BIOS 6301: Assignment 8

Charlie Rhea

Due Tuesday, 14 November, 1:00 PM

$5^{n=day}$ points taken off for each day late.

30 points total.

Submit a single knitr file (named `homework8.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 `homework8.rmd` or include author name may result in 5 points taken off.

Question 1

15 points

Install the `readxl` package and run the following

```
library(readxl)
fn <- 'icd10.xlsx'
if(file.access(fn, mode = 4) == -1) {
    url <- "https://www.cdc.gov/nhsn/xls/icd10-pcs-pcm-nhsn-opc.xlsx"
    download.file(url, destfile = fn, mode = 'wb')
}
dat <- readxl::read_excel(fn, sheet = 2)
```

1. Show the class of `dat`. (1 point)

Object 'dat' has 3 classes: 'tbl_df', 'tbl', and 'data.frame'.

```
class(dat)
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

2. Show the methods available for objects of the given class (if there are multiple classes, show methods for all classes). (3 points)

```
methods(class="tbl_df")
```

```
##  [1] [             [[            [[<-          [<-           $
##  [6] $<-           as.data.frame coerce        initialize    names<-
## [11] Ops           row.names<-   show          slotsFromS3   str
## see '?methods' for accessing help and source code
```

```
methods(class="tbl")
```

```
##  [1] [[<-        [<-         $<-         coerce      format      initialize
##  [7] Ops         print       show        slotsFromS3
## see '?methods' for accessing help and source code
```

```
methods(class="data.frame")
```

```
##  [1] [             [[            [[<-          [<-           $<-
##  [6] aggregate     anyDuplicated anyNA         as.data.frame as.list
## [11] as.matrix     as.vector     by            cbind         coerce
## [16] dim           dimnames      dimnames<-    droplevels    duplicated
## [21] edit          format        formula       head          initialize
## [26] is.na         Math          merge         na.exclude    na.omit
## [31] Ops           plot          print         prompt        rbind
## [36] row.names     row.names<-   rowsum        show          slotsFromS3
## [41] split         split<-       stack         str           subset
## [46] summary       Summary       t             tail          transform
## [51] type.convert  unique        unstack       within        xtfrm
## see '?methods' for accessing help and source code
```

3. If you call `print(dat)`, what print method is being dispatched? (1 point)

There is a 'print.tbl_df' method within print. Since we know 'tbl_df' is the first class type for 'dat', then 'print.tbl_df' is the method that is being dispatched when we call print(dat).

```
#methods(print)
print(head(dat))
```

```
## # A tibble: 6 x 4
##   Procedure Code Categ~1 'ICD-10-PCS Codes' Procedure Code Descr~2 'Code Status'
##   <chr>                  <chr>              <chr>                  <chr>
## 1 AAA                    04B00ZZ            Excision of Abdominal~ No change
## 2 AAA                    04B04ZZ            Excision of Abdominal~ No change
## 3 AAA                    04R007Z            Replacement of Abdomi~ No change
## 4 AAA                    04R00JZ            Replacement of Abdomi~ No change
## 5 AAA                    04R00KZ            Replacement of Abdomi~ No change
## 6 AAA                    04R047Z            Replacement of Abdomi~ No change
## # i abbreviated names: 1: 'Procedure Code Category',
## #   2: 'Procedure Code Descriptions'
```

4. Set the class of `dat` to be a data.frame. (1 point)

```
dat <- as.data.frame(dat)
class(dat)
```

```
## [1] "data.frame"
```

5. If you call `print(dat)` again, what print method is being dispatched? (1 point)

Since 'dat' is now a data.frame, when we call the print commend, the print.data.frame method is being dispatched

