

Bios 6301: Assignment 6

Elizabeth Sigworth

Thursday, 1 December

Grade 50/50

Due Thursday, 1 December, 1:00 PM

$5^{n=\text{day}}$ 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:

```
set.seed(280)
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 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.

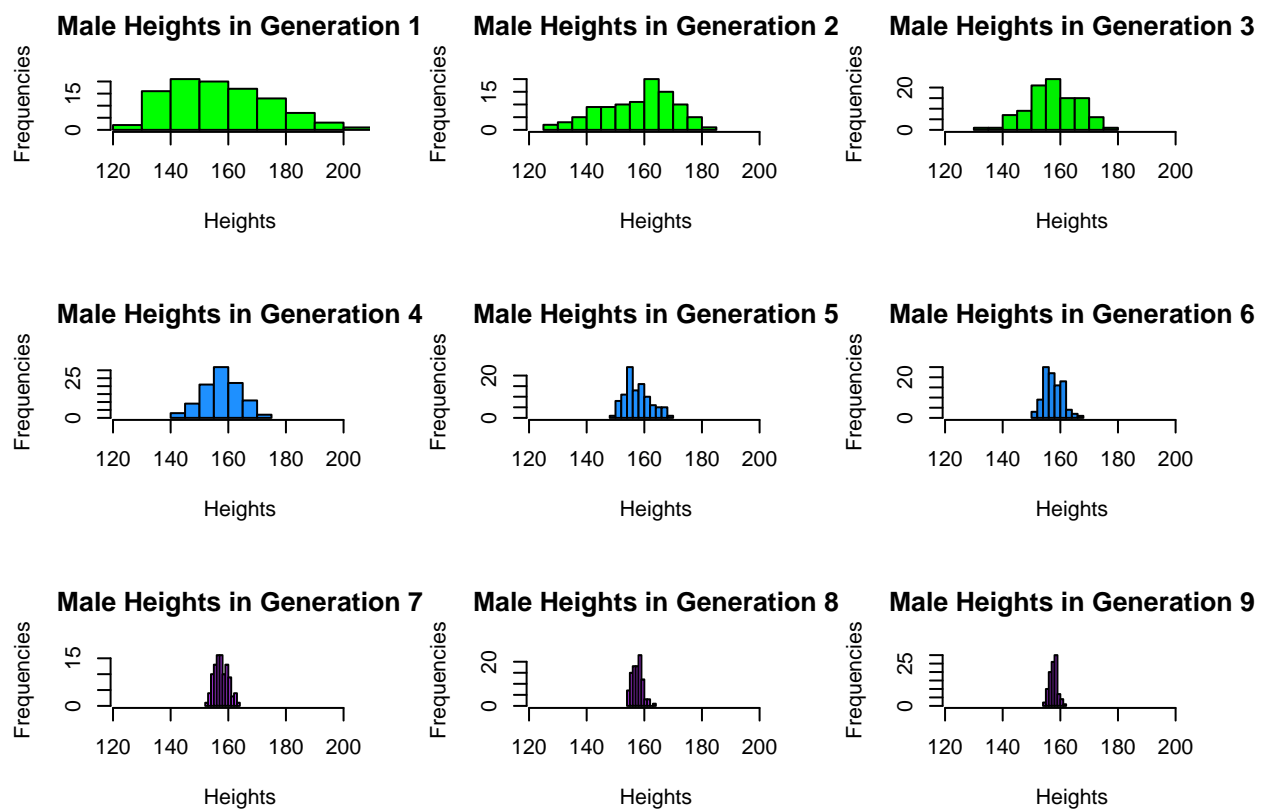
```
gens2 <- data.frame(m=rep(NA,900),f=rep(NA,900),gen=rep(1:9,each=100))
gens2[1:100,1:2] <- pop
for(i in 2:9){
  start <- (i-2)*100+1
```

