Bios 6301: Assignment 8

Megan Taylor

Due Tuesday, 16 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

```
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)

```
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(,"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(,"tbl")
```

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

```
methods(,'data.frame')
```

```
##  [1] $<-           [             [[            [[<-          [<-
##  [6] aggregate     anyDuplicated anyNA         as.data.frame as.list
## [11] as.matrix     by            cbind         coerce        dim
## [16] dimnames      dimnames<-    droplevels    duplicated    edit
## [21] format        formula       head          initialize    is.na
## [26] Math          merge         na.exclude    na.omit       Ops
## [31] plot          print         prompt        rbind         row.names
## [36] row.names<-   rowsum        show          slotsFromS3   split
## [41] split<-       stack         str           subset        summary
## [46] Summary       t             tail          transform     type.convert
## [51] 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)

```
methods(print)
```

print.tbl_df

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

```
class(dat) = 'data.frame'
class(dat)
```

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

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

print.data.frame

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

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

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

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

7. Check your function (2 points)

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

```
## [1] 26
```

```
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)

```
nUnique.data.frame <- function(df) {
  a <- c()
  for (n in 1:ncol(df)){
    a <- c(a, length(unique(df[,n])))
  }
  a
}
```

9. Check your function (2 points)

```
nUnique(dat)
```

```
## [1]   39 9697 9697    4
```

Question 2

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)

```r
# function that creates an s3 class medicalRecord for list with specified objects
medicalRecord <- function(x) {
  # get attributes from the makePatient list
  a <- list(name = x[[1]], gender = x[[2]], date_of_birth = x[[3]], date_of_admission = x[[4]], pulse =
  # define class
  class(a) <- "medicalRecord"
  # return the medical record
  a
}

set.seed(8)
m <- makePatient()
m1 <- medicalRecord(m)
m1
```

```
## $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)

```r
mean.medicalRecord <- function(x){
  # pulse, temperature, fluids
  b = c(mean(x[[5]]), mean(x[[6]]), mean(x[[7]]))
  b
}

mean(m1)
```

```
## [1] 78.500 97.785  0.400
```

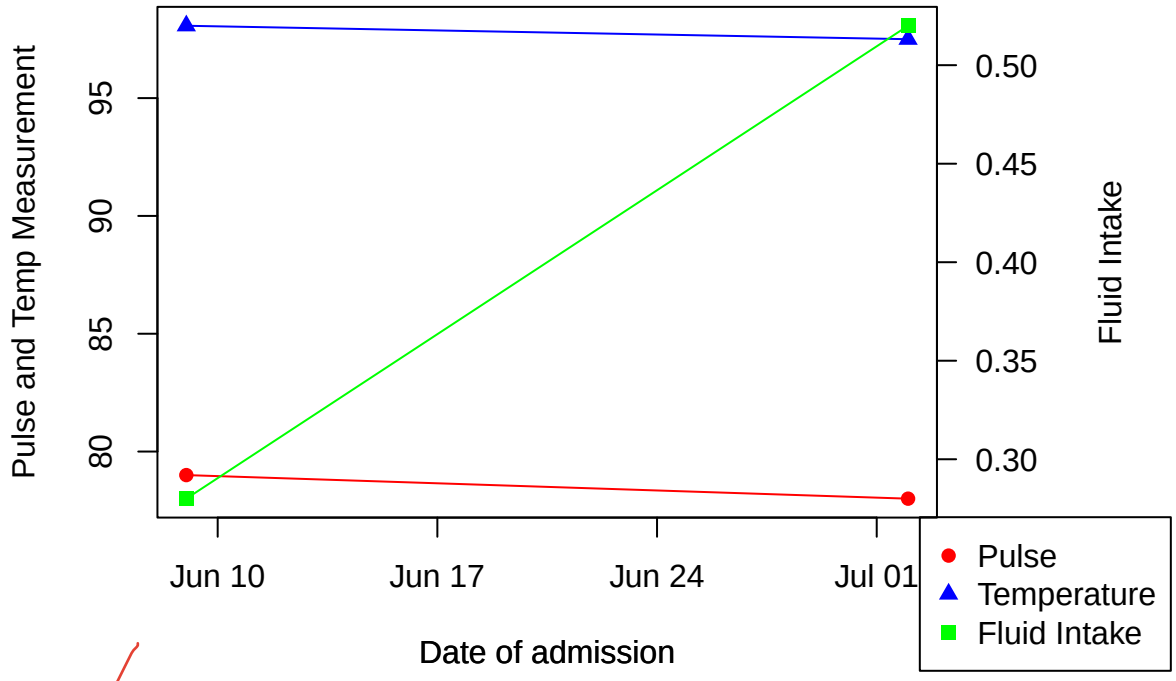
```
print.medicalRecord <- function(x){
  print(paste('Medical Record for', x[[1]]))
  print(paste("Gender:", x[[2]]))
  print(paste("Date of Birth:", x[[3]]))
  # get the order of dates
  d <- order(x[[4]])
  for (i in d){
    # print each entry by date
    cat("\n")
    print(paste("Date of admission:", x[[4]][i]))
    print(paste("Pulse:", x[[5]][i]))
    print(paste("Temperature:", x[[6]][i]))
    print(paste("Fluid Intake:", x[[7]][i]))
  }
}

print(m1)
```

```
## [1] "Medical Record for Yes"
## [1] "Gender: male"
## [1] "Date of Birth: 1977-05-03"
##
## [1] "Date of admission: 2013-06-09"
## [1] "Pulse: 79"
## [1] "Temperature: 98.07"
## [1] "Fluid Intake: 0.28"
##
## [1] "Date of admission: 2013-07-02"
## [1] "Pulse: 78"
## [1] "Temperature: 97.5"
## [1] "Fluid Intake: 0.52"
```

```
plot.medicalRecord <- function(x){
  # get order of dates
  d <- order(x[[4]])
  # extend the margins
  par(mar=c(5.1, 4.1, 4.1, 8.1), xpd=TRUE)
  # plot entries by date, using separate y-axis for fluid intake because scale is very different
  plot(x[[4]][d], x[[5]][d], ylim = range(x[[5]], x[[6]]), xlab = "Date of admission", ylab = "Pulse an
  lines(x[[4]][d], x[[5]][d], col = 'red')
  points(x[[4]][d], x[[6]][d], col = 'blue', pch = 17)
  lines(x[[4]][d], x[[6]][d], col = 'blue')
  par(new=TRUE)
  plot(x[[4]][d], x[[7]][d], ylim = range(x[[7]]), xlab = "Date of admission", ylab = "", col = "green"
  lines(x[[4]][d], x[[7]][d], col = 'green')
  # add other axis and legend
  mtext("Fluid Intake",side=4,line=4)
  axis(4, ylim=range(x[[7]]),las=1)
  legend("bottomright",inset=c(-0.3,-0.3),legend=c("Pulse","Temperature", "Fluid Intake") ,pch=c(16,17,
}

plot(m1)
```



