

# Assignment 6

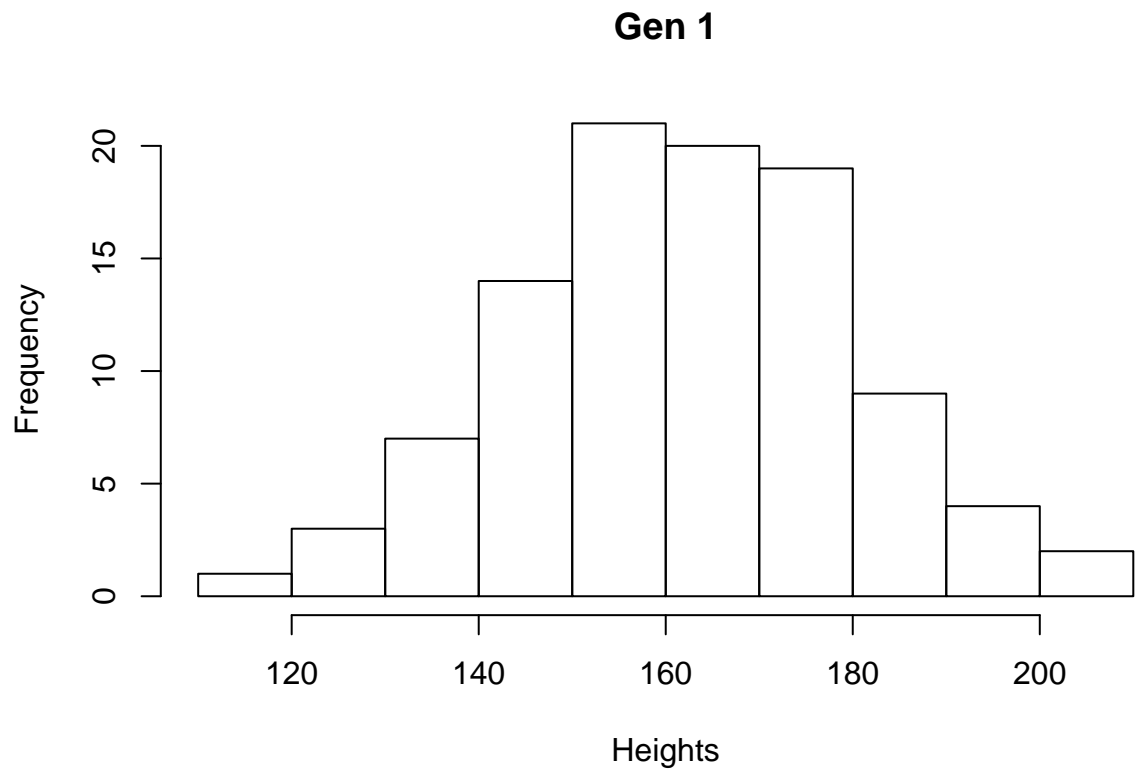
*James Lopez*

*12/1/2016*

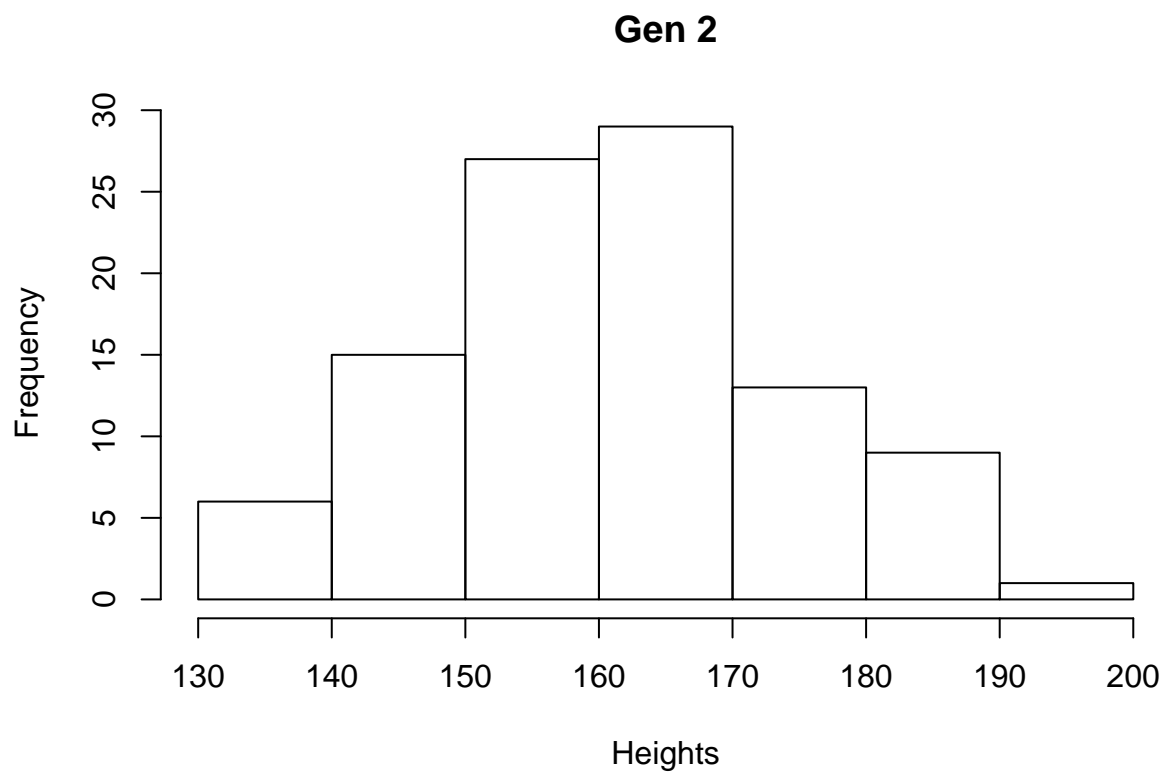
**Grade: 42/50** Nice job on defining classes. Please see comments regarding the bootstrap problem. Also, I'd be help you re-write the bootstrap problem so that it runs in seconds. Just let me know.