```

end <- (i-1)*100
data <- gens2[start:end,1:2]
assign.start <- ((i-1)*100+1)
assign.end <- i*100
gens2[assign.start:assign.end,1:2] <- next_gen(data)
}

par(mfrow=c(3,3))
palette <- c("green","green1","green2","dodgerblue","dodgerblue1",
            "dodgerblue2","darkorchid1","darkorchid2","darkorchid3")
for(i in 1:9){
  title <- paste("Male Heights in Generation ", i, sep="")
  hist(gens2$m[which(gens2$gen==i)],main=title,xlab="Heights",ylab="Frequencies",col=palette[i],
       xlim = c(min(gens2$m[which(gens2$gen==1)]),max(gens2$m[which(gens2$gen==1)])))
}

```



Question 2

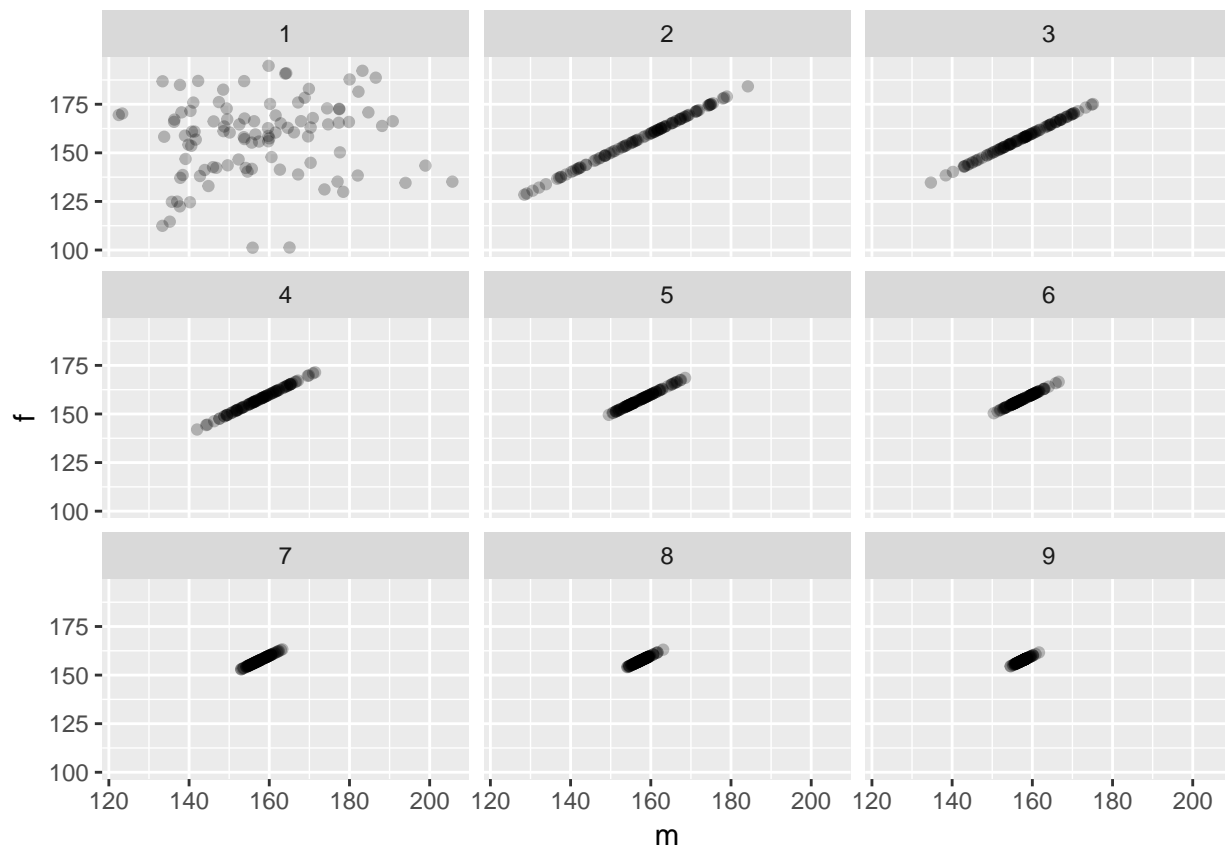
10 points

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

