homework8

Haoyang Yi

11/7/2022

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)

```
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(dat)[1])
```

```
##  [1] $             $<-           [             [[            [[<-
##  [6] [<-           arrange_      as.data.frame coerce        distinct_
## [11] filter_       fortify       group_data    initialize    mutate_
## [16] names<-       nest          nest_legacy   Ops           row.names<-
## [21] show          slice_        slotsFromS3   str           summarise_
## [26] tbl_sum
## see '?methods' for accessing help and source code
```

```
methods(,class(dat)[2])
```

```
##  [1] $<-         [[<-        [<-         as.tbl      coerce      format
##  [7] fortify     glimpse     initialize  Ops         print       show
## [13] slotsFromS3 tbl_sum
## see '?methods' for accessing help and source code
```

```
methods(,class(dat)[3])
```

```
##   [1] $<-              [                [[               [[<-
##   [5] [<-              add_count        aggregate        anti_join
##   [9] anyDuplicated    anyNA            arrange          arrange_
##  [13] as.col_spec      as.data.frame    as.list          as.matrix
##  [17] as.tbl           as.vector        as_factor        as_tibble
##  [21] auto_copy        by               cbind            coerce
```

```
##  [25] collapse          collect           complete          complete_
##  [29] compute           count             dim               dimnames
##  [33] dimnames<-        distinct          distinct_         do
##  [37] do_               dplyr_col_modify  dplyr_reconstruct dplyr_row_slice
##  [41] drop_na           drop_na_          droplevels        duplicated
##  [45] edit              expand            expand_           extract
##  [49] extract_          fill              fill_             filter
##  [53] filter_           format            formula           fortify
##  [57] full_join         gather            gather_           ggplot_add
##  [61] glimpse           group_by          group_by_         group_data
##  [65] group_indices     group_indices_    group_keys        group_map
##  [69] group_modify      group_nest        group_size        group_split
##  [73] group_trim        group_vars        groups            head
##  [77] initialize        inner_join        intersect         is.na
##  [81] left_join         Math              merge             mutate
##  [85] mutate_           n_groups          na.exclude        na.omit
##  [89] nest              nest_by           nest_join         nest_legacy
##  [93] Ops               pivot_longer      pivot_wider       plot
##  [97] print             prompt            pull              rbind
## [101] relocate          rename            rename_           rename_with
## [105] replace_na        right_join        row.names         row.names<-
## [109] rows_append       rows_delete       rows_insert       rows_patch
## [113] rows_update       rows_upsert       rowsum            rowwise
## [117] same_src          sample_frac       sample_n          select
## [121] select_           semi_join         separate          separate_
## [125] separate_rows     separate_rows_    setdiff           setequal
## [129] show              slice             slice_            slice_head
## [133] slice_max         slice_min         slice_sample      slice_tail
## [137] slotsFromS3       split             split<-           spread
## [141] spread_           stack             str               subset
## [145] summarise         summarise_        summary           Summary
## [149] t                 tail              tally             tbl_vars
## [153] transform         transmute         transmute_        type.convert
## [157] ungroup           union             union_all         unique
## [161] unite             unite_            unnest            unnest_legacy
## [165] 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) print.default is being dispatched when printing dat.

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) print.data.frame is being dispatched when printing dat with class = 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)

```r
nUnique.default = function(x){
  return(length(unique(x)))
}
```

7. Check your function (2 points)

```r
nUnique(letters) # should return 26
nUnique(sample(10, 100, replace = TRUE)) # should return 10 (probably)
```

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

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)
m = makePatient()
names(m) = c("name","gender","date_of_birth","date_of_admission","pulse","temperature","fluid_intake")
class(m) = "medicalRecord"
attributes(m)
```

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

```
## [1] "medicalRecord"
```

```
print(m)
```

```
## $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"
```

```
class(m)
```

```
## [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){
  vec=c(mean(x$pulse), mean(x$temperature),mean(x$fluid_intake))
  names(vec)=c("pulse","temperature","fluid_intake")
  return(vec)
}
mean(m)
```