## Question 1

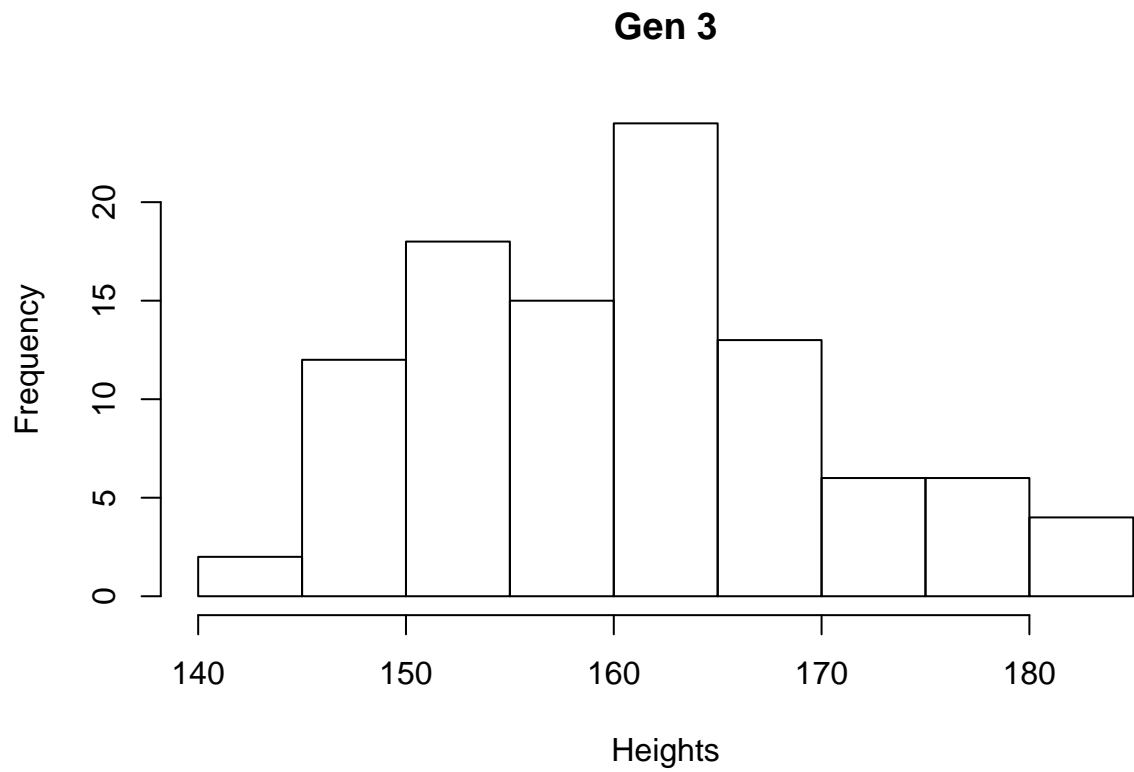
```
set.seed(1)
pop <- data.frame(m = rnorm(100, 160, 20), f = rnorm(100, 160, 20))
next_gen <- function(pop) {
  pop$m <- sample(pop$m)
  pop$m <- rowMeans(pop)
  pop$f <- pop$m
  pop
}
next_gen2 <- next_gen(pop)
next_gen3 <- next_gen(next_gen2)
next_gen4 <- next_gen(next_gen3)
next_gen5 <- next_gen(next_gen4)
next_gen6 <- next_gen(next_gen5)
next_gen7 <- next_gen(next_gen6)
next_gen8 <- next_gen(next_gen7)
next_gen9 <- next_gen(next_gen8)
next_gen10 <- next_gen(next_gen9)
hist(pop$m, main='Gen 1', xlab = 'Heights')
```



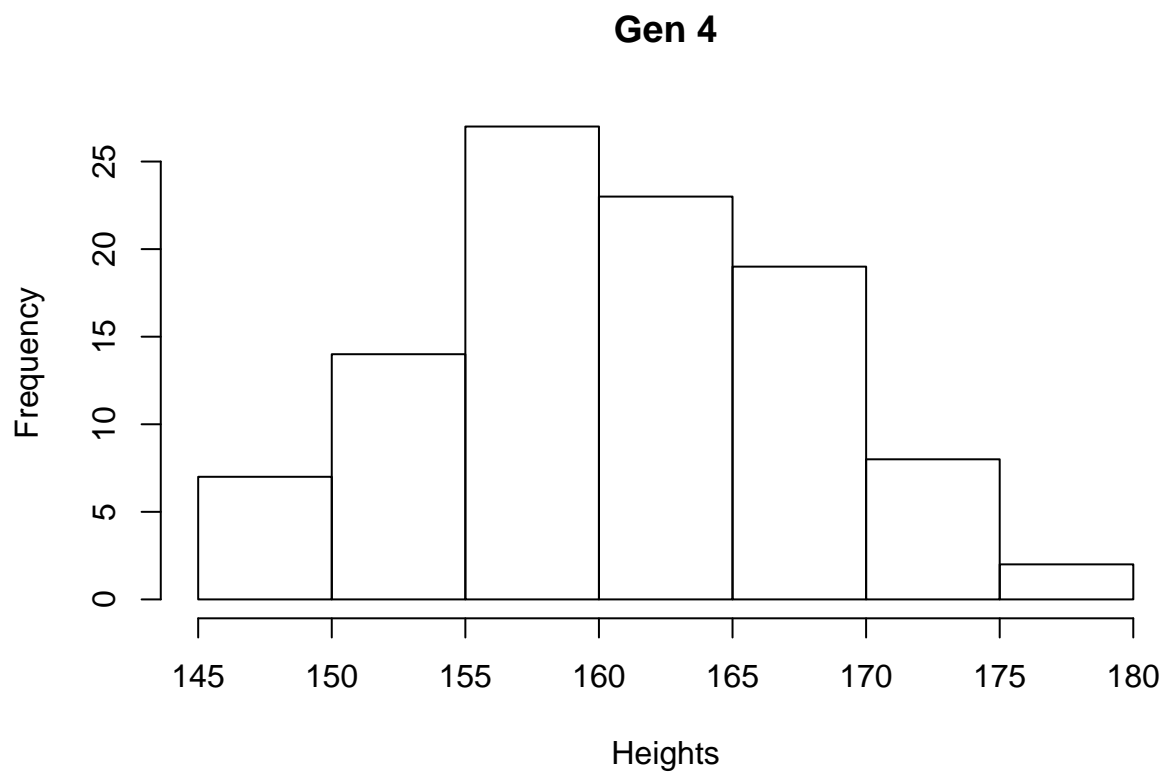
```
hist(next_gen2$m, main = 'Gen 2',xlab = 'Heights')
```



```
hist(next_gen3$m, main = 'Gen 3',xlab = 'Heights')
```