```r
print(head(dat))
```

```
##   Procedure Code Category ICD-10-PCS Codes
## 1                     AAA          04B00ZZ
## 2                     AAA          04B04ZZ
## 3                     AAA          04R007Z
## 4                     AAA          04R00JZ
## 5                     AAA          04R00KZ
## 6                     AAA          04R047Z
##                                                                                Procedure Code Descriptions
## 1                                                                 Excision of Abdominal Aorta, Open Approach
## 2                                                 Excision of Abdominal Aorta, Percutaneous Endoscopic Approach
## 3                                  Replacement of Abdominal Aorta with Autologous Tissue Substitute, Open Approach
## 4                                         Replacement of Abdominal Aorta with Synthetic Substitute, Open Approach
## 5                                Replacement of Abdominal Aorta with Nonautologous Tissue Substitute, Open Approach
## 6 Replacement of Abdominal Aorta with Autologous Tissue Substitute, Percutaneous Endoscopic Approach
##   Code Status
## 1   No change
## 2   No change
## 3   No change
## 4   No change
## 5   No change
## 6   No change
```

Define a new generic function **nUnique** with the code below.

```r
nUnique <- function(x) {
    UseMethod('nUnique')
}
```

6. Write a default method for **nUnique** to count the number of unique values in an element. (2 points)

Using the nUnique default method, I was able to determine the number of unique values for two variables within the 'dat' dataset. For 'Procedure Code Category' I found 39 unique values, and for 'ICD-10-PCS Codes' I found 8993 unique values.

```r
nUnique.default <- function(x) {
    length(unique(x))
}

v1 <- dat$`Procedure Code Category`
nUnique(v1)
```

```
## [1] 39
```

```r
v2 <- dat$`ICD-10-PCS Codes`
nUnique(v2)
```

```
## [1] 8993
```

7. Check your function (2 points)

```r
nUnique(letters) # should return 26
```

```r
nUnique(sample(10, 100, replace = TRUE)) # should return 10 (probably)
```

```
## [1] 10
```

8. Write a data.frame method for `nUnique` to operate on data.frame objects. This version should return counts for each column in a data.frame. (2 points)

```r
nUnique.data.frame <- function(x) {
  for(i in colnames(x)){
    cat("Unique values in", i, ":", length(unique(x[,i])), "\n")
}}
```

9. Check your function (2 points)

