Bios 6301: Assignment 6

Yan Yan

Due Thursday, 15 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

```
install.packages("readxl",repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/qg/xlrthxc54zgcyfbpy58skljh0000gn/T//Rtmp8XuBnq/downloaded_packages
```

```
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 3.4.4
```

```
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)
}
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)

```
lapply(class(dat),function(x)methods(,x))
```

```
## [[1]]
##  [1] [             [[            [[<-          [<-          $
##  [6] $<-           as.data.frame coerce        format       initialize
## [11] Ops           print         row.names<-   show         slotsFromS3
## see '?methods' for accessing help and source code
##
## [[2]]
##  [1] [[<-        [<-         $<-         coerce      format
```

```
##  [6] initialize  Ops         print       show        slotsFromS3
## see '?methods' for accessing help and source code
##
## [[3]]
##  [1] [             [[            [[<-          [<-           $
##  [6] $<-           aggregate     anyDuplicated 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     unique
## [51] unstack       within
## see '?methods' for accessing help and source code
```

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

Since the first class for dat is tbl_df and there is a print.tbl_df for this class, it will be dispatched.

```
getAnywhere(print.tbl_df)
```

```
## A single object matching 'print.tbl_df' was found
## It was found in the following places
##   registered S3 method for print from namespace tibble
##   namespace:tibble
## with value
##
## function (x, ..., n = NULL, width = NULL, n_extra = NULL)
## {
##     cat_line(format(x, ..., n = n, width = width, n_extra = n_extra))
##     invisible(x)
## }
## <environment: namespace:tibble>
```

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)

Now dat only have one class "data.frame", so print.data.frame will be dispatched.

```
getAnywhere(print.data.frame)
```

```
## A single object matching 'print.data.frame' was found
## It was found in the following places
##   package:base
##   registered S3 method for print from namespace base
##   namespace:base
## with value
##
## function (x, ..., digits = NULL, quote = FALSE, right = TRUE,
##     row.names = TRUE)
## {
```

```
##     n <- length(row.names(x))
##     if (length(x) == 0L) {
##         cat(sprintf(ngettext(n, "data frame with 0 columns and %d row",
##             "data frame with 0 columns and %d rows"), n), "\n",
##             sep = "")
##     }
##     else if (n == 0L) {
##         print.default(names(x), quote = FALSE)
##         cat(gettext("<0 rows> (or 0-length row.names)\n"))
##     }
##     else {
##         m <- as.matrix(format.data.frame(x, digits = digits,
##             na.encode = FALSE))
##         if (!isTRUE(row.names))
##             dimnames(m)[[1L]] <- if (identical(row.names, FALSE))
##                 rep.int("", n)
##             else row.names
##         print(m, ..., quote = quote, right = right)
##     }
##     invisible(x)
## }
## <bytecode: 0x7f88ac935510>
## <environment: namespace:base>
```

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) {
  len <- length(unique(x))
  return(len)
}
```

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){
    apply(df,2,function(x)length(unique(x)))
}
```

9. Check your function (2 points)

```
nUnique(dat)
```

```
## Procedure Code \r\nCategory                ICD-10 CODES
##                           39                        8321
```

```
## Procedure Code Descriptions                 Code Status
##                         8320                          15
```

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)

```
set.seed(8)
# make a function to name elements and assign the class as medicalRecord
medicalRecord <- function(){
  patient <- makePatient()
  names(patient) <- c("name", "gender", "date_of_birth", "date_of_admission", "pulse", "temperature", "
  class(patient) <- "medicalRecord"
  return(patient)
}
myRecord <- medicalRecord()
print(myRecord)
```

```
## $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"
```

```
print(class(myRecord))
```

```
## [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
mean.medicalRecord <- function(mr){
  avg_pulse <- mean(mr$pulse)
  avg_temperature <- mean(mr$temperature)
  avg_fluids <- mean(mr$fluid_intake)
  out <- data.frame(name = mr$name, gender = mr$gender, date_of_birth = mr$date_of_birth, avg_pulse, avg
  return(out)
}

