Bios 6301 Assignment 5

Rui Wang

11/10/2016

Due Tuesday, 15 November, 1:00 PM

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

50 points total.

Submit a single knitr file (named `homework5.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 `homework5.rmd` or include author name may result in 5 points taken off.

Question 1

24 points

Import the HAART dataset (`haart.csv`) from the GitHub repository into R, and perform the following manipulations: (4 points each)

```
url <- "https://github.com/fonnesbeck/Bios6301/raw/master/datasets/haart.csv"
haart <- read.csv(url, header = TRUE, stringsAsFactors = FALSE)
haart$init.date <- as.Date(haart$init.date,"%m/%d/%y")
haart$last.visit <- as.Date(haart$last.visit,"%m/%d/%y")
haart$date.death <- as.Date(haart$date.death,"%m/%d/%y")
```

1. Convert date columns into a usable (for analysis) format. Use the `table` command to display the counts of the year from `init.date`.

```
table(format(haart$init.date, "%Y"))
```

```
##
## 1998 2000 2001 2002 2003 2004 2005 2006 2007
##    1    5   17   60  270  292  207  104   44
```

2. Create an indicator variable (one which takes the values 0 or 1 only) to represent death within 1 year of the initial visit. How many observations died in year 1?

```
# eliminate more than 1 year of the initial visit and marked NA in the date.death
haart$indicator <- ifelse((haart$date.death - haart$init.date > 365 | is.na(haart$date.death)),0,1)
sum(haart$indicator==1)
```

```
## [1] 92
```

3. Use the `init.date`, `last.visit` and `death.date` columns to calculate a followup time (in days), which is the difference between the first and either the last visit or a death event (whichever comes first). If these times are longer than 1 year, censor them (this means if the value is above 365, set followup to 365). Print the quantile for this new variable.

```
# if NA in the last.visit, use the date.death - init
# if
haart$follow.up <- ifelse(is.na(haart$last.visit), haart$date.death - haart$init.date,
                          haart$last.visit - haart$init.date)
```

```r
haart$follow.up[haart$follow.up > 365] <- 365
quantile(haart$follow.up)
```

```
##     0%    25%    50%    75%   100%
##   0.00 320.75 365.00 365.00 365.00
```

4. Create another indicator variable representing loss to followup; this means the observation is not known to be dead but does not have any followup visits after the first year. How many records are lost-to-followup?

```r
haart$loss.to.followup <- ifelse(haart$follow.up < 365 & haart$death == 0, 1,0)
sum(haart$loss.to.followup)
```

```
## [1] 173
```

5. Recall our work in class, which separated the `init.reg` field into a set of indicator variables, one for each unique drug. Create these fields and append them to the database as new columns. Which drug regimen are found over 100 times?

```r
# create a list for all users
row.reg <- strsplit(haart[, 'init.reg'], ",")
# find all the unique drugs
all.reg <- unique(unlist(strsplit(haart[, 'init.reg'], ",")))
# create a datebase for user.reg
user.reg <- sapply(all.reg, function(j) sapply(row.reg, function(i) j %in% i))
# append it to haart datebase
haart <- cbind(haart, user.reg)
# check which drug regimen are found over 100 times
colnames(user.reg)[colSums(user.reg) > 100]
```

```
## [1] "3TC" "AZT" "EFV" "NVP" "D4T"
```