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)

```
cohort <- function(nx) {
  c = c()
  # create a medical record for each of the number requested
  for (n in 1:nx){
    x = makePatient()
    a <- list(name = x[[1]], gender = x[[2]], date_of_birth = x[[3]], date_of_admission = x[[4]], pulse
  class(a) <- "medicalRecord"

  c = c(c, a)
  }
  class(c) <- "cohort"
  c
}

set.seed(8)
co <- cohort(10)

mean.cohort <- function(c){
```

```
  b <- c()
  for (n in 1:(length(c)/7)){
    s <- c(mean(c[[5+7*(n-1)]]), mean(c[[6+7*(n-1)]]), mean(c[[7+7*(n-1)]]))

    b <- c(b, s)
  }
  b
}

print.cohort <- function(c){
  for (n in 1:(length(c)/7)){
    if (n>1) cat("\n", "\n")
    print(paste('Medical Record for', c[[1+7*(n-1)]]))
    print(paste("Gender:", c[[2+7*(n-1)]]))
    print(paste("Date of Birth:", c[[3+7*(n-1)]]))
    d <- order(c[[4+7*(n-1)]])
    for (i in d){
      cat("\n")
      print(paste("Date of admission:", c[[4+7*(n-1)]][i]))
      print(paste("Pulse:", c[[5+7*(n-1)]][i]))
      print(paste("Temperature:", c[[6+7*(n-1)]][i]))
      print(paste("Fluid Intake:", c[[7+7*(n-1)]][i]))
    }
  }
}

mean(co)
```

```
##  [1] 78.5000000 97.7850000  0.4000000 86.3333333 98.3966667  0.4133333
##  [7] 77.0000000 98.6475000  0.5200000 83.1666667 98.4850000  0.2966667
## [13] 83.5000000 98.4500000  0.4525000 84.4000000 98.4840000  0.5220000
## [19] 76.5000000 98.3800000  0.3975000 75.0000000 98.3675000  0.5225000
## [25] 73.0000000 98.3600000  0.1500000 77.0000000 98.5400000  0.1500000
```

```
print(co)
```

```
## [1] "Medical Record for Yes"
## [1] "Gender: male"
## [1] "Date of Birth: 1977-05-03"
##
## [1] "Date of admission: 2013-06-09"
## [1] "Pulse: 79"
## [1] "Temperature: 98.07"
## [1] "Fluid Intake: 0.28"
##
## [1] "Date of admission: 2013-07-02"
## [1] "Pulse: 78"
## [1] "Temperature: 97.5"
## [1] "Fluid Intake: 0.52"
##
##
## [1] "Medical Record for Fal"
## [1] "Gender: male"
```

```
## [1] "Date of Birth: 1988-05-24"
##
## [1] "Date of admission: 2010-11-16"
## [1] "Pulse: 76"
## [1] "Temperature: 98.23"
## [1] "Fluid Intake: 0.18"
##
## [1] "Date of admission: 2013-03-24"
## [1] "Pulse: 87"
## [1] "Temperature: 98.21"
## [1] "Fluid Intake: 0.1"
##
## [1] "Date of admission: 2013-09-12"
## [1] "Pulse: 96"
## [1] "Temperature: 98.75"
## [1] "Fluid Intake: 0.96"
##
##
## [1] "Medical Record for Zog"
## [1] "Gender: male"
## [1] "Date of Birth: 1988-12-14"
##
## [1] "Date of admission: 2010-02-24"
## [1] "Pulse: 84"
## [1] "Temperature: 98.54"
## [1] "Fluid Intake: 0.4"
##
## [1] "Date of admission: 2013-03-25"
## [1] "Pulse: 69"
## [1] "Temperature: 98.49"
## [1] "Fluid Intake: 0.81"
##
## [1] "Date of admission: 2013-07-29"
## [1] "Pulse: 75"
## [1] "Temperature: 98.82"
## [1] "Fluid Intake: 0.59"
##
## [1] "Date of admission: 2013-10-27"
## [1] "Pulse: 80"
## [1] "Temperature: 98.74"
## [1] "Fluid Intake: 0.28"
##
##
## [1] "Medical Record for Yol"
## [1] "Gender: male"
## [1] "Date of Birth: 1986-03-11"
##
## [1] "Date of admission: 2010-02-22"
## [1] "Pulse: 84"
## [1] "Temperature: 98.87"
## [1] "Fluid Intake: 0.39"
##
## [1] "Date of admission: 2011-12-27"
## [1] "Pulse: 89"
```