```r
nUnique(dat)
```

Question 2

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'))
  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)
p <- makePatient()
names(p) <- c("name", "gender", "date_of_birth", "date_of_admission", "pulse", "temperature", "fluid_in
class(p) <- 'medicalRecord'
print(p)
```

```
## $name
## [1] "Yes"
##
## $gender
## [1] male
## Levels: female male
##
## $date_of_birth
## [1] "1977-05-03"
##
## $date_of_admission
## [1] "2013-06-09" "2013-07-02"
##
## $pulse
## [1] 79 78
##
## $temperature
## [1] 98.07 97.50
##
## $fluid_intake
## [1] 0.28 0.52
##
## 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. (5 points) points)

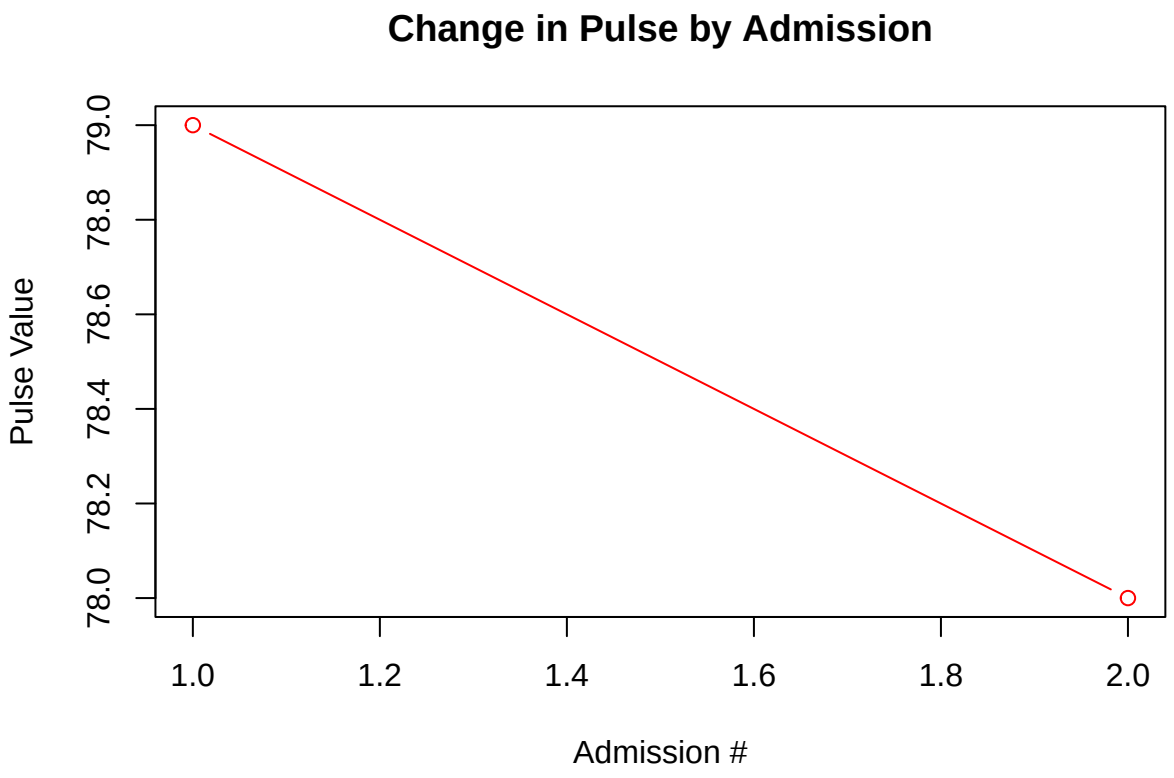
```
mean.medicalRecord <- function(x) {
  list(mean(p[[5]]),mean(p[[6]]), mean(p[[7]]))
}