6. The dataset `haart2.csv` contains a few additional observations for the same study. Import these and append them to your master dataset (if you were smart about how you coded the previous steps, cleaning the additional observations should be easy!). Show the first five records and the last five records of the complete (and clean) data set.

```r
url1 <- "https://github.com/fonnesbeck/Bios6301/raw/master/datasets/haart.csv"
haart <- read.csv(url1, header = TRUE, stringsAsFactors = FALSE)
url2 <- "https://github.com/fonnesbeck/Bios6301/raw/master/datasets/haart2.csv"
haart2 <- read.csv(url2, header = TRUE, stringsAsFactors = FALSE)
# combine these two databases
haart <- rbind(haart, haart2)
# copy the code from 1-5
haart$init.date <- as.Date(haart$init.date,"%m/%d/%y")
haart$last.visit <- as.Date(haart$last.visit,"%m/%d/%y")
haart$date.death <- as.Date(haart$date.death,"%m/%d/%y")
haart$indicator <- ifelse((haart$date.death - haart$init.date > 365 | is.na(haart$date.death)),0,1)
haart$follow.up <- ifelse(is.na(haart$date.death), haart$last.visit - haart$init.date,
                          haart$date.death - haart$init.date)
haart$follow.up[haart$follow.up > 365] <- 365
haart$loss.to.followup <- ifelse(haart$follow.up < 365 & haart$death == 0, 1,0)
# create a list for all users
row.reg <- strsplit(haart[, 'init.reg'], ",")
# find all the unique drugs
all.reg <- unique(unlist(strsplit(haart[, 'init.reg'], ",")))
# create a datebase for user.reg
```

```r
user.reg <- sapply(all.reg, function(j) sapply(row.reg, function(i) j %in% i))
# append it to haart datebase
haart <- cbind(haart, user.reg)
head(haart, 5)
```

```
##   male age aids cd4baseline logvl  weight hemoglobin    init.reg
## 1    1  25    0          NA    NA      NA         NA 3TC,AZT,EFV
## 2    1  49    0         143    NA 58.0608         11 3TC,AZT,EFV
## 3    1  42    1         102    NA 48.0816          1 3TC,AZT,EFV
## 4    0  33    0         107    NA 46.0000         NA 3TC,AZT,NVP
## 5    1  27    0          52     4      NA         NA 3TC,D4T,EFV
##    init.date last.visit death date.death indicator follow.up
## 1 2003-07-01 2007-02-26     0       <NA>         0       365
## 2 2004-11-23 2008-02-22     0       <NA>         0       365
## 3 2003-04-30 2005-11-21     1 2006-01-11         0       365
## 4 2006-03-25 2006-05-05     1 2006-05-07         1        43
## 5 2004-09-01 2007-11-13     0       <NA>         0       365
##   loss.to.followup  3TC   AZT   EFV   NVP   D4T   ABC   DDI   IDV   LPV
## 1                0 TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## 2                0 TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## 3                0 TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## 4                0 TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
## 5                0 TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE
##     RTV   SQV   FTC   TDF   DDC   NFV   T20   ATV   FPV
## 1 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 2 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 3 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 4 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 5 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```r
tail(haart, 5)
```

```
##      male      age aids cd4baseline    logvl  weight hemoglobin
## 1000    0 40.00000    1         131       NA 46.2672          8
## 1001    0 27.00000    0         232       NA      NA         NA
## 1002    1 38.72142    0         170       NA 84.0000         NA
## 1003    1 23.00000   NA         154 3.995635 65.5000         14
## 1004    0 31.00000    0         236       NA 45.8136         NA
##         init.reg  init.date last.visit death date.death indicator
## 1000 3TC,D4T,NVP 2003-07-03 2008-02-29     0       <NA>         0
## 1001 3TC,AZT,NVP 2003-12-01 2004-01-05     0       <NA>         0
## 1002 3TC,AZT,NVP 2002-09-26 2004-03-29     0       <NA>         0
## 1003 3TC,DDI,EFV 2007-01-31 2007-04-16     0       <NA>         0
## 1004 3TC,D4T,NVP 2003-12-03 2007-10-11     0       <NA>         0
##      follow.up loss.to.followup  3TC   AZT   EFV   NVP   D4T   ABC   DDI
## 1000       365                0 TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE
## 1001        35                1 TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE
## 1002       365                0 TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE
## 1003        75                1 TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE
## 1004       365                0 TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE
##        IDV   LPV   RTV   SQV   FTC   TDF   DDC   NFV   T20   ATV   FPV
## 1000 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1001 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1002 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
## 1003 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1004 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

Question 2

14 points

Use the following code to generate data for patients with repeated measures of A1C (a test for levels of blood glucose).

```
genData <- function(n) {
    if(exists(".Random.seed", envir = .GlobalEnv)) {
        save.seed <- get(".Random.seed", envir= .GlobalEnv)
        on.exit(assign(".Random.seed", save.seed, envir = .GlobalEnv))
    } else {
        on.exit(rm(".Random.seed", envir = .GlobalEnv))
    }
    set.seed(n)
    subj <- ceiling(n / 10)
    id <- sample(subj, n, replace=TRUE)
    times <- as.integer(difftime(as.POSIXct("2005-01-01"), as.POSIXct("2000-01-01"), units='secs'))
    dt <- as.POSIXct(sample(times, n), origin='2000-01-01')
    mu <- runif(subj, 4, 10)
    a1c <- unsplit(mapply(rnorm, tabulate(id), mu, SIMPLIFY=FALSE), id)
    data.frame(id, dt, a1c)
}
x <- genData(500)
```

Perform the following manipulations: (2 points each)

1. Order the data set by `id` and `dt`.

```
x <- x[order(x$id, x$dt), ]
row.names(x) <- NULL
```

2. For each `id`, determine if there is more than a one year gap in between observations. Add a new row at the one year mark, with the `a1c` value set to missing. A two year gap would require two new rows, and so forth.

```
library("lubridate")
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date
```

```
x_new <- x
# append the new row after x_new
for (i in seq(nrow(x)-1)) {
  # if the same id
  if (x$id[i]==x$id[i+1]) {
    temp <- floor(difftime(x$dt[i+1], x$dt[i], units='days')/365)
  }
  if (temp > 0) {
   # create a row-size(temp) dataframe as newrow
  newrow <- data.frame(matrix(nrow = temp, ncol = 3))
```

```
  newrow[, 1] <- x$id[i]
  newrow[, 2] <- as.character.Date(x$dt[i] + years(1:temp))
  colnames(newrow) <- c("id","dt","a1c")
  x_new <- rbind(x_new,newrow)
  }
}
# set x_new to x back
x <- x_new
# reorder x by id and dt
x <- x[order(x$id, x$dt), ]
row.names(x) <- NULL
```

3. Create a new column `visit`. For each `id`, add the visit number. This should be 1 to `n` where `n` is the number of observations for an individual. This should include the observations created with missing a1c values.