```

library(ggplot2)
p <- ggplot(data = gens2, aes(x=m,y=f)) + facet_wrap(~gen,nrow=3)
g <- p + geom_point(alpha=.25) +
  scale_x_continuous(limits = c(min(gens2$m[which(gens2$gen==1)]),max(gens2$m[which(gens2$gen==1)]))) +
  scale_y_continuous(limits = c(min(gens2$f[which(gens2$gen==1)]),max(gens2$f[which(gens2$gen==1)])))
g

```



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)

```
set.seed(496)
base.sample <- function(n){
  start.sample <- data.frame(Treatment=rbinom(n,1,.5))
  start.sample$Outcome <- rnorm(n,60,20)
  for (j in 1:n) {
    if (start.sample$Treatment[j]==1) {
      start.sample$Outcome[j] <- start.sample$Outcome[j] + 5
    }
  }
  return(start.sample)
}

bootstrap.mean <- function(data,n){
  means.0 <- vector()
```

```

means.1 <- vector()
for(i in 1:1000){
  test.sample <- data[sample(nrow(data),n,replace=TRUE),]
  means <- aggregate(test.sample[,2],list(test.sample$Treatment),mean)
  means.0[i] <- means[1,2]
  means.1[i] <- means[2,2]
}
total.0 <- mean(means.0)
total.1 <- mean(means.1)
bounds.0 <- quantile(means.0,c(0.025,0.975))
names(bounds.0) <- NULL
bounds.1 <- quantile(means.1,c(0.025,0.975))
names(bounds.1) <- NULL
results.0 <- data.frame(mean=total.0,lower=bounds.0[1],upper=bounds.0[2],n)
results.1 <- data.frame(mean=total.1,lower=bounds.1[1],upper=bounds.1[2],n)
final <- rbind(results.0,results.1)
return(final)
}