p2 <- mean(p)
names(p2) <- c("Avg. Pulse", "Avg. Temperature", " Avg. Fluid Intake")
print(p2)
```

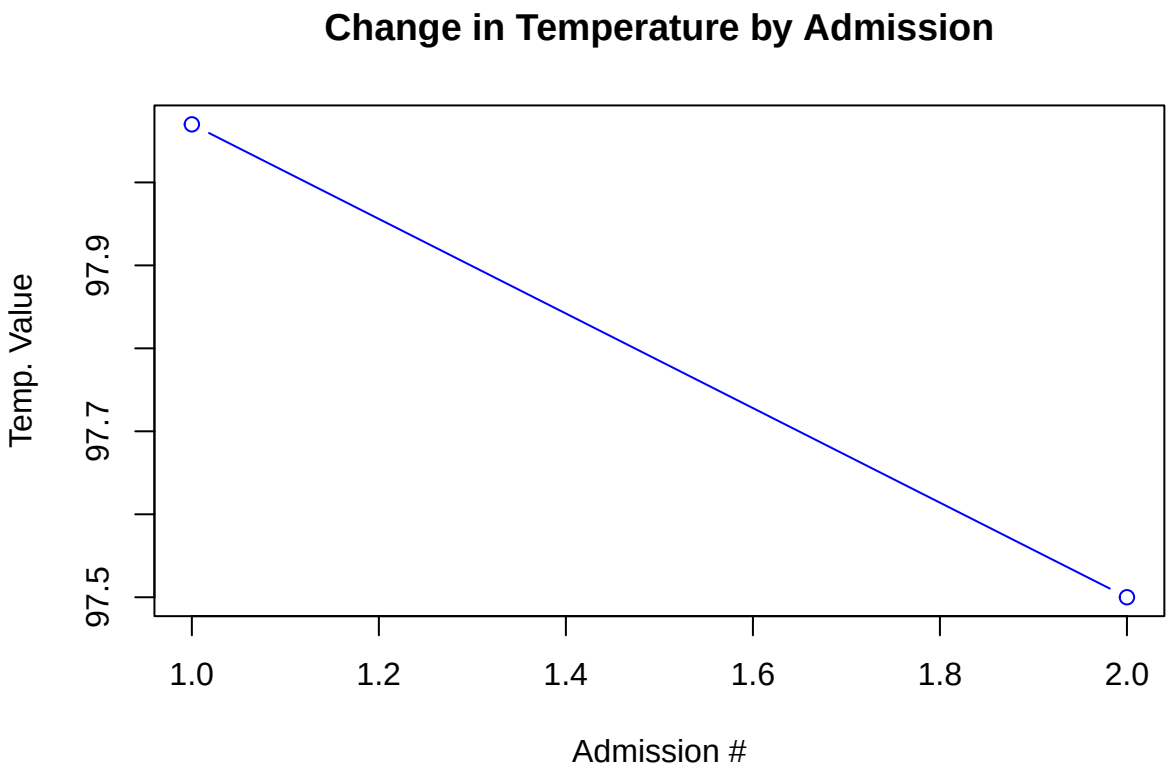
```
## $`Avg. Pulse`
## [1] 78.5
##
## $`Avg. Temperature`
## [1] 97.785
##
## $` Avg. Fluid Intake`
## [1] 0.4
```

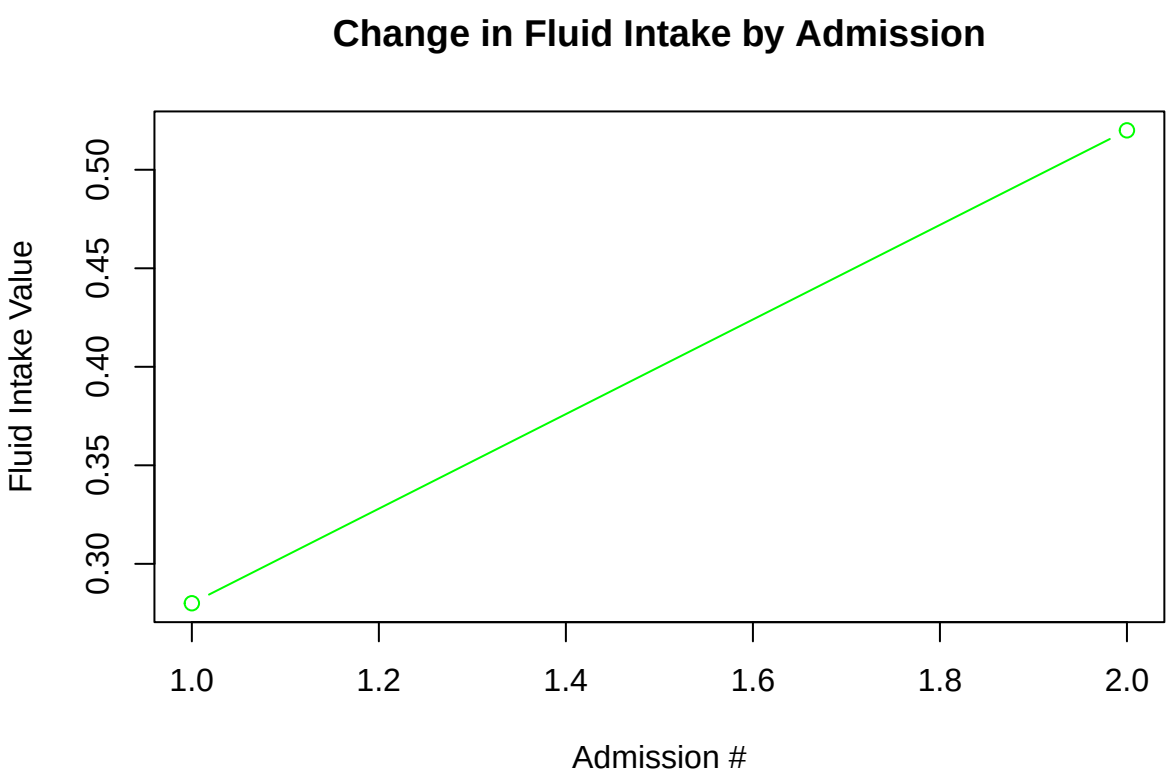
```
print.medicalRecord<- function(med) {
  cat(sprintf("Date of Admission: %s\nName: %s\nGender: %s\nDate of Birth: %s\nPulse: %s\nTemperature: %
med$date_of_admission, med$name, med$gender, med$date_of_birth, med$pulse, med$temperature, med$fluid_in
}