mean.medicalRecord(myRecord)
```

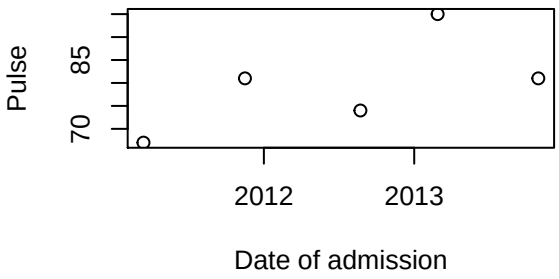
```
##   name gender date_of_birth avg_pulse avg_temperature avg_fluids
## 1  Mev   male    1976-08-09      79.6           98.53      0.492
```

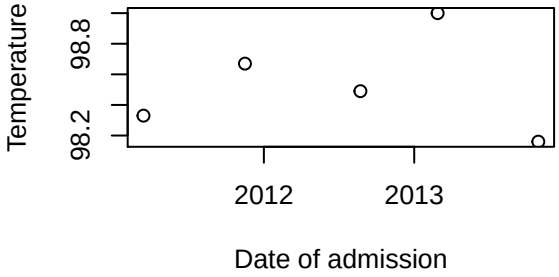
```
# print
print.medicalRecord <- function(mr){
    out <- list()
    myorder <- order(mr$date_of_admission)
    out$name <- mr$name
    out$gender <- mr$gender
    out$date_of_birth <- mr$date_of_birth
    out$mat <- data.frame(date_of_admission=mr$date_of_admission[myorder],pulse=mr$pulse[myorder],tempe
    print(out)
}

print.medicalRecord(myRecord)
```

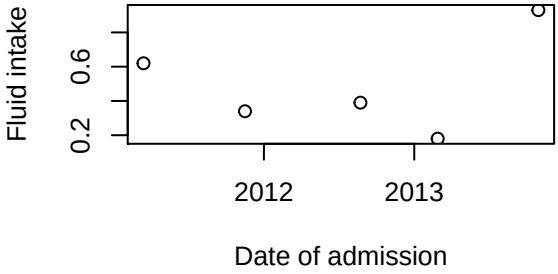
```
## $name
## [1] "Mev"
##
## $gender
## [1] male
## Levels: female male
##
## $date_of_birth
## [1] "1976-08-09"
##
## $mat
```

```
##   date_of_admission pulse temperature fluid_intake
## 1        2011-03-14    67       98.33         0.62
## 2        2011-11-16    81       98.67         0.93
## 3        2012-08-23    74       98.49         0.18
## 4        2013-02-27    95       99.00         0.39
## 5        2013-10-30    81       98.16         0.34
```

```r
# plot
plot.medicalRecord <- function(mr){
  layout(matrix(c(1,0,2,3), 2, 2, byrow = TRUE))
  plot(mr$date_of_admission, mr$pulse, xlab= "Date of admission", ylab ="Pulse")
  plot(mr$date_of_admission, mr$temperature,xlab= "Date of admission", ylab ="Temperature")
  plot(mr$date_of_admission, mr$fluid_intake, xlab= "Date of admission", ylab ="Fluid intake" )
}

plot.medicalRecord(myRecord)
```







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)

```r
set.seed(8)
cohortRecord <- function(i){
 cohort <- lapply(1:i,function(x){y<-medicalRecord();y})
 class(cohort) <- "cohort"
 return(cohort)
}

mycohort <- cohortRecord(10)

# mean for cohort
```

```r
mean.cohort <- function(cors){
  tab <- lapply(cors, mean)
  tab <- do.call(rbind,tab)
  return(tab)
}

mean.cohort(mycohort)
```

```
##    name gender date_of_birth avg_pulse avg_temperature avg_fluids
## 1   Mev   male    1976-08-09  79.60000        98.53000  0.4920000
## 2   Yul   male    1988-06-28  78.00000        98.49500  0.2450000
## 3   Zet female    1970-06-13  81.50000        98.44000  0.4033333
## 4   Qih female    1987-08-30  78.00000        98.60000  0.6500000
## 5   Wut   male    1974-06-28  88.33333        98.05000  0.5866667
## 6   Juy   male    1983-06-09  83.50000        98.45000  0.4525000
## 7   God female    1990-02-12  83.00000        98.01000  0.9700000
## 8   Fut   male    1970-01-11  77.50000        98.14833  0.3366667
## 9   Pet   male    1979-01-01  77.00000        98.83000  0.4450000
## 10  Yed   male    1977-11-11  79.33333        98.30000  0.6583333
```