straps <- data.frame()
for(i in 1:10){
  size <- 250*i
  strap <- cbind(strap=as.factor(c(i,i)),group=c(0,1),bootstrap.mean(base.sample(size),size))
  straps <- rbind(straps,strap)
}

```

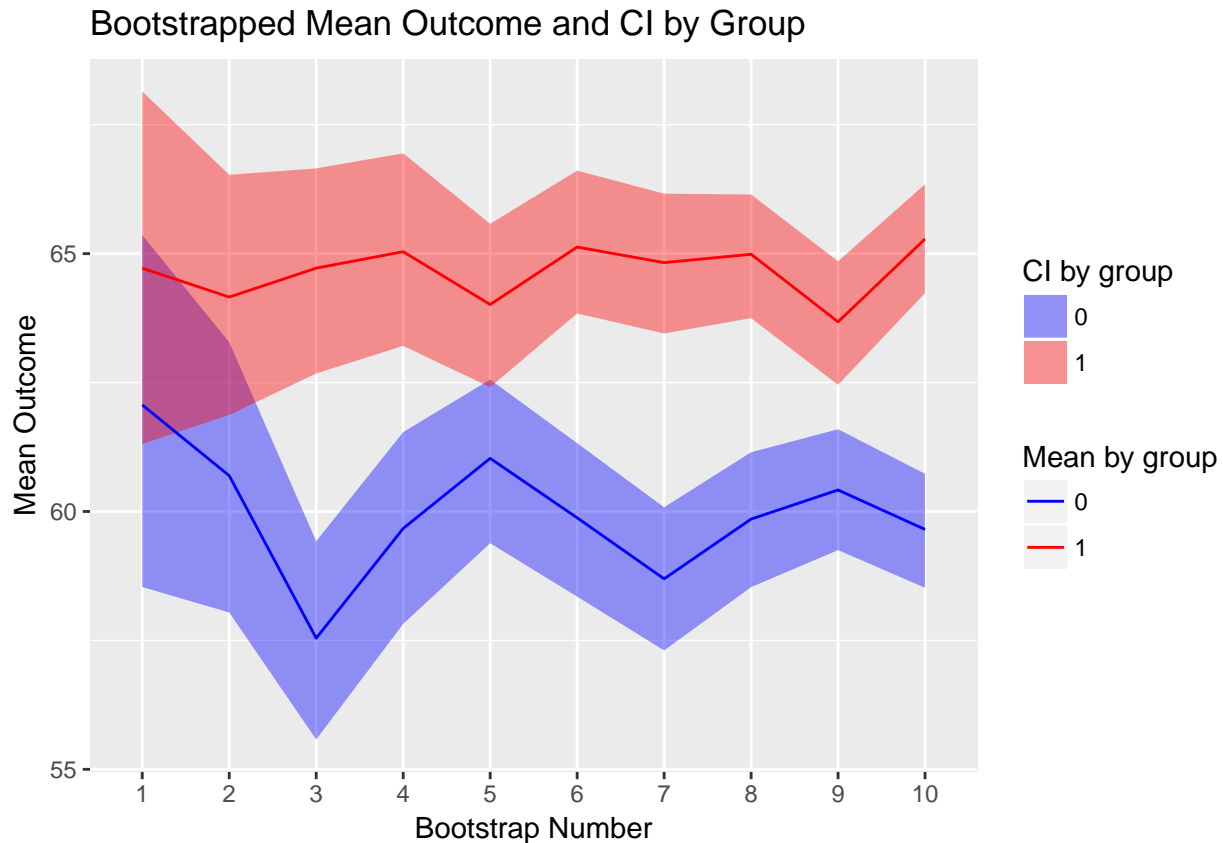
straps

##	strap	group	mean	lower	upper	n
## 1	1	0	62.06658	58.53494	65.36099	250
## 2	1	1	64.71846	61.30786	68.14628	250
## 3	2	0	60.69346	58.04222	63.28050	500
## 4	2	1	64.15819	61.87193	66.52928	500
## 5	3	0	57.54294	55.57933	59.42326	750
## 6	3	1	64.71853	62.67566	66.65292	750
## 7	4	0	59.67094	57.81941	61.53862	1000
## 8	4	1	65.03752	63.21173	66.94556	1000
## 9	5	0	61.03048	59.38398	62.55720	1250
## 10	5	1	64.01325	62.41021	65.57806	1250
## 11	6	0	59.87766	58.34965	61.32383	1500
## 12	6	1	65.12767	63.84059	66.60979	1500
## 13	7	0	58.69369	57.30558	60.08155	1750
## 14	7	1	64.82536	63.45169	66.16315	1750
## 15	8	0	59.85279	58.53294	61.14799	2000
## 16	8	1	64.98773	63.74892	66.14614	2000
## 17	9	0	60.41698	59.25141	61.59454	2250
## 18	9	1	63.67754	62.45588	64.85133	2250
## 19	10	0	59.65159	58.52069	60.73054	2500
## 20	10	1	65.28460	64.23559	66.34646	2500

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)

```
p <- ggplot(data=straps, aes(x=strap, y=mean)) +
  scale_fill_manual(values=c("blue","red"),name="CI by group") +
  scale_colour_manual(values=c("blue","red"),name="Mean by group") +
  geom_ribbon(aes(ymin=straps$lower, ymax=straps$upper,group=factor(group),
    fill=factor(group)), alpha=0.4) +
  geom_line(aes(group=factor(group), colour=factor(group))) +
  ylab("Mean Outcome") +
  xlab("Bootstrap Number") +
  ggtitle("Bootstrapped Mean Outcome and CI by Group")
```

p



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)