print(p)
```

```
## Date of Admission: 2013-06-09
## Name: Yes
## Gender: male
## Date of Birth: 1977-05-03
## Pulse: 79
## Temperature: 98.07
## Fluid Intake 0.28
##  Date of Admission: 2013-07-02
## Name: Yes
## Gender: male
## Date of Birth: 1977-05-03
## Pulse: 78
## Temperature: 97.5
## Fluid Intake 0.52
##
```

```
plot.medicalRecord <- function(x){
plot(x[[5]],type='b',col='red',main="Change in Pulse by Admission",xlab='Admission #', ylab='Pulse Valu
plot(x[[6]],type='b',col='blue',main="Change in Temperature by Admission",xlab='Admission #', ylab='Temp
plot(x[[7]],type='b',col='green',main="Change in Fluid Intake by Admission",xlab='Admission #', ylab='Fl
}

plot(p)
```




Change in Temperature by Admission
Temp. Value
97.5
97.7
97.9
1.0
1.2
1.4
1.6
1.8
2.0
Admission #



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) points)

```
rm(.Random.seed, envir = globalenv())
set.seed(8)
cohort = list()
  for (i in 1:10){
    x = makePatient()
    names(x) <- c("name", "gender", "date_of_birth", "date_of_admission", "pulse", "temperature", "fluid
    class(x) <- 'medicalRecord'
    cohort[[i]] = x
}
class(cohort) <- 'cohort'
class(cohort)
```

```
## [1] "cohort"
```

```
#Mean Method for Cohort
mean.cohort <- function(x){
  lapply(x,mean)
}

mean(cohort)
```

```
## [[1]]
## [[1]][[1]]
## [1] 78.5
##
## [[1]][[2]]
## [1] 97.785
##
## [[1]][[3]]
## [1] 0.4
##
##
## [[2]]
## [[2]][[1]]
## [1] 78.5
##
## [[2]][[2]]
## [1] 97.785
##
## [[2]][[3]]
## [1] 0.4
##
##
## [[3]]
## [[3]][[1]]
## [1] 78.5
##
## [[3]][[2]]
## [1] 97.785
##
## [[3]][[3]]
## [1] 0.4
##
##
## [[4]]
## [[4]][[1]]
## [1] 78.5
##
## [[4]][[2]]
## [1] 97.785
##
## [[4]][[3]]
## [1] 0.4
##
##
## [[5]]
## [[5]][[1]]
## [1] 78.5
##
## [[5]][[2]]
## [1] 97.785
##
## [[5]][[3]]
## [1] 0.4
##
```

```
##
## [[6]]
## [[6]][[1]]
## [1] 78.5
##
## [[6]][[2]]
## [1] 97.785
##
## [[6]][[3]]
## [1] 0.4
##
##
## [[7]]
## [[7]][[1]]
## [1] 78.5
##
## [[7]][[2]]
## [1] 97.785
##
## [[7]][[3]]
## [1] 0.4
##
##
## [[8]]
## [[8]][[1]]
## [1] 78.5
##
## [[8]][[2]]
## [1] 97.785
##
## [[8]][[3]]
## [1] 0.4
##
##
## [[9]]
## [[9]][[1]]
## [1] 78.5
##
## [[9]][[2]]
## [1] 97.785
##
## [[9]][[3]]
## [1] 0.4
##
##
## [[10]]
## [[10]][[1]]
## [1] 78.5
##
## [[10]][[2]]
## [1] 97.785
##
## [[10]][[3]]
## [1] 0.4
```