```r
# print for cohort
print.cohort <- function(cors){
    out <- lapply(cors, print)
    print(out)
}

print(mycohort)
```

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

```
## [1] "1988-06-28"
##
## $mat
##   date_of_admission pulse temperature fluid_intake
## 1        2012-01-16    76       98.92         0.14
## 2        2013-08-07    80       98.07         0.35
##
## $name
## [1] "Zet"
##
## $gender
## [1] female
## Levels: female male
##
## $date_of_birth
## [1] "1970-06-13"
##
## $mat
##   date_of_admission pulse temperature fluid_intake
## 1        2010-03-21    79       98.58         0.03
## 2        2010-04-01    73       98.32         0.72
## 3        2012-08-29    88       98.47         0.25
## 4        2013-06-01    84       98.22         0.59
## 5        2013-11-03    72       98.54         0.61
## 6        2014-02-05    93       98.51         0.22
##
## $name
## [1] "Qih"
##
## $gender
## [1] female
## Levels: female male
##
## $date_of_birth
## [1] "1987-08-30"
##
## $mat
##   date_of_admission pulse temperature fluid_intake
## 1        2011-06-22    78        98.6         0.65
##
## $name
## [1] "Wut"
##
## $gender
## [1] male
## Levels: female male
##
## $date_of_birth
## [1] "1974-06-28"
##
## $mat
##   date_of_admission pulse temperature fluid_intake
## 1        2010-04-12    76       98.05         0.97
## 2        2011-02-16    93       98.26         0.14
```

```
## 3        2012-04-12    96       97.84         0.65
##
## $name
## [1] "Juy"
##
## $gender
## [1] male
## Levels: female male
##
## $date_of_birth
## [1] "1983-06-09"
##
## $mat
##   date_of_admission pulse temperature fluid_intake
## 1        2010-03-10    81       99.11         0.26
## 2        2010-03-25    90       98.58         0.29
## 3        2010-04-18    75       98.58         0.60
## 4        2010-06-10    88       97.53         0.66
##
## $name
## [1] "God"
##
## $gender
## [1] female
## Levels: female male
##
## $date_of_birth
## [1] "1990-02-12"
##
## $mat
##   date_of_admission pulse temperature fluid_intake
## 1        2010-03-12    83       98.01         0.97
##
## $name
## [1] "Fut"
##
## $gender
## [1] male
## Levels: female male
##
## $date_of_birth
## [1] "1970-01-11"
##
## $mat
##   date_of_admission pulse temperature fluid_intake
## 1        2011-04-07    80       97.87         0.31
## 2        2011-04-14    83       97.91         0.13
## 3        2011-08-16    66       98.49         0.73
## 4        2013-03-15    74       98.38         0.00
## 5        2013-06-20    74       98.41         0.36
## 6        2013-11-12    88       97.83         0.49
##
## $name
## [1] "Pet"
```

```
##
## $gender
## [1] male
## Levels: female male
##
## $date_of_birth
## [1] "1979-01-01"
##
## $mat
##   date_of_admission pulse temperature fluid_intake
## 1        2010-10-30    85       98.84         0.60
## 2        2012-05-10    69       98.82         0.29
##
## $name
## [1] "Yed"
##
## $gender
## [1] male
## Levels: female male
##
## $date_of_birth
## [1] "1977-11-11"
##
## $mat
##   date_of_admission pulse temperature fluid_intake
## 1        2010-01-28    63       97.95         0.79
## 2        2010-03-06    81       98.45         0.50
## 3        2010-07-10    98       98.65         0.67
## 4        2010-08-27    66       97.68         0.94
## 5        2011-06-18    83       98.00         0.69
## 6        2013-01-06    85       99.07         0.36
##
## [[1]]
## [[1]]$name
## [1] "Mev"
##
## [[1]]$gender
## [1] male
## Levels: female male
##
## [[1]]$date_of_birth
## [1] "1976-08-09"
##
## [[1]]$mat
##   date_of_admission pulse temperature fluid_intake
## 1        2011-03-14    67       98.33         0.62
## 2        2011-11-16    81       98.67         0.93
## 3        2012-08-23    74       98.49         0.18
## 4        2013-02-27    95       99.00         0.39
## 5        2013-10-30    81       98.16         0.34
##
##
## [[2]]
## [[2]]$name
```