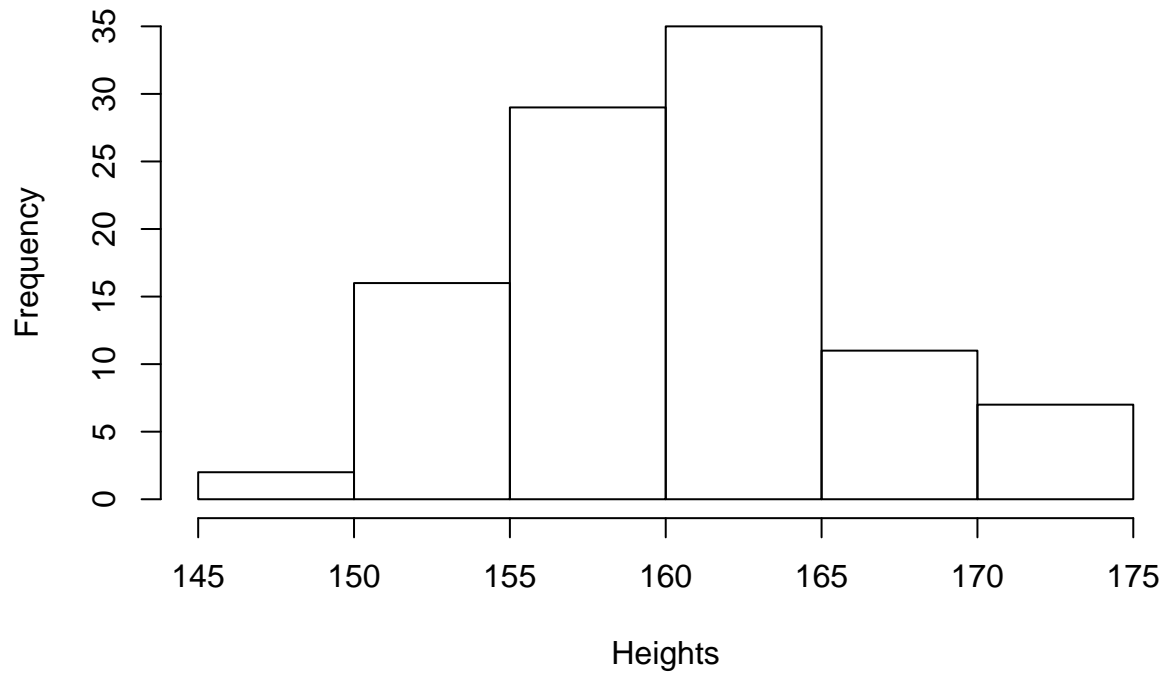


```
hist(next_gen4$m, main = 'Gen 4',xlab = 'Heights')
```



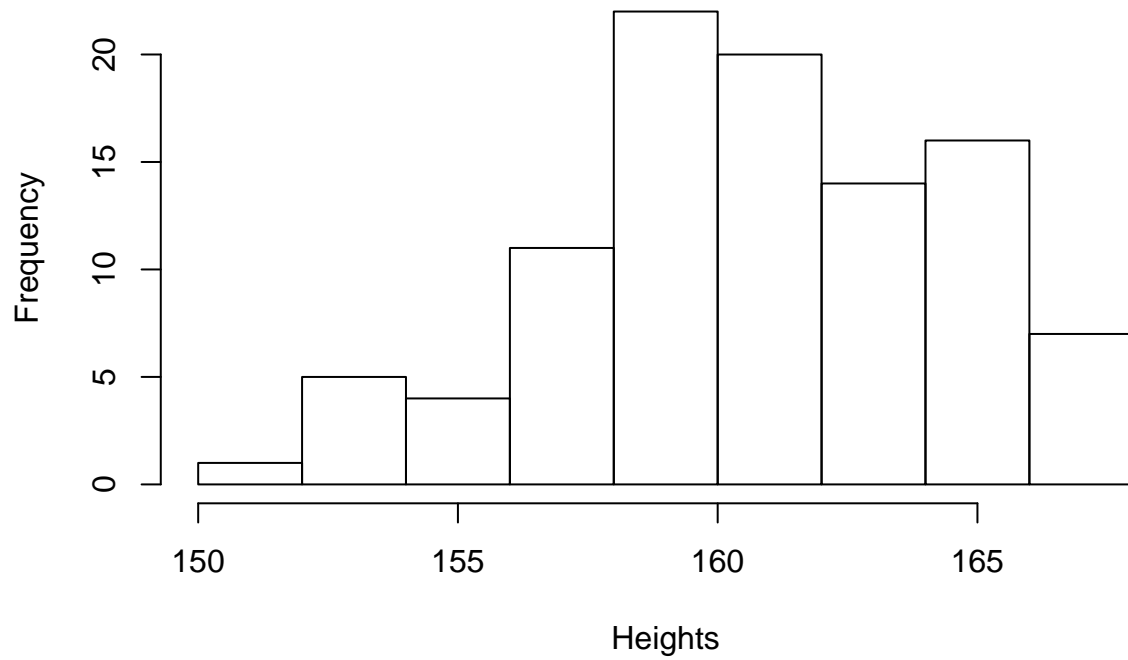
```
hist(next_gen5$m, main = 'Gen 5',xlab = 'Heights')
```