```

set.seed(8)
record <- makePatient()
names(record) <- c("name", "gender", "date_of_birth", "date_of_admission", "pulse", "temperature", "fluid_intake")
class(record) <- "medicalRecord"
print.default(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
##
## $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"

```

```
class(record)
```

```
## [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)

```

mean.medicalRecord <- function(x){
  pulse <- mean(x$pulse)
  temperature <- mean(x$temperature)
  fluids <- mean(x$fluid_intake)
}

```

```

results <- list(mean_pulse = pulse, mean_temperature = temperature, mean_fluids = fluids)
return(results)
}

print.medicalRecord <- function(x){
  n_obs <- length(x$date_of_admission)
  chart <- data.frame(Name=rep(x$name,n_obs),Gender=rep(x$gender,n_obs),
                      Date_of_Birth=rep(x$date_of_birth,n_obs),Date_of_Admission=x$date_of_admission,
                      Pulse=x$pulse,Temperature=x$temperature,Fluid_Intake=x$fluid_intake)
  chart <- chart[order(chart$Date_of_Admission),]
  return(chart)
}

plot.medicalRecord <- function(x){
  chart <- print.medicalRecord(x)
  par(mfrow=c(1,3))
  plot(x=chart$Date_of_Admission,y=chart$Pulse,main="Pulse by Admission",
       xlab="Date of Admission",ylab="Pulse",type="l",lwd=2,col="red")
  plot(x=chart$Date_of_Admission,y=chart$Temperature,main="Temperature by Admission",
       xlab="Date of Admission",ylab="Temperature",type="l",lwd=2,col="green")
  plot(x=chart$Date_of_Admission,y=chart$Fluids,main="Fluid Intake by Admission",
       xlab="Date of Admission",ylab="Fluid Intake",type="l",lwd=2,col="purple")
}

mean(record)

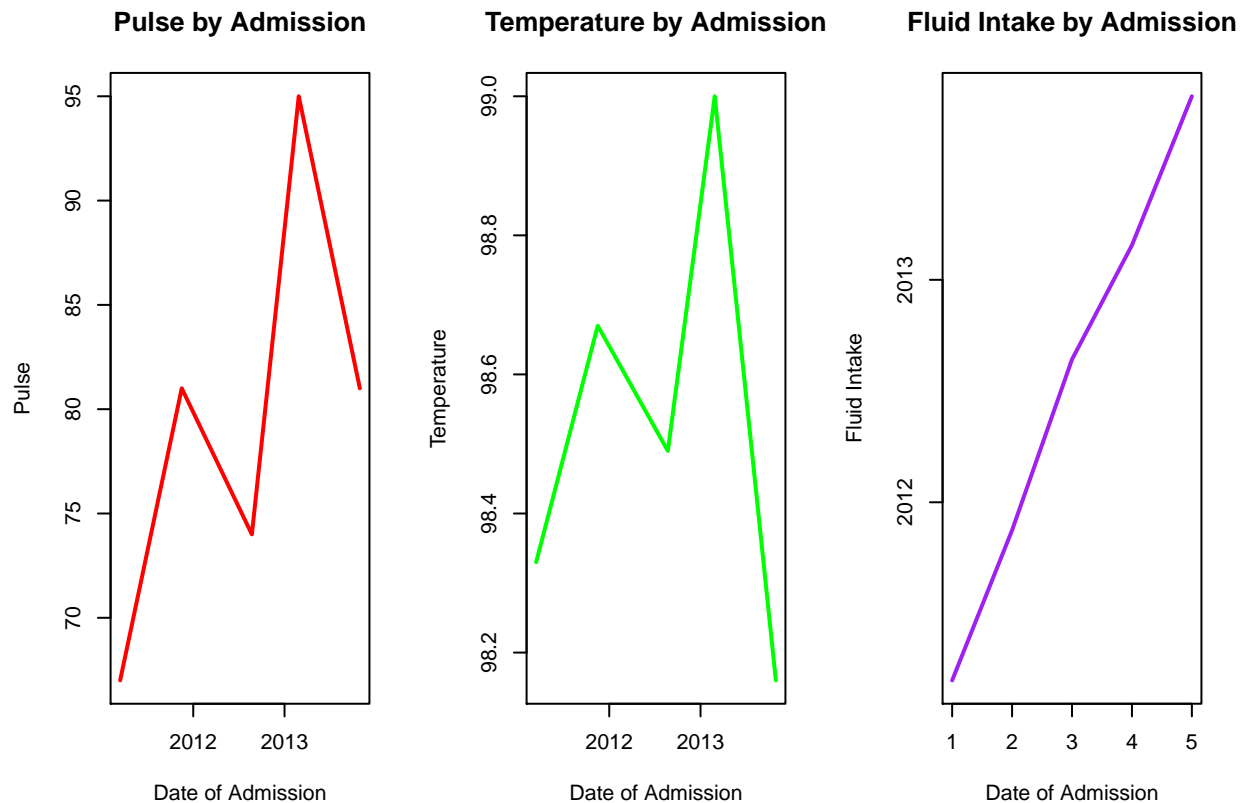
## $mean_pulse
## [1] 79.6
##
## $mean_temperature
## [1] 98.53
##
## $mean_fluids
## [1] 0.492

print(record)

##   Name Gender Date_of_Birth Date_of_Admission Pulse Temperature
## 1  Mev    male   1976-08-09      2011-03-14     67         98.33
## 5  Mev    male   1976-08-09      2011-11-16     81         98.67
## 4  Mev    male   1976-08-09      2012-08-23     74         98.49
## 3  Mev    male   1976-08-09      2013-02-27     95         99.00
## 2  Mev    male   1976-08-09      2013-10-30     81         98.16
##   Fluid_Intake
## 1           0.62
## 5           0.34
## 4           0.39
## 3           0.18
## 2           0.93

plot(record)