```
## [1] "Yul"
##
## [[2]]$gender
## [1] male
## Levels: female male
##
## [[2]]$date_of_birth
## [1] "1988-06-28"
##
## [[2]]$mat
##   date_of_admission pulse temperature fluid_intake
## 1        2012-01-16    76       98.92         0.14
## 2        2013-08-07    80       98.07         0.35
##
##
## [[3]]
## [[3]]$name
## [1] "Zet"
##
## [[3]]$gender
## [1] female
## Levels: female male
##
## [[3]]$date_of_birth
## [1] "1970-06-13"
##
## [[3]]$mat
##   date_of_admission pulse temperature fluid_intake
## 1        2010-03-21    79       98.58         0.03
## 2        2010-04-01    73       98.32         0.72
## 3        2012-08-29    88       98.47         0.25
## 4        2013-06-01    84       98.22         0.59
## 5        2013-11-03    72       98.54         0.61
## 6        2014-02-05    93       98.51         0.22
##
##
## [[4]]
## [[4]]$name
## [1] "Qih"
##
## [[4]]$gender
## [1] female
## Levels: female male
##
## [[4]]$date_of_birth
## [1] "1987-08-30"
##
## [[4]]$mat
##   date_of_admission pulse temperature fluid_intake
## 1        2011-06-22    78        98.6         0.65
##
##
## [[5]]
## [[5]]$name
```

```
## [1] "Wut"
##
## [[5]]$gender
## [1] male
## Levels: female male
##
## [[5]]$date_of_birth
## [1] "1974-06-28"
##
## [[5]]$mat
##   date_of_admission pulse temperature fluid_intake
## 1        2010-04-12    76       98.05         0.97
## 2        2011-02-16    93       98.26         0.14
## 3        2012-04-12    96       97.84         0.65
##
##
## [[6]]
## [[6]]$name
## [1] "Juy"
##
## [[6]]$gender
## [1] male
## Levels: female male
##
## [[6]]$date_of_birth
## [1] "1983-06-09"
##
## [[6]]$mat
##   date_of_admission pulse temperature fluid_intake
## 1        2010-03-10    81       99.11         0.26
## 2        2010-03-25    90       98.58         0.29
## 3        2010-04-18    75       98.58         0.60
## 4        2010-06-10    88       97.53         0.66
##
##
## [[7]]
## [[7]]$name
## [1] "God"
##
## [[7]]$gender
## [1] female
## Levels: female male
##
## [[7]]$date_of_birth
## [1] "1990-02-12"
##
## [[7]]$mat
##   date_of_admission pulse temperature fluid_intake
## 1        2010-03-12    83       98.01         0.97
##
##
## [[8]]
## [[8]]$name
## [1] "Fut"
```

```
##
## [[8]]$gender
## [1] male
## Levels: female male
##
## [[8]]$date_of_birth
## [1] "1970-01-11"
##
## [[8]]$mat
##   date_of_admission pulse temperature fluid_intake
## 1        2011-04-07    80       97.87         0.31
## 2        2011-04-14    83       97.91         0.13
## 3        2011-08-16    66       98.49         0.73
## 4        2013-03-15    74       98.38         0.00
## 5        2013-06-20    74       98.41         0.36
## 6        2013-11-12    88       97.83         0.49
##
##
## [[9]]
## [[9]]$name
## [1] "Pet"
##
## [[9]]$gender
## [1] male
## Levels: female male
##
## [[9]]$date_of_birth
## [1] "1979-01-01"
##
## [[9]]$mat
##   date_of_admission pulse temperature fluid_intake
## 1        2010-10-30    85       98.84         0.60
## 2        2012-05-10    69       98.82         0.29
##
##
## [[10]]
## [[10]]$name
## [1] "Yed"
##
## [[10]]$gender
## [1] male
## Levels: female male
##
## [[10]]$date_of_birth
## [1] "1977-11-11"
##
## [[10]]$mat
##   date_of_admission pulse temperature fluid_intake
## 1        2010-01-28    63       97.95         0.79
## 2        2010-03-06    81       98.45         0.50
## 3        2010-07-10    98       98.65         0.67
## 4        2010-08-27    66       97.68         0.94
## 5        2011-06-18    83       98.00         0.69
## 6        2013-01-06    85       99.07         0.36
```