**Gen 5**



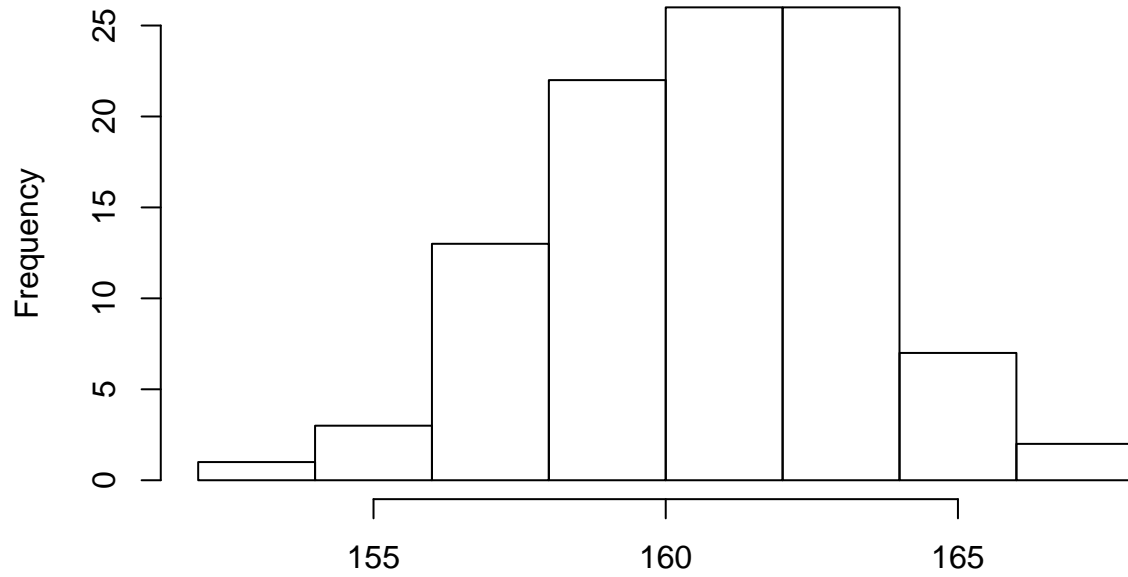
```
hist(next_gen6$m, main = 'Gen 6',xlab = 'Heights')
```

**Gen 6**



```
hist(next_gen7$m, main = 'Gen 7',xlab = 'Heights')
```

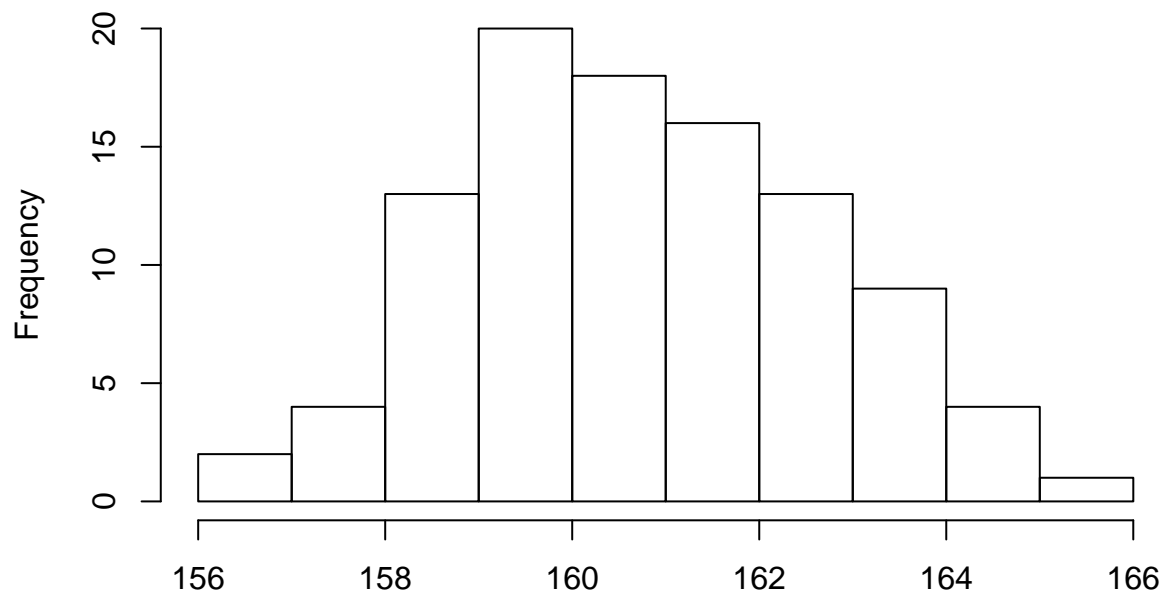
### Gen 7



### Heights

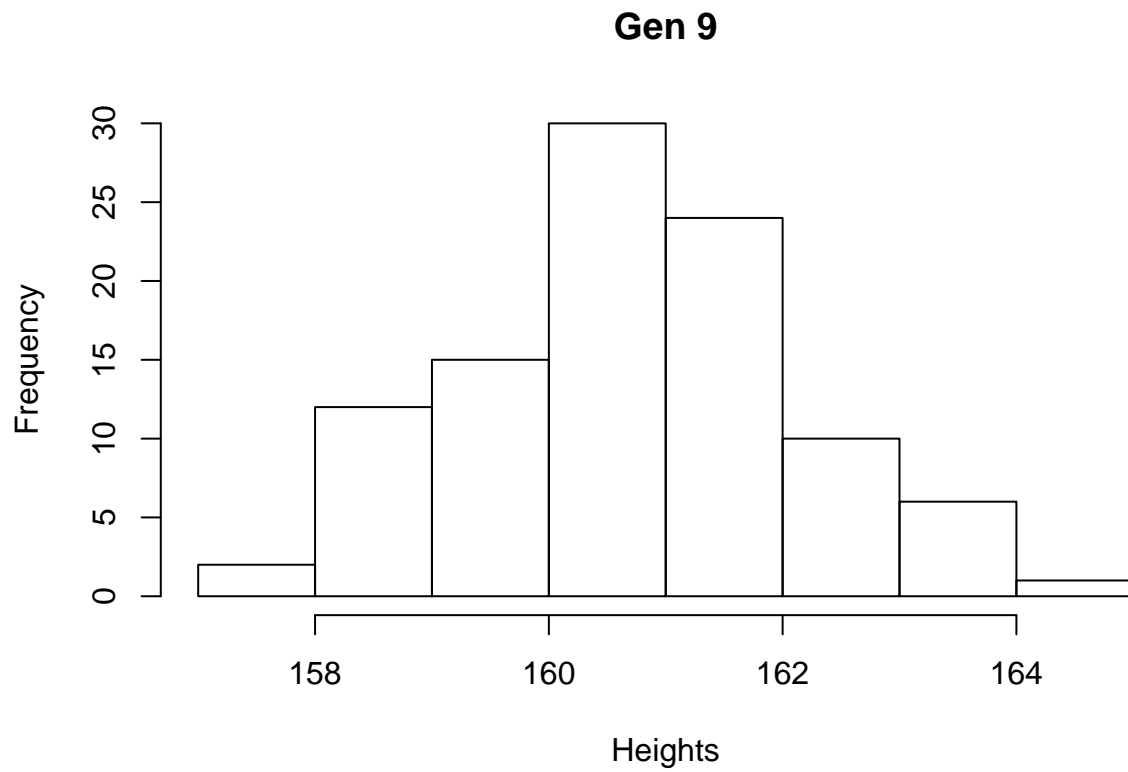
```
hist(next_gen8$m, main = 'Gen 8',xlab = 'Heights')
```

