4]
results[20, 2:5] <- l[5:8]
```

```r
# plot the graph
library(ggplot2)
results0 <- results[which(results[, 2] == 0), ]
results1 <- results[which(results[, 2] == 1), ]

p1 <- ggplot(results, aes(x = interval, y = mean, group = treat)) + geom_line() +
    theme(axis.title.x = element_blank(), axis.title.y = element_blank()) +
    xlim(1, 10)
# + scale_x_discrete(limits=seq(1:10))
p2 <- p1 + geom_ribbon(data = results0, aes(ymin = lb, ymax = ub, fill = "red"),
    alpha = 0.5) + guides(fill = FALSE)
p3 <- p2 + geom_ribbon(data = results1, aes(ymin = lb, ymax = ub, fill = "blue"),
    alpha = 0.5) + guides(fill = FALSE)
p3
```



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)

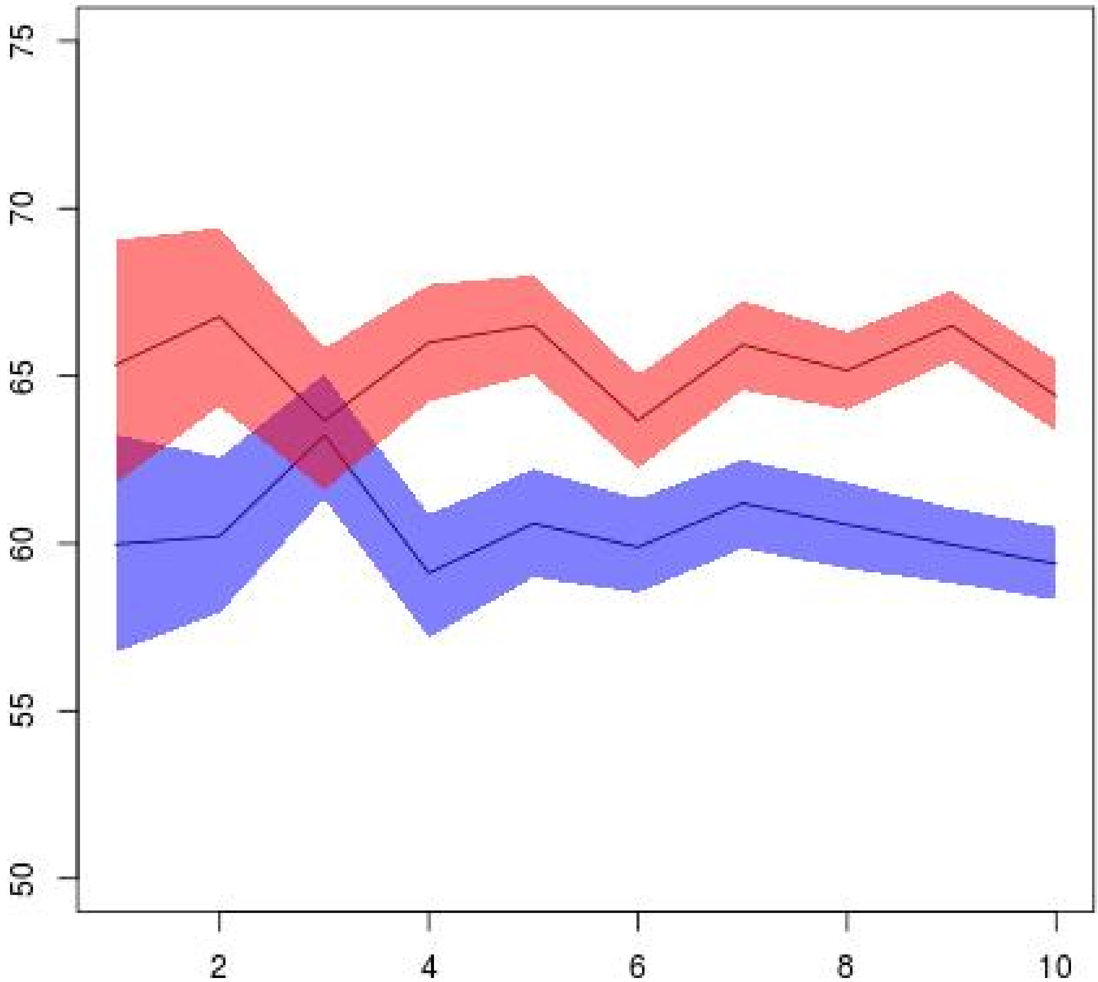


Figure 2: bp interval plot