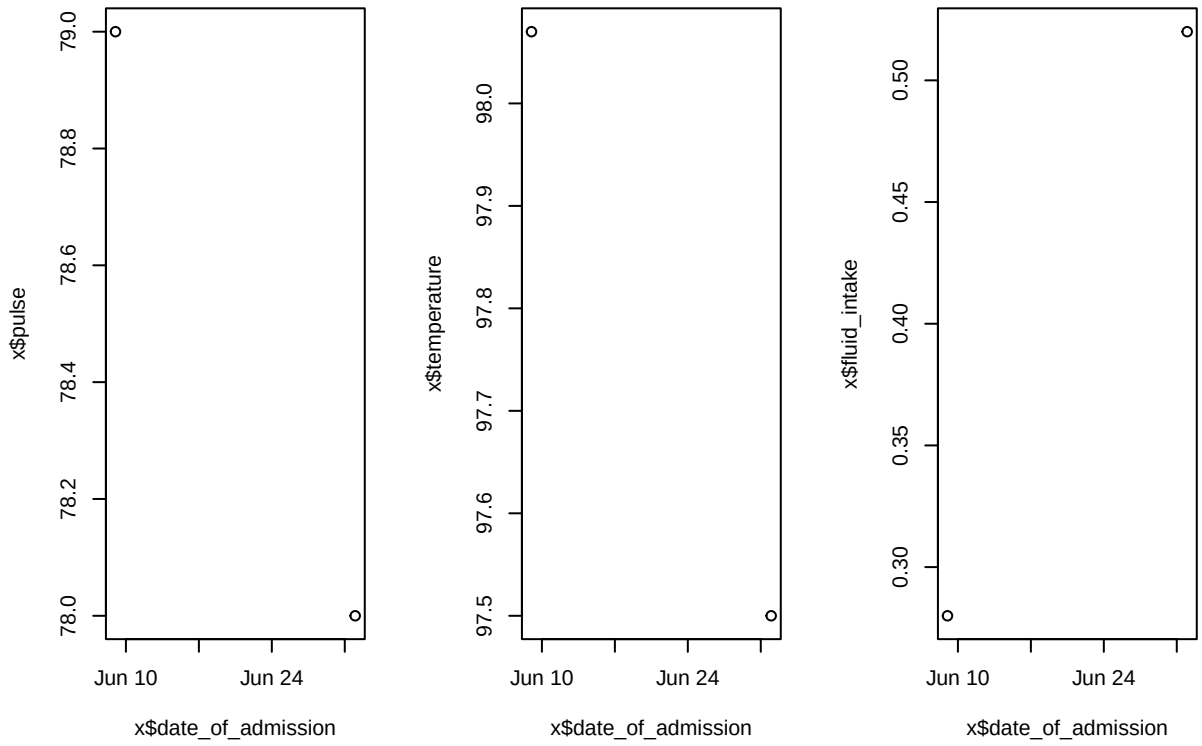
```
##        pulse  temperature fluid_intake
##       78.500       97.785        0.400
```

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

```
##   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.medicalRecord = function(x){
  par(mfrow=c(1,3))
  plot(x$date_of_admission,x$pulse)
  plot(x$date_of_admission,x$temperature)
  plot(x$date_of_admission,x$fluid_intake)
}
plot(m)
```



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){
  m = makePatient()
  names(m) = c("name","gender","date_of_birth","date_of_admission","pulse","temperature","fluid_intake"
  cohort[[i]]=m
}
class(cohort)="medicalCohort"
mean.medicalCohort = function(x){
  res = matrix(0,length(x),3)
  for(i in 1:length(x)){
    res[i,]=mean.medicalRecord(x[[i]])
```

```
  }
  colnames(res)=c("pulse","temperature","fluid_intake")
  return(res)
}
mean(cohort)
```

```
##          pulse temperature fluid_intake
##  [1,] 78.50000    97.78500    0.4000000
##  [2,] 86.33333    98.39667    0.4133333
##  [3,] 77.00000    98.64750    0.5200000
##  [4,] 83.16667    98.48500    0.2966667
##  [5,] 83.50000    98.45000    0.4525000
##  [6,] 84.40000    98.48400    0.5220000
##  [7,] 76.50000    98.38000    0.3975000
##  [8,] 75.00000    98.36750    0.5225000
##  [9,] 73.00000    98.36000    0.1500000
## [10,] 77.00000    98.54000    0.1500000
```

```
print.medicalCohort = function(x){
  res = data.frame()
  for (i in 1:length(x)){
    res=rbind(res,print.medicalRecord(x[[i]]))
  }
  return(res)
}
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
```