### Gen 8

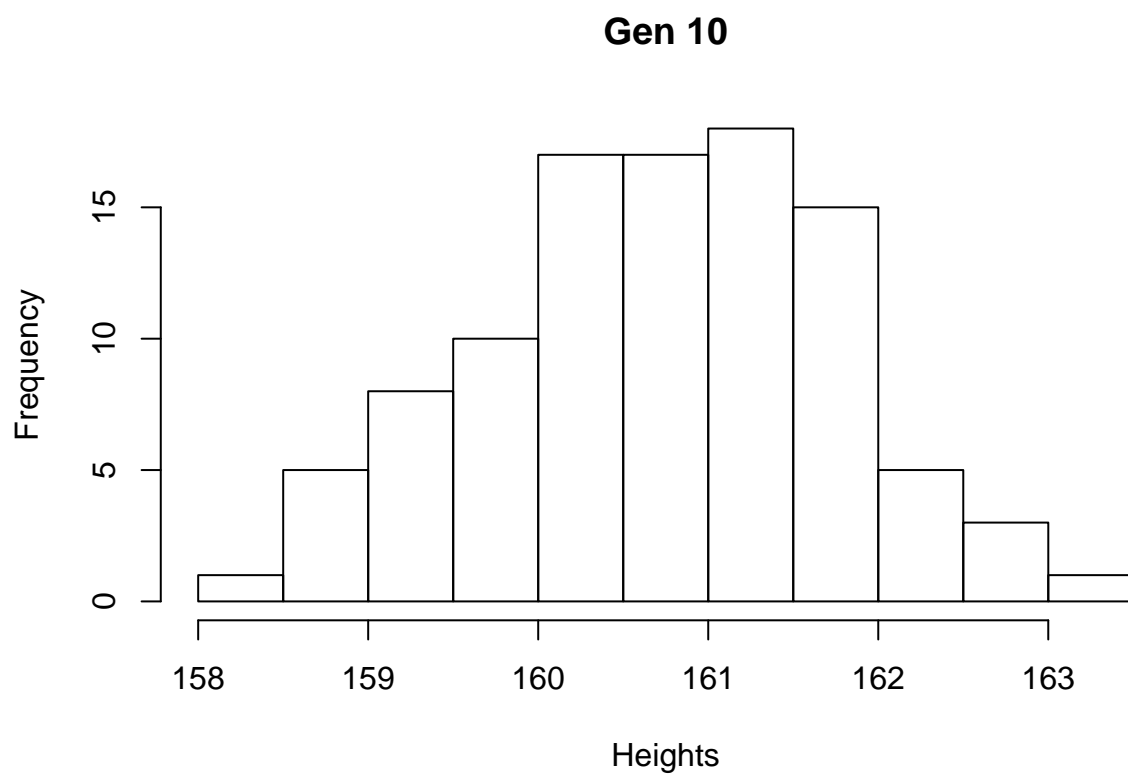


### Heights

```
hist(next_gen9$m, main = 'Gen 9',xlab = 'Heights')
```

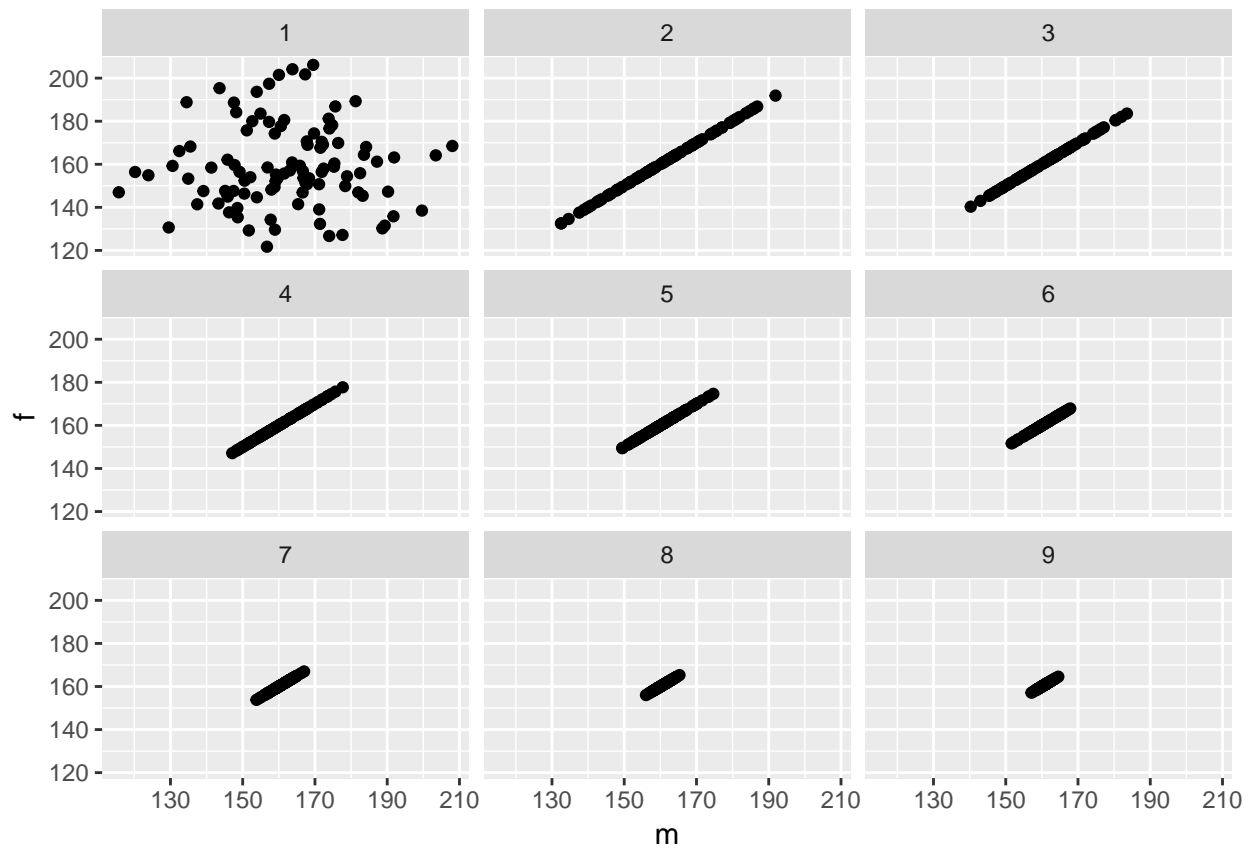


```
hist(next_gen10$m, main = 'Gen 10',xlab = 'Heights')
```