```r
set.seed(8)
pat1 <- makePatient()
names(pat1) <- c('name','gender','dob','doa','pulse','temp','fluid')
class(pat1) <- 'medicalRecord'
pat1
```

```
## $name
## [1] "Mev"
##
## $gender
## [1] male
## Levels: female male
##
## $dob
## [1] "1976-08-09"
##
## $doa
## [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
class(pat1)
```

```
## [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(obj) {
    means <- as.data.frame(matrix(data = c(mean(obj$pulse), mean(obj$temp),
        mean(obj$fluid)), ncol = 3, nrow = 1))
    names(means) <- c("pulse", "temp", "fluid")
    row.names(means) <- obj$name
    means
    return(means)
}

print.medicalRecord <- function(obj) {
    cat(obj$name, "\n")
    cat("gender:", paste(obj$gender), "\n")
```

```r
    cat(paste("dob:", obj$dob, "\n"))

    # last 4 items are 4 variables that are attached to the visit date, group
    # together
    x <- as.data.frame(matrix(data = NA, nrow = length(obj$doa), ncol = 4))
    x[, 1] <- obj$doa
    x[, 2] <- obj$pulse
    x[, 3] <- obj$temp
    x[, 4] <- obj$fluid
    x
    x <- x[order(x[, 1]), ]
    names(x) = c("date of admission", "pulse", "temperature", "fluid intake")
    print(x, row.names = FALSE)
}


plot.medicalRecord <- function(obj) {
    x <- as.data.frame(matrix(data = NA, nrow = length(obj$doa), ncol = 4))
    x[, 1] <- obj$doa
    x[, 2] <- obj$pulse
    x[, 3] <- obj$temp
    x[, 4] <- obj$fluid
    head <- c("doa", "pulse", "temp", "fluid")
    names(x) = head
    x
    p1 <- ggplot(x, aes(x = doa, y = pulse)) + geom_line()
    p2 <- ggplot(x, aes(x = doa, y = temp)) + geom_line()
    p3 <- ggplot(x, aes(x = doa, y = fluid)) + geom_line()
    library(gridExtra)
    grid.arrange(p1, p2, p3, nrow = 3)
}


