

Bios 6301: Assignment 8

Yeji Ko

Due Tuesday, 16 November, 1:00 PM

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

```
for(i in 1:length(class(dat))){
  print(methods(class=class(dat)[i]))
}
```

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

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

```
findMethod <- function(generic, ...) {
  ch <- deparse(substitute(generic))
  f <- X <- function(x, ...) UseMethod("X")
  for(m in methods(ch)) assign(sub(ch, "X", m, fixed = TRUE), "body<-"(f, value = m))
  X(...)
}
```

```
findMethod(print, dat)
```

```
## [1] "print.tbl"
```

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

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

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

```
findMethod(print, dat)
```

```
## [1] "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(x) {
  data <- c()

  for(i in 1: ncol(x)){
    data <-c(data, length(sapply(x, unique)[[i]]))
  }
}
```

```

}
names(data) <- colnames(x)
print(data)
}

```

9. Check your function (2 points)

```
nUnique(dat)
```

```
##      Procedure Code Category      ICD-10-PCS Codes
##                               39                9697
## Procedure Code Descriptions      Code Status
##                               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)

```

set.seed(8)
patient <- makePatient()

names(patient) <- c("name", "gender", "date_of_birth", "date_of_admission",
                  "pulse", "temperature", "fluid_intake")
class(patient) <- 'medicalRecord'
print(patient)

```

```

## $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"
```

```
attributes(patient)
```

```
## $names
## [1] "name"          "gender"          "date_of_birth"
## [4] "date_of_admission" "pulse"           "temperature"
## [7] "fluid_intake"
##
## $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)

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
mean.medicalRecord <- function(x) {
  vector <- c(mean(x$pulse), mean(x$temperature), mean(x$fluid_intake))
  names(vector) <- c("pulse", "temperature", "fluid_intake")
  print(vector)
}

print.medicalRecord <- function(x) {
  data <- data.frame(matrix(NA, length(x$date_of_admission), length(x)))
  for(i in 1:length(x)){
    data[,i] <- x[[i]]
  }
  colnames(data) <- names(x)
  data <- arrange(data,date_of_admission)
```

```

data
}

plot.medicalRecord <- function(x) {
  par(mfrow = c(1,3))
  plot(x$date_of_admission, x$pulse, col = 'red')
  plot(x$date_of_admission, x$temperature, col = 'red')
  plot(x$date_of_admission, x$fluid_intake, col = 'red')
}

mean(patient)

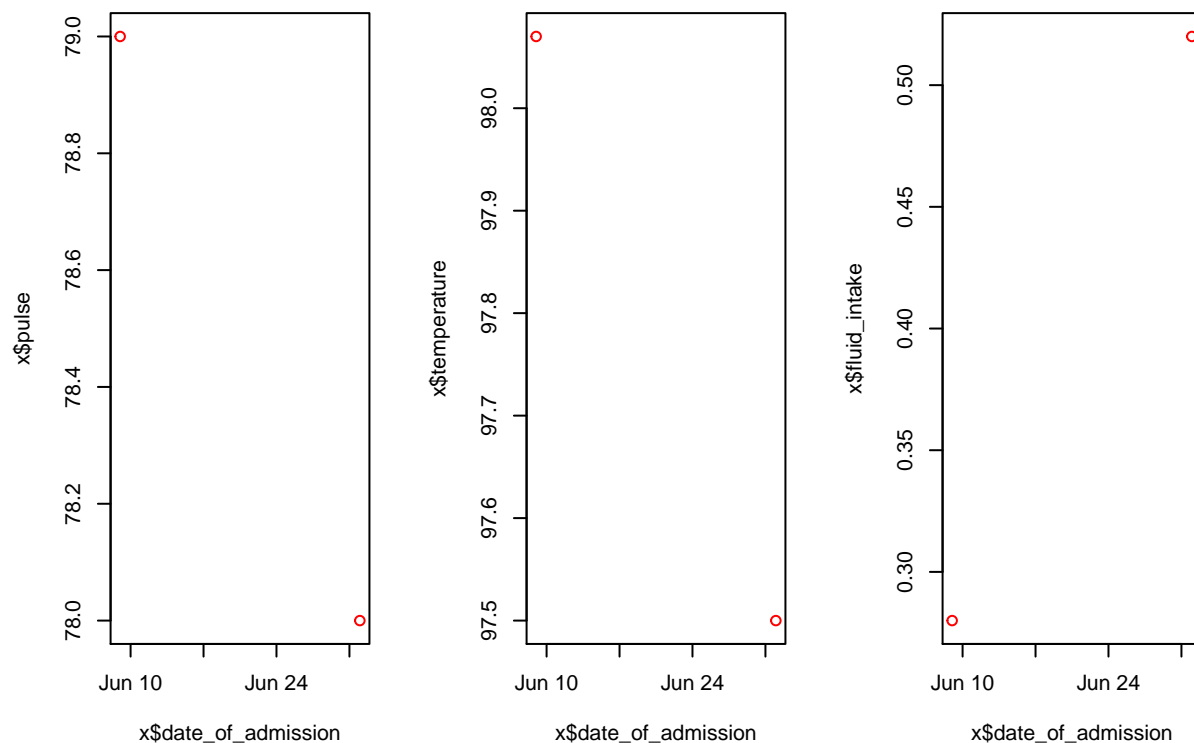
##      pulse  temperature fluid_intake
##      78.500      97.785      0.400

print(patient)