## Question 2

```
library(ggplot2)
gens <- data.frame(rbind(pop,next_gen2,next_gen3,next_gen4,next_gen5,next_gen6,next_gen7,next_gen8,next_gen9))
# Creating a new dataframe with 9 generations
gens[, 'generation'] <- rep(1:9, each=100)
# Creating a new column that indicates which generation each observation belongs to
ggplot(data=gens)+geom_point(mapping=aes(x=m, y=f))+facet_wrap(~generation)
```



## Question 3

```
set.seed(100)

#sample 1
loops <- 1e3
group <- rnorm(250, 60, 20)
assign <- sample(c(0,1), 250, replace = TRUE)
exp <- data.frame(group, assign)
for(x in 1:250) {
  if(exp[x,'assign'] == 1) {
    exp[x,1] <- exp[x,1] +5
  }
  x <- x+1
}
```

```

for(i in seq(loops)) {
  conf.int <- matrix(nrow=loops,ncol=2)
  for(i in seq(nrow(conf.int))) {
    r <- sample(group, replace = TRUE)
    conf.int[i,1] <- mean(r) + qnorm(0.025)*sd(r) / sqrt(length(r))
    conf.int[i,2] <- mean(r) + qnorm(0.975)*sd(r) / sqrt(length(r))
  }
}
bmeans <- numeric(2)
bmeans[1] <- mean(conf.int[,1])
bmeans[2] <- mean(conf.int[,2])
bmeans

#sample 2

group2 <- rnorm(500, 60, 20)
assign2 <- sample(c(0,1), 500, replace = TRUE)
exp2 <- data.frame(group2, assign2)
for(x in 1:500) {
  if(exp2[x,'assign2'] == 1) {
    exp2[x,1] <- exp2[x,1] +5
  }
  x <- x+1
}

for(i in seq(loops)) {
  conf.int <- matrix(nrow=loops,ncol=2)
  for(i in seq(nrow(conf.int))) {
    r <- sample(group2, replace = TRUE)
    conf.int[i,1] <- mean(r) + qnorm(0.025)*sd(r) / sqrt(length(r))
    conf.int[i,2] <- mean(r) + qnorm(0.975)*sd(r) / sqrt(length(r))
  }
}
cmeans <- numeric(2)
cmeans[1] <- mean(conf.int[,1])
cmeans[2] <- mean(conf.int[,2])
cmeans

# sample 3

group3 <- rnorm(750, 60, 20)
assign3 <- sample(c(0,1), 750, replace = TRUE)
exp3 <- data.frame(group3, assign3)
for(x in 1:750) {
  if(exp3[x,'assign3'] == 1) {
    exp3[x,1] <- exp3[x,1] +5
  }
  x <- x+1
}

for(i in seq(loops)) {
  conf.int <- matrix(nrow=loops,ncol=2)
  for(i in seq(nrow(conf.int))) {

```



```

    r <- sample(group3, replace = TRUE)
    conf.int[i,1] <- mean(r) + qnorm(0.025)*sd(r) / sqrt(length(r))
    conf.int[i,2] <- mean(r) + qnorm(0.975)*sd(r) / sqrt(length(r))
  }
}
dmeans <- numeric(2)
dmeans[1] <- mean(conf.int[,1])
dmeans[2] <- mean(conf.int[,2])
dmeans

# sample 4

group4 <- rnorm(1000, 60, 20)
assign4 <- sample(c(0,1), 1000, replace = TRUE)
exp4 <- data.frame(group4, assign4)
for(x in 1:1000) {
  if(exp4[x,'assign4'] == 1) {
    exp4[x,1] <- exp4[x,1] +5
  }
  x <- x+1
}

for(i in seq(loops)) {
  conf.int <- matrix(nrow=loops,ncol=2)
  for(i in seq(nrow(conf.int))) {
    r <- sample(group4, replace = TRUE)
    conf.int[i,1] <- mean(r) + qnorm(0.025)*sd(r) / sqrt(length(r))
    conf.int[i,2] <- mean(r) + qnorm(0.975)*sd(r) / sqrt(length(r))
  }
}
emeans <- numeric(2)
emeans[1] <- mean(conf.int[,1])
emeans[2] <- mean(conf.int[,2])
emeans

# sample 5

group5 <- rnorm(1250, 60, 20)
assign5 <- sample(c(0,1), 1250, replace = TRUE)
exp5 <- data.frame(group5, assign5)
for(x in 1:1250) {
  if(exp5[x,'assign5'] == 1) {
    exp5[x,1] <- exp5[x,1] +5
  }
  x <- x+1
}

for(i in seq(loops)) {
  conf.int <- matrix(nrow=loops,ncol=2)
  for(i in seq(nrow(conf.int))) {
    r <- sample(group5, replace = TRUE)
    conf.int[i,1] <- mean(r) + qnorm(0.025)*sd(r) / sqrt(length(r))
    conf.int[i,2] <- mean(r) + qnorm(0.975)*sd(r) / sqrt(length(r))
  }
}