```r
#Print Method for Cohort
print.cohort <- function(x){
  for (i in 1:10){
    print(cohort[[i]])
  }
}

print(cohort)
```

```
## Date of Admission: 2013-06-09
## Name: Yes
## Gender: male
## Date of Birth: 1977-05-03
## Pulse: 79
## Temperature: 98.07
## Fluid Intake 0.28
##  Date of Admission: 2013-07-02
## Name: Yes
## Gender: male
## Date of Birth: 1977-05-03
## Pulse: 78
## Temperature: 97.5
## Fluid Intake 0.52
##
## Date of Admission: 2010-11-16
## Name: Fal
## Gender: male
## Date of Birth: 1988-05-24
## Pulse: 76
## Temperature: 98.23
## Fluid Intake 0.18
##  Date of Admission: 2013-09-12
## Name: Fal
## Gender: male
## Date of Birth: 1988-05-24
## Pulse: 96
## Temperature: 98.75
## Fluid Intake 0.96
##  Date of Admission: 2013-03-24
## Name: Fal
## Gender: male
## Date of Birth: 1988-05-24
## Pulse: 87
## Temperature: 98.21
## Fluid Intake 0.1
##
## Date of Admission: 2013-03-25
## Name: Zog
## Gender: male
## Date of Birth: 1988-12-14
## Pulse: 69
## Temperature: 98.49
## Fluid Intake 0.81
```

```
##  Date of Admission: 2013-07-29
## Name: Zog
## Gender: male
## Date of Birth: 1988-12-14
## Pulse: 75
## Temperature: 98.82
## Fluid Intake 0.59
##  Date of Admission: 2013-10-27
## Name: Zog
## Gender: male
## Date of Birth: 1988-12-14
## Pulse: 80
## Temperature: 98.74
## Fluid Intake 0.28
##  Date of Admission: 2010-02-24
## Name: Zog
## Gender: male
## Date of Birth: 1988-12-14
## Pulse: 84
## Temperature: 98.54
## Fluid Intake 0.4
##
## Date of Admission: 2014-01-28
## Name: Yol
## Gender: male
## Date of Birth: 1986-03-11
## Pulse: 69
## Temperature: 98.29
## Fluid Intake 0.03
##  Date of Admission: 2013-03-24
## Name: Yol
## Gender: male
## Date of Birth: 1986-03-11
## Pulse: 78
## Temperature: 98.44
## Fluid Intake 0.13
##  Date of Admission: 2012-03-10
## Name: Yol
## Gender: male
## Date of Birth: 1986-03-11
## Pulse: 87
## Temperature: 98.78
## Fluid Intake 0.12
##  Date of Admission: 2010-02-22
## Name: Yol
## Gender: male
## Date of Birth: 1986-03-11
## Pulse: 84
## Temperature: 98.87
## Fluid Intake 0.39
##  Date of Admission: 2011-12-27
## Name: Yol
## Gender: male
## Date of Birth: 1986-03-11
```