mean(pat1)
```

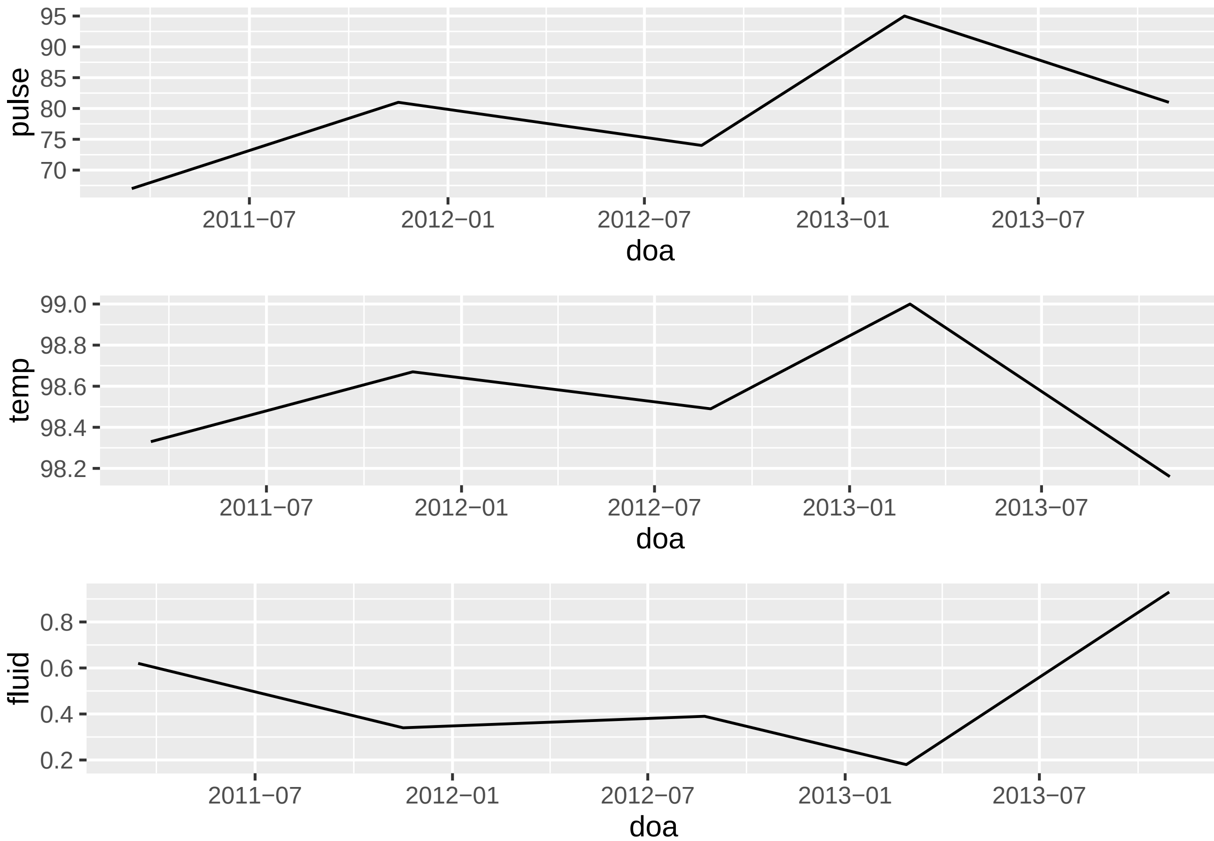
```
##     pulse  temp fluid
## Mev  79.6 98.53 0.492
```

```r
print(pat1)
```

```
## Mev
## gender: male
## dob: 1976-08-09
##  date of admission pulse temperature fluid intake
##         2011-03-14    67       98.33         0.62
##         2011-11-16    81       98.67         0.34
##         2012-08-23    74       98.49         0.39
##         2013-02-27    95       99.00         0.18
##         2013-10-30    81       98.16         0.93
```

```r
plot(pat1)
```

```
## Warning: package 'gridExtra' was built under R version 3.2.4
```



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)

I present two solutions below.

Solution 1: Turn cohort into a group of lists of class 'medicalRecords' and utilize functions from part (2) of this question.

```
set.seed(8)
mean.cohortRecords <- function(obj) {
  means <- as.data.frame(matrix(data=NA, nrow=ncol(obj), ncol=3))
  names(means) <- c('pulse','temp','fluid')
  row.names(means) <- obj[1,]
  for(i in 1:ncol(obj)) {
    pt <- obj[,i]
    names(pt) <- c('name','gender','dob','doa','pulse','temp','fluid')
    class(pt) = 'medicalRecord'
    means[i,] = mean(pt)
  }
  return(means)
}
```