```

```

    }
  }
  fmeans <- numeric(2)
  fmeans[1] <- mean(conf.int[,1])
  fmeans[2] <- mean(conf.int[,2])
  fmeans

# sample 6

group6 <- rnorm(1500, 60, 20)
assign6 <- sample(c(0,1), 1500, replace = TRUE)
exp6 <- data.frame(group6, assign6)
for(x in 1:1500) {
  if(exp6[x,'assign6'] == 1) {
    exp6[x,1] <- exp6[x,1] +5
  }
  x <- x+1
}

for(i in seq(loops)) {
  conf.int <- matrix(nrow=loops,ncol=2)
  for(i in seq(nrow(conf.int))) {
    r <- sample(group6, replace = TRUE)
    conf.int[i,1] <- mean(r) + qnorm(0.025)*sd(r) / sqrt(length(r))
    conf.int[i,2] <- mean(r) + qnorm(0.975)*sd(r) / sqrt(length(r))
  }
}
gmeans <- numeric(2)
gmeans[1] <- mean(conf.int[,1])
gmeans[2] <- mean(conf.int[,2])
gmeans

# sample 7

group7 <- rnorm(1750, 60, 20)
assign7 <- sample(c(0,1), 1750, replace = TRUE)
exp7 <- data.frame(group7, assign7)
for(x in 1:1750) {
  if(exp7[x,'assign7'] == 1) {
    exp7[x,1] <- exp7[x,1] +5
  }
  x <- x+1
}

for(i in seq(loops)) {
  conf.int <- matrix(nrow=loops,ncol=2)
  for(i in seq(nrow(conf.int))) {
    r <- sample(group7, replace = TRUE)
    conf.int[i,1] <- mean(r) + qnorm(0.025)*sd(r) / sqrt(length(r))
    conf.int[i,2] <- mean(r) + qnorm(0.975)*sd(r) / sqrt(length(r))
  }
}

```

```

hmeans <- numeric(2)
hmeans[1] <- mean(conf.int[,1])
hmeans[2] <- mean(conf.int[,2])
hmeans

# sample 8

group8 <- rnorm(2000, 60, 20)
assign8 <- sample(c(0,1), 2000, replace = TRUE)
exp8 <- data.frame(group8, assign8)
for(x in 1:2000) {
  if(exp8[x,'assign8'] == 1) {
    exp8[x,1] <- exp8[x,1] +5
  }
  x <- x+1
}

for(i in seq(loops)) {
  conf.int <- matrix(nrow=loops,ncol=2)
  for(i in seq(nrow(conf.int))) {
    r <- sample(group8, replace = TRUE)
    conf.int[i,1] <- mean(r) + qnorm(0.025)*sd(r) / sqrt(length(r))
    conf.int[i,2] <- mean(r) + qnorm(0.975)*sd(r) / sqrt(length(r))
  }
}
imeans <- numeric(2)
imeans[1] <- mean(conf.int[,1])
imeans[2] <- mean(conf.int[,2])
imeans

# sample 9

group9 <- rnorm(2250, 60, 20)
assign9 <- sample(c(0,1), 2250, replace = TRUE)
exp9 <- data.frame(group9, assign9)
for(x in 1:2250) {
  if(exp9[x,'assign9'] == 1) {
    exp9[x,1] <- exp9[x,1] +5
  }
  x <- x+1
}

for(i in seq(loops)) {
  conf.int <- matrix(nrow=loops,ncol=2)
  for(i in seq(nrow(conf.int))) {
    r <- sample(group9, replace = TRUE)
    conf.int[i,1] <- mean(r) + qnorm(0.025)*sd(r) / sqrt(length(r))
    conf.int[i,2] <- mean(r) + qnorm(0.975)*sd(r) / sqrt(length(r))
  }
}
jmeans <- numeric(2)
jmeans[1] <- mean(conf.int[,1])
jmeans[2] <- mean(conf.int[,2])

```

```

jmeans

# sample 10

group10 <- rnorm(2500, 60, 20)
assign10 <- sample(c(0,1), 2500, replace = TRUE)
exp10 <- data.frame(group10, assign10)
for(x in 1:2500) {
  if(exp10[x,'assign10'] == 1) {
    exp10[x,1] <- exp10[x,1] +5
  }
  x <- x+1
}

for(i in seq(loops)) {
  conf.int <- matrix(nrow=loops,ncol=2)
  for(i in seq(nrow(conf.int))) {
    r <- sample(group10, replace = TRUE)
    conf.int[i,1] <- mean(r) + qnorm(0.025)*sd(r) / sqrt(length(r))
    conf.int[i,2] <- mean(r) + qnorm(0.975)*sd(r) / sqrt(length(r))
  }
}
kmeans <- numeric(2)
kmeans[1] <- mean(conf.int[,1])
kmeans[2] <- mean(conf.int[,2])
kmeans

# Putting it all together
totmeans <- rbind(bmeans, cmeans, dmeans, emeans, fmeans, gmeans, hmeans, imeans, jmeans, kmeans)

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)
}

par(new=FALSE)
plot(1:10, totmeans[,1],
     xlim=c(1,10),
     ylim=c(10,80),
     xlab="Bootstrap sample",
     ylab="95% CI Mean",
     col=makeTransparent('blue',alpha=0.5)#)
)
lines(totmeans[,2], col='blue')
lines(totmeans[,1], col='red')

```

**JC Grading -1** An extra parenthesis (now commented out) caused the program to not compile. Also, I would be happy to talk through with you to write this code to run much more quickly and without having to

repeat the same general code each time.

**JC Grading -2** The treatment assignment is assigned randomly, but you don't want to sample the treatment assignment with replacement.

**JC Grading -5** After getting your bootstrap samples, summarize the mean and standard error separately by treatment group.