```
## [1] "Temperature: 98.27"
## [1] "Fluid Intake: 0.97"
##
## [1] "Date of admission: 2012-03-10"
## [1] "Pulse: 87"
## [1] "Temperature: 98.78"
## [1] "Fluid Intake: 0.12"
##
## [1] "Date of admission: 2012-11-26"
## [1] "Pulse: 92"
## [1] "Temperature: 98.26"
## [1] "Fluid Intake: 0.14"
##
## [1] "Date of admission: 2013-03-24"
## [1] "Pulse: 78"
## [1] "Temperature: 98.44"
## [1] "Fluid Intake: 0.13"
##
## [1] "Date of admission: 2014-01-28"
## [1] "Pulse: 69"
## [1] "Temperature: 98.29"
## [1] "Fluid Intake: 0.03"
##
##
## [1] "Medical Record for Yak"
## [1] "Gender: female"
## [1] "Date of Birth: 1983-09-15"
##
## [1] "Date of admission: 2011-07-19"
## [1] "Pulse: 75"
## [1] "Temperature: 98.58"
## [1] "Fluid Intake: 0.6"
##
## [1] "Date of admission: 2012-04-07"
## [1] "Pulse: 88"
## [1] "Temperature: 97.53"
## [1] "Fluid Intake: 0.29"
##
## [1] "Date of admission: 2012-07-11"
## [1] "Pulse: 81"
## [1] "Temperature: 99.11"
## [1] "Fluid Intake: 0.66"
##
## [1] "Date of admission: 2012-08-30"
## [1] "Pulse: 90"
## [1] "Temperature: 98.58"
## [1] "Fluid Intake: 0.26"
##
##
## [1] "Medical Record for Gaf"
## [1] "Gender: female"
## [1] "Date of Birth: 1978-04-27"
##
## [1] "Date of admission: 2010-07-19"
```

```
## [1] "Pulse: 91"
## [1] "Temperature: 98.01"
## [1] "Fluid Intake: 0.47"
##
## [1] "Date of admission: 2011-05-03"
## [1] "Pulse: 90"
## [1] "Temperature: 98.61"
## [1] "Fluid Intake: 0.36"
##
## [1] "Date of admission: 2012-04-24"
## [1] "Pulse: 89"
## [1] "Temperature: 98.32"
## [1] "Fluid Intake: 0.42"
##
## [1] "Date of admission: 2012-08-06"
## [1] "Pulse: 77"
## [1] "Temperature: 98.96"
## [1] "Fluid Intake: 0.74"
##
## [1] "Date of admission: 2013-08-21"
## [1] "Pulse: 75"
## [1] "Temperature: 98.52"
## [1] "Fluid Intake: 0.62"
##
##
## [1] "Medical Record for Kuw"
## [1] "Gender: female"
## [1] "Date of Birth: 1980-11-07"
##
## [1] "Date of admission: 2010-10-03"
## [1] "Pulse: 82"
## [1] "Temperature: 98.49"
## [1] "Fluid Intake: 0.12"
##
## [1] "Date of admission: 2010-10-29"
## [1] "Pulse: 81"
## [1] "Temperature: 98.17"
## [1] "Fluid Intake: 0.93"
##
## [1] "Date of admission: 2011-09-16"
## [1] "Pulse: 72"
## [1] "Temperature: 98.21"
## [1] "Fluid Intake: 0.29"
##
## [1] "Date of admission: 2012-07-10"
## [1] "Pulse: 71"
## [1] "Temperature: 98.65"
## [1] "Fluid Intake: 0.25"
##
##
## [1] "Medical Record for Mav"
## [1] "Gender: female"
## [1] "Date of Birth: 1989-07-16"
##
```

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