```
## Pulse: 89
## Temperature: 98.27
## Fluid Intake 0.97
##  Date of Admission: 2012-11-26
## Name: Yol
## Gender: male
## Date of Birth: 1986-03-11
## Pulse: 92
## Temperature: 98.26
## Fluid Intake 0.14
##
## Date of Admission: 2012-08-30
## Name: Yak
## Gender: female
## Date of Birth: 1983-09-15
## Pulse: 90
## Temperature: 98.58
## Fluid Intake 0.26
##  Date of Admission: 2012-04-07
## Name: Yak
## Gender: female
## Date of Birth: 1983-09-15
## Pulse: 88
## Temperature: 97.53
## Fluid Intake 0.29
##  Date of Admission: 2011-07-19
## Name: Yak
## Gender: female
## Date of Birth: 1983-09-15
## Pulse: 75
## Temperature: 98.58
## Fluid Intake 0.6
##  Date of Admission: 2012-07-11
## Name: Yak
## Gender: female
## Date of Birth: 1983-09-15
## Pulse: 81
## Temperature: 99.11
## Fluid Intake 0.66
##
## Date of Admission: 2012-04-24
## Name: Gaf
## Gender: female
## Date of Birth: 1978-04-27
## Pulse: 89
## Temperature: 98.32
## Fluid Intake 0.42
##  Date of Admission: 2010-07-19
## Name: Gaf
## Gender: female
## Date of Birth: 1978-04-27
## Pulse: 91
## Temperature: 98.01
## Fluid Intake 0.47
```

```
##  Date of Admission: 2012-08-06
## Name: Gaf
## Gender: female
## Date of Birth: 1978-04-27
## Pulse: 77
## Temperature: 98.96
## Fluid Intake 0.74
##  Date of Admission: 2013-08-21
## Name: Gaf
## Gender: female
## Date of Birth: 1978-04-27
## Pulse: 75
## Temperature: 98.52
## Fluid Intake 0.62
##  Date of Admission: 2011-05-03
## Name: Gaf
## Gender: female
## Date of Birth: 1978-04-27
## Pulse: 90
## Temperature: 98.61
## Fluid Intake 0.36
##
## Date of Admission: 2011-09-16
## Name: Kuw
## Gender: female
## Date of Birth: 1980-11-07
## Pulse: 72
## Temperature: 98.21
## Fluid Intake 0.29
##  Date of Admission: 2010-10-29
## Name: Kuw
## Gender: female
## Date of Birth: 1980-11-07
## Pulse: 81
## Temperature: 98.17
## Fluid Intake 0.93
##  Date of Admission: 2012-07-10
## Name: Kuw
## Gender: female
## Date of Birth: 1980-11-07
## Pulse: 71
## Temperature: 98.65
## Fluid Intake 0.25
##  Date of Admission: 2010-10-03
## Name: Kuw
## Gender: female
## Date of Birth: 1980-11-07
## Pulse: 82
## Temperature: 98.49
## Fluid Intake 0.12
##
## Date of Admission: 2012-03-02
## Name: Mav
## Gender: female
```

```
## Date of Birth: 1989-07-16
## Pulse: 63
## Temperature: 99.07
## Fluid Intake 0.01
##  Date of Admission: 2010-06-11
## Name: Mav
## Gender: female
## Date of Birth: 1989-07-16
## Pulse: 83
## Temperature: 98.45
## Fluid Intake 0.79
##  Date of Admission: 2010-02-08
## Name: Mav
## Gender: female
## Date of Birth: 1989-07-16
## Pulse: 66
## Temperature: 97.95
## Fluid Intake 0.79
##  Date of Admission: 2010-04-19
## Name: Mav
## Gender: female
## Date of Birth: 1989-07-16
## Pulse: 88
## Temperature: 98
## Fluid Intake 0.5
##
## Date of Admission: 2012-06-24
## Name: Fel
## Gender: male
## Date of Birth: 1985-08-16
## Pulse: 65
## Temperature: 98.21
## Fluid Intake 0.06
##  Date of Admission: 2010-09-26
## Name: Fel
## Gender: male
## Date of Birth: 1985-08-16
## Pulse: 81
## Temperature: 98.51
## Fluid Intake 0.24
##
## Date of Admission: 2010-03-14
## Name: Say
## Gender: female
## Date of Birth: 1974-09-22
## Pulse: 77
## Temperature: 98.54
## Fluid Intake 0.15
##
```