## Question 4

part 1

```
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)
}

makePatient()

## [[1]]
## [1] "Pag"
##
## [[2]]
## [1] male
## Levels: female male
##
## [[3]]
## [1] "1989-11-05"
##
## [[4]]
## [1] "2010-04-26" "2012-02-14" "2012-06-19" "2011-10-18" "2010-01-04"
##
## [[5]]
## [1] 78 84 88 70 79
##
## [[6]]
## [1] 98.02 98.27 98.55 97.97 98.71
##
## [[7]]
## [1] 0.88 0.09 0.50 0.75 0.61

class(makePatient)

## [1] "function"

set.seed(8)
medicalRecord <- makePatient()
```

```

medicalRecord

## [[1]]
## [1] "Mev"
##
## [[2]]
## [1] male
## Levels: female male
##
## [[3]]
## [1] "1976-08-09"
##
## [[4]]
## [1] "2011-03-14" "2013-10-30" "2013-02-27" "2012-08-23" "2011-11-16"
##
## [[5]]
## [1] 67 81 95 74 81
##
## [[6]]
## [1] 98.33 98.16 99.00 98.49 98.67
##
## [[7]]
## [1] 0.62 0.93 0.18 0.39 0.34

names(medicalRecord) <- c('name','gender','date_of_birth','date_of_admission','pulse','temp','fluid')
class(medicalRecord) <- "medicalRecord"
class(medicalRecord)

## [1] "medicalRecord"

part 2

mean.medicalRecord <- function(x) {
  c(mean(x$pulse),
    mean(x$temp),
    mean(x$fluid))
}

mean.medicalRecord(medicalRecord)

## [1] 79.600 98.530 0.492

print.medicalRecord <- function(x) {
  name <- x$name
  gender <- x$gender
  date_of_birth <- x$date_of_birth
  date_of_admission <- x$date_of_admission[order(x$date_of_admission)]
  pulse <- x$pulse[order(x$date_of_admission)]
  temp <- x$temp[order(x$date_of_admission)]
  fluid <- x$fluid[order(x$date_of_admission)]
  treatments <- data.frame(date_of_admission, pulse, temp, fluid)
  print(name)
  print(gender)
  print(date_of_birth, sep='\n')
  print(treatments)
}

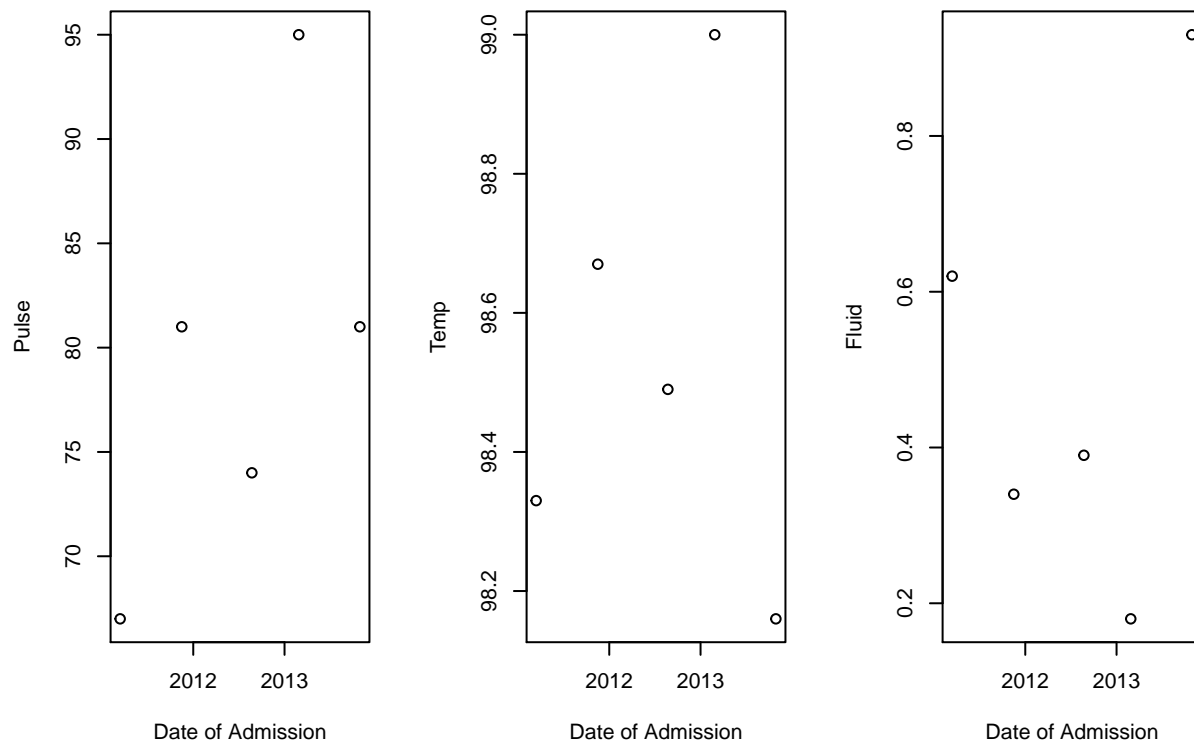
```

```
print.medicalRecord(medicalRecord)
```

```
## [1] "Mev"
## [1] male
## Levels: female male
## [1] "1976-08-09"
##   date_of_admission pulse  temp fluid
## 1      2011-03-14     67 98.33  0.62
## 2      2011-11-16     81 98.67  0.34
## 3      2012-08-23     74 98.49  0.39
## 4      2013-02-27     95 99.00  0.18
## 5      2013-10-30     81 98.16  0.93
```

```
plot.medicalRecord <- function(x) {
  par(mfrow=c(1,3))
  plot(x$date_of_admission, x$pulse, xlab = 'Date of Admission', ylab='Pulse')
  plot(x$date_of_admission, x$temp, xlab = 'Date of Admission', ylab='Temp')
  plot(x$date_of_admission, x$fluid, xlab = 'Date of Admission', ylab='Fluid')
}
```

```
plot.medicalRecord(medicalRecord)
```



part 3

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

mean.cohort <- function(x){
  for(medicalRecord in x){
    class(medicalRecord) <- 'medicalRecord'
```

```

    #Separate each record more cleanly
    cat(sprintf("~Measurements for Patient: %s\n", patient$name))
    cat(mean(medicalRecord), "\n")
  }
}

print.cohort <- function(x){
  for(medicalRecord in x){
    class(medicalRecord) <- 'medicalRecord'
    #Separate each record more cleanly
    cat(sprintf("\n~Medical Record"), "\n")
    print(medicalRecord)
  }
}

```