```
print.cohortRecords <- function(obj) {
  for(i in 1:ncol(obj)) {
    pt <- obj[,i]
    names(pt) <- c('name','gender','dob','doa','pulse','temp','fluid')
    class(pt) = 'medicalRecord'
    print(pt)
  }
}


set.seed(8)
cohort <- replicate(10, makePatient())
class(cohort) <- 'cohortRecords'
# cohort


mean(cohort)
```

```
##        pulse     temp     fluid
## Mev 79.60000 98.53000 0.4920000
## Yul 78.00000 98.49500 0.2450000
## Zet 81.50000 98.44000 0.4033333
## Qih 78.00000 98.60000 0.6500000
## Wut 88.33333 98.05000 0.5866667
## Juy 83.50000 98.45000 0.4525000
## God 83.00000 98.01000 0.9700000
## Fut 77.50000 98.14833 0.3366667
## Pet 77.00000 98.83000 0.4450000
## Yed 79.33333 98.30000 0.6583333
```

```
print(cohort)
```

```
## Mev
## gender: male
## dob: 1976-08-09
##  date of admission pulse temperature fluid intake
##         2011-03-14    67       98.33         0.62
##         2011-11-16    81       98.67         0.34
##         2012-08-23    74       98.49         0.39
##         2013-02-27    95       99.00         0.18
##         2013-10-30    81       98.16         0.93
## Yul
## gender: male
## dob: 1988-06-28
##  date of admission pulse temperature fluid intake
##         2012-01-16    76       98.92         0.14
##         2013-08-07    80       98.07         0.35
## Zet
## gender: female
## dob: 1970-06-13
##  date of admission pulse temperature fluid intake
##         2010-03-21    79       98.58         0.22
```

```
##         2010-04-01    73       98.32         0.61
##         2012-08-29    88       98.47         0.59
##         2013-06-01    84       98.22         0.25
##         2013-11-03    72       98.54         0.03
##         2014-02-05    93       98.51         0.72
## Qih
## gender: female
## dob: 1987-08-30
##  date of admission pulse temperature fluid intake
##         2011-06-22    78        98.6         0.65
## Wut
## gender: male
## dob: 1974-06-28
##  date of admission pulse temperature fluid intake
##         2010-04-12    76       98.05         0.65
##         2011-02-16    93       98.26         0.97
##         2012-04-12    96       97.84         0.14
## Juy
## gender: male
## dob: 1983-06-09
##  date of admission pulse temperature fluid intake
##         2010-03-10    81       99.11         0.66
##         2010-03-25    90       98.58         0.26
##         2010-04-18    75       98.58         0.60
##         2010-06-10    88       97.53         0.29
## God
## gender: female
## dob: 1990-02-12
##  date of admission pulse temperature fluid intake
##         2010-03-12    83       98.01         0.97
## Fut
## gender: male
## dob: 1970-01-11
##  date of admission pulse temperature fluid intake
##         2011-04-07    80       97.87         0.36
##         2011-04-14    83       97.91         0.00
##         2011-08-16    66       98.49         0.13
##         2013-03-15    74       98.38         0.31
##         2013-06-20    74       98.41         0.49
##         2013-11-12    88       97.83         0.73
## Pet
## gender: male
## dob: 1979-01-01
##  date of admission pulse temperature fluid intake
##         2010-10-30    85       98.84         0.60
##         2012-05-10    69       98.82         0.29
## Yed
## gender: male
## dob: 1977-11-11
##  date of admission pulse temperature fluid intake
##         2010-01-28    63       97.95         0.94
##         2010-03-06    81       98.45         0.67
##         2010-07-10    98       98.65         0.79
##         2010-08-27    66       97.68         0.36
```

```
##         2011-06-18    83       98.00          0.69
##         2013-01-06    85       99.07          0.50
```

Solution 2: return matrix of means without using additional methods

```
mean.cohortRecords <- function(obj) {
  mean.pulse <- numeric(ncol(obj))
  mean.temp <- numeric(ncol(obj))
  mean.fluid <- numeric(ncol(obj))
  for(i in 1:ncol(obj)) {
    mean.pulse[i] = mean(unlist(obj[5,i]))
    mean.temp[i] = mean(unlist(obj[6,i]))
    mean.fluid[i] = mean(unlist(obj[7,i]))
  }
  mean.pulse
  mean.temp
  mean.fluid

  means <- as.data.frame(matrix(data=c(mean.pulse,mean.temp,mean.fluid), nrow=ncol(obj), ncol=3,
                                byrow=FALSE))
  row.names(means) = unlist(obj[1,])
  names(means) = c('pulse','temp','fluid')
  return(means)
}