```



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)

```
mean.cohort <- function(x){
  names <- vector()
  pulses <- vector()
  temps <- vector()
  fluids <- vector()
  for(i in 1:ncol(x)){
    names[i] <- x[,i]$name
    pulses[i] <- mean(x[,i]$pulse)
    temps[i] <- mean(x[,i]$temperature)
    fluids[i] <- mean(x[,i]$fluid_intake)
  }
  results <- data.frame(Name=names,Mean_Pulse=pulses,Mean_Temperature=temps,Mean_Fluid_Intake=fluids)
  return(results)
}

print.cohort <- function(x){
  cohort.chart <- data.frame()
  for(i in 1:ncol(x)){
    n_obs <- length(x[,i]$date_of_admission)
    chart <- data.frame(Name=rep(x[,i]$name,n_obs),Gender=rep(x[,i]$gender,n_obs),
                       Date_of_Birth=rep(x[,i]$date_of_birth,n_obs),
                       Date_of_Admission=x[,i]$date_of_admission,
                       Pulse=x[,i]$pulse,Temperature=x[,i]$temperature,Fluid_Intake=x[,i]$fluid_intake)
    cohort.chart <- cohort.chart[order(cohort.chart$Date_of_Admission),]
  }
}
```



```

    cohort.chart <- rbind(cohort.chart,chart)
  }
  return(cohort.chart)
}
set.seed(8)
cohort <- replicate(10,makePatient())
rownames(cohort) <- c("name","gender","date_of_birth","date_of_admission",
                      "pulse","temperature","fluid_intake")
class(cohort) <- "cohort"
mean(cohort)

```

```

##      Name Mean_Pulse Mean_Temperature Mean_Fluid_Intake
## 1   Mev    79.60000         98.53000         0.4920000
## 2   Yul    78.00000         98.49500         0.2450000
## 3   Zet    81.50000         98.44000         0.4033333
## 4   Qih    78.00000         98.60000         0.6500000
## 5   Wut    88.33333         98.05000         0.5866667
## 6   Juy    83.50000         98.45000         0.4525000
## 7   God    83.00000         98.01000         0.9700000
## 8   Fut    77.50000         98.14833         0.3366667
## 9   Pet    77.00000         98.83000         0.4450000
## 10  Yed    79.33333         98.30000         0.6583333