##  name gender date_of_birth date_of_admission pulse temperature fluid_intake
## 1  Yes   male   1977-05-03      2013-06-09      79      98.07      0.28
## 2  Yes   male   1977-05-03      2013-07-02      78      97.50      0.52

plot(patient)

```



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)

```

set.seed(8)

cohort <- list()

```

```

for(i in 1:10) {
  patient <- makePatient()
  names(patient) <- c("name", "gender", "date_of_birth", "date_of_admission",
                     "pulse", "temperature", "fluid_intake")
  # class(patient) <- "medicalRecord"
  cohort[[i]] <- patient
}

class(cohort) <- "medicalCohort"

mean.medicalCohort <- function(x){
  mat <- matrix(NA, 10, 3)
  for(i in 1:10) {
    mat[i,] <- mean.medicalRecord(x[[i]])
  }
  colMeans(mat)
}

mean(cohort)

##      pulse temperature fluid_intake
##      78.500      97.785      0.400
##      pulse temperature fluid_intake
## 86.3333333 98.3966667 0.4133333
##      pulse temperature fluid_intake
##      77.0000      98.6475      0.5200
##      pulse temperature fluid_intake
## 83.1666667 98.4850000 0.2966667
##      pulse temperature fluid_intake
##      83.5000      98.4500      0.4525
##      pulse temperature fluid_intake
##      84.400      98.484      0.522
##      pulse temperature fluid_intake
##      76.5000      98.3800      0.3975
##      pulse temperature fluid_intake
##      75.0000      98.3675      0.5225
##      pulse temperature fluid_intake
##      73.00      98.36      0.15
##      pulse temperature fluid_intake
##      77.00      98.54      0.15
## [1] 79.44000 98.38957 0.38245

print.medicalCohort <- function(x) {
  dat <- data.frame()
  for(i in 1:10) {
    dat <- rbind(dat, print.medicalRecord(cohort[[i]]))
  }
  dat
}

print(cohort)

##      name gender date_of_birth date_of_admission pulse temperature fluid_intake
## 1   Yes   male   1977-05-03      2013-06-09      79      98.07      0.28

```

## 2	Yes	male	1977-05-03	2013-07-02	78	97.50	0.52
## 3	Fal	male	1988-05-24	2010-11-16	76	98.23	0.18
## 4	Fal	male	1988-05-24	2013-03-24	87	98.21	0.10
## 5	Fal	male	1988-05-24	2013-09-12	96	98.75	0.96
## 6	Zog	male	1988-12-14	2010-02-24	84	98.54	0.40
## 7	Zog	male	1988-12-14	2013-03-25	69	98.49	0.81
## 8	Zog	male	1988-12-14	2013-07-29	75	98.82	0.59
## 9	Zog	male	1988-12-14	2013-10-27	80	98.74	0.28
## 10	Yol	male	1986-03-11	2010-02-22	84	98.87	0.39
## 11	Yol	male	1986-03-11	2011-12-27	89	98.27	0.97
## 12	Yol	male	1986-03-11	2012-03-10	87	98.78	0.12
## 13	Yol	male	1986-03-11	2012-11-26	92	98.26	0.14
## 14	Yol	male	1986-03-11	2013-03-24	78	98.44	0.13
## 15	Yol	male	1986-03-11	2014-01-28	69	98.29	0.03
## 16	Yak	female	1983-09-15	2011-07-19	75	98.58	0.60
## 17	Yak	female	1983-09-15	2012-04-07	88	97.53	0.29
## 18	Yak	female	1983-09-15	2012-07-11	81	99.11	0.66
## 19	Yak	female	1983-09-15	2012-08-30	90	98.58	0.26
## 20	Gaf	female	1978-04-27	2010-07-19	91	98.01	0.47
## 21	Gaf	female	1978-04-27	2011-05-03	90	98.61	0.36
## 22	Gaf	female	1978-04-27	2012-04-24	89	98.32	0.42
## 23	Gaf	female	1978-04-27	2012-08-06	77	98.96	0.74
## 24	Gaf	female	1978-04-27	2013-08-21	75	98.52	0.62
## 25	Kuw	female	1980-11-07	2010-10-03	82	98.49	0.12
## 26	Kuw	female	1980-11-07	2010-10-29	81	98.17	0.93
## 27	Kuw	female	1980-11-07	2011-09-16	72	98.21	0.29
## 28	Kuw	female	1980-11-07	2012-07-10	71	98.65	0.25
## 29	Mav	female	1989-07-16	2010-02-08	66	97.95	0.79
## 30	Mav	female	1989-07-16	2010-04-19	88	98.00	0.50
## 31	Mav	female	1989-07-16	2010-06-11	83	98.45	0.79
## 32	Mav	female	1989-07-16	2012-03-02	63	99.07	0.01
## 33	Fel	male	1985-08-16	2010-09-26	81	98.51	0.24
## 34	Fel	male	1985-08-16	2012-06-24	65	98.21	0.06
## 35	Say	female	1974-09-22	2010-03-14	77	98.54	0.15