# creating a method print without using print.medicalRecord seems redundant and unnecessary


set.seed(8)
cohort <- replicate(10, makePatient())
class(cohort) <- 'cohortRecords'


mean(cohort)
```

```
##        pulse     temp     fluid
## Mev 79.60000 98.53000 0.4920000
## Yul 78.00000 98.49500 0.2450000
## Zet 81.50000 98.44000 0.4033333
## Qih 78.00000 98.60000 0.6500000
## Wut 88.33333 98.05000 0.5866667
## Juy 83.50000 98.45000 0.4525000
## God 83.00000 98.01000 0.9700000
## Fut 77.50000 98.14833 0.3366667
## Pet 77.00000 98.83000 0.4450000
## Yed 79.33333 98.30000 0.6583333
```

```
print(cohort)
```

```
## Mev
## gender: male
## dob: 1976-08-09
```

```
##  date of admission pulse temperature fluid intake
##         2011-03-14    67       98.33         0.62
##         2011-11-16    81       98.67         0.34
##         2012-08-23    74       98.49         0.39
##         2013-02-27    95       99.00         0.18
##         2013-10-30    81       98.16         0.93
## Yul
## gender: male
## dob: 1988-06-28
##  date of admission pulse temperature fluid intake
##         2012-01-16    76       98.92         0.14
##         2013-08-07    80       98.07         0.35
## Zet
## gender: female
## dob: 1970-06-13
##  date of admission pulse temperature fluid intake
##         2010-03-21    79       98.58         0.22
##         2010-04-01    73       98.32         0.61
##         2012-08-29    88       98.47         0.59
##         2013-06-01    84       98.22         0.25
##         2013-11-03    72       98.54         0.03
##         2014-02-05    93       98.51         0.72
## Qih
## gender: female
## dob: 1987-08-30
##  date of admission pulse temperature fluid intake
##         2011-06-22    78        98.6         0.65
## Wut
## gender: male
## dob: 1974-06-28
##  date of admission pulse temperature fluid intake
##         2010-04-12    76       98.05         0.65
##         2011-02-16    93       98.26         0.97
##         2012-04-12    96       97.84         0.14
## Juy
## gender: male
## dob: 1983-06-09
##  date of admission pulse temperature fluid intake
##         2010-03-10    81       99.11         0.66
##         2010-03-25    90       98.58         0.26
##         2010-04-18    75       98.58         0.60
##         2010-06-10    88       97.53         0.29
## God
## gender: female
## dob: 1990-02-12
##  date of admission pulse temperature fluid intake
##         2010-03-12    83       98.01         0.97
## Fut
## gender: male
## dob: 1970-01-11
##  date of admission pulse temperature fluid intake
##         2011-04-07    80       97.87         0.36
##         2011-04-14    83       97.91         0.00
##         2011-08-16    66       98.49         0.13
```

```
##         2013-03-15    74       98.38         0.31
##         2013-06-20    74       98.41         0.49
##         2013-11-12    88       97.83         0.73
## Pet
## gender: male
## dob: 1979-01-01
##  date of admission pulse temperature fluid intake
##         2010-10-30    85       98.84         0.60
##         2012-05-10    69       98.82         0.29
## Yed
## gender: male
## dob: 1977-11-11
##  date of admission pulse temperature fluid intake
##         2010-01-28    63       97.95         0.94
##         2010-03-06    81       98.45         0.67
##         2010-07-10    98       98.65         0.79
##         2010-08-27    66       97.68         0.36
##         2011-06-18    83       98.00         0.69
##         2013-01-06    85       99.07         0.50
```