```

```
print(cohort)
```

```

##      Name Gender Date_of_Birth Date_of_Admission Pulse Temperature
## 1   Mev   male   1976-08-09      2011-03-14      67      98.33
## 5   Mev   male   1976-08-09      2011-11-16      81      98.67
## 4   Mev   male   1976-08-09      2012-08-23      74      98.49
## 3   Mev   male   1976-08-09      2013-02-27      95      99.00
## 2   Mev   male   1976-08-09      2013-10-30      81      98.16
## 11  Yul   male   1988-06-28      2012-01-16      76      98.92
## 21  Yul   male   1988-06-28      2013-08-07      80      98.07
## 6   Zet   female  1970-06-13      2010-03-21      79      98.58
## 51  Zet   female  1970-06-13      2010-04-01      73      98.32
## 41  Zet   female  1970-06-13      2012-08-29      88      98.47
## 31  Zet   female  1970-06-13      2013-06-01      84      98.22
## 12  Zet   female  1970-06-13      2013-11-03      72      98.54
## 22  Zet   female  1970-06-13      2014-02-05      93      98.51
## 13  Qih   female  1987-08-30      2011-06-22      78      98.60
## 32  Wut   male   1974-06-28      2010-04-12      76      98.05
## 14  Wut   male   1974-06-28      2011-02-16      93      98.26
## 23  Wut   male   1974-06-28      2012-04-12      96      97.84
## 42  Juy   male   1983-06-09      2010-03-10      81      99.11
## 15  Juy   male   1983-06-09      2010-03-25      90      98.58
## 33  Juy   male   1983-06-09      2010-04-18      75      98.58
## 24  Juy   male   1983-06-09      2010-06-10      88      97.53
## 16  God   female  1990-02-12      2010-03-12      83      98.01
## 52  Fut   male   1970-01-11      2011-04-07      80      97.87
## 43  Fut   male   1970-01-11      2011-04-14      83      97.91
## 25  Fut   male   1970-01-11      2011-08-16      66      98.49
## 17  Fut   male   1970-01-11      2013-03-15      74      98.38
## 61  Fut   male   1970-01-11      2013-06-20      74      98.41
## 34  Fut   male   1970-01-11      2013-11-12      88      97.83

```

## 18	Pet	male	1979-01-01	2010-10-30	85	98.84
## 26	Pet	male	1979-01-01	2012-05-10	69	98.82
## 44	Yed	male	1977-11-11	2010-01-28	63	97.95
## 35	Yed	male	1977-11-11	2010-03-06	81	98.45
## 19	Yed	male	1977-11-11	2010-07-10	98	98.65
## 62	Yed	male	1977-11-11	2010-08-27	66	97.68
## 53	Yed	male	1977-11-11	2011-06-18	83	98.00
## 27	Yed	male	1977-11-11	2013-01-06	85	99.07
##	Fluid_Intake					
## 1						0.62
## 5						0.34
## 4						0.39
## 3						0.18
## 2						0.93
## 11						0.14
## 21						0.35
## 6						0.22
## 51						0.61
## 41						0.59
## 31						0.25
## 12						0.03
## 22						0.72
## 13						0.65
## 32						0.65
## 14						0.97
## 23						0.14
## 42						0.66
## 15						0.26
## 33						0.60
## 24						0.29
## 16						0.97
## 52						0.36
## 43						0.00
## 25						0.13
## 17						0.31
## 61						0.49
## 34						0.73
## 18						0.60
## 26						0.29
## 44						0.94
## 35						0.67
## 19						0.79
## 62						0.36
## 53						0.69
## 27						0.50