```
x$visit <- 1
id_temp <- x$id[1]
for (i in 2:nrow(x)) {
  # if found not the same id, change id_temp
  if (x$id[i] !=  id_temp) {id_temp <- x$id[i]}
  else {
      x$visit[i] <- x$visit[i-1] + 1
  }
}
```

4. For each `id`, replace missing values with the mean `a1c` value for that individual.

```
for (i in 1:nrow(x)) {
  if(is.na(x$a1c[i])) {
    x$a1c[i] <- mean(x$a1c[which(x$id == x$id[i])], na.rm = TRUE)
  }
}
```

5. Print mean `a1c` for each `id`.

```
id_list <- unique(x$id)
for (i in id_list) {
  print(i)
  print(mean(x$a1c[x$id == i]))
}
```

```
## [1] 1
## [1] 4.063372
## [1] 2
## [1] 7.544643
## [1] 3
## [1] 6.75764
## [1] 4
## [1] 3.892127
## [1] 5
## [1] 9.512311
## [1] 6
## [1] 7.555965
## [1] 7
## [1] 9.161686
```

```
## [1] 8
## [1] 7.189064
## [1] 9
## [1] 9.283873
## [1] 10
## [1] 7.975217
## [1] 11
## [1] 6.917562
## [1] 12
## [1] 7.034021
## [1] 13
## [1] 9.145282
## [1] 14
## [1] 6.623756
## [1] 15
## [1] 8.012406
## [1] 16
## [1] 4.222158
## [1] 17
## [1] 3.996034
## [1] 18
## [1] 9.164873
## [1] 19
## [1] 5.50721
## [1] 20
## [1] 3.726675
## [1] 21
## [1] 8.140939
## [1] 22
## [1] 5.637501
## [1] 23
## [1] 7.366889
## [1] 24
## [1] 7.439316
## [1] 25
## [1] 6.877135
## [1] 26
## [1] 6.556759
## [1] 27
## [1] 4.926457
## [1] 28
## [1] 7.433917
## [1] 29
## [1] 4.508086
## [1] 30
## [1] 6.045577
## [1] 31
## [1] 7.116586
## [1] 32
## [1] 6.568791
## [1] 33
## [1] 6.494069
## [1] 34
## [1] 6.768615
```

```
## [1] 35
## [1] 8.4767
## [1] 36
## [1] 9.60441
## [1] 37
## [1] 9.606253
## [1] 38
## [1] 5.355979
## [1] 39
## [1] 6.917013
## [1] 40
## [1] 9.530136
## [1] 41
## [1] 9.802424
## [1] 42
## [1] 3.89177
## [1] 43
## [1] 6.095849
## [1] 44
## [1] 9.09167
## [1] 45
## [1] 6.737204
## [1] 46
## [1] 9.621763
## [1] 47
## [1] 9.231489
## [1] 48
## [1] 6.4046
## [1] 49
## [1] 6.096076
## [1] 50
## [1] 8.962319
```

6. Print total number of visits for each `id`.

```r
for (i in id_list) {
  print(max(x$visit[x$id == i]))
}
```

```
## [1] 11
## [1] 20
## [1] 14
## [1] 12
## [1] 14
## [1] 10
## [1] 10
## [1] 12
## [1] 11
## [1] 12
## [1] 10
## [1] 10
## [1] 9
## [1] 12
## [1] 9
## [1] 11
```

```
## [1] 12
## [1] 10
## [1] 11
## [1] 9
## [1] 10
## [1] 8
## [1] 8
## [1] 15
## [1] 12
## [1] 14
## [1] 11
## [1] 14
## [1] 10
## [1] 7
## [1] 11
## [1] 5
## [1] 9
## [1] 12
## [1] 11
## [1] 9
## [1] 17
## [1] 15
## [1] 8
## [1] 8
## [1] 17
## [1] 14
## [1] 11
## [1] 12
## [1] 14
## [1] 9
## [1] 12
## [1] 11
## [1] 12
## [1] 10
```

7. Print the observations for `id = 15`.

```r
print(x[x$id == 15, ])
```

```
##     id                  dt      a1c visit
## 168 15 2000-04-30 00:34:50 7.527105     1
## 169 15 2001-01-17 21:11:02 5.898371     2
## 170 15 2001-04-25 06:23:05 8.566593     3
## 171 15 2002-04-25 06:23:05 8.012406     4
## 172 15 2003-04-25 06:23:05 8.012406     5
## 173 15 2003-06-06 14:06:00 9.133769     6
## 174 15 2004-06-06 14:06:00 8.012406     7
## 175 15 2004-08-20 17:47:11 8.936190     8
## 176 15 2005-08-20 17:47:11 8.012406     9
```

Question 3

10 points

Import the `addr.txt` file from the GitHub repository. This file contains a listing of names and addresses (thanks google). Parse each line to create a data.frame with the following columns: lastname, firstname, streetno, streetname, city, state, zip. Keep middle initials or abbreviated names in the firstname column. Print out the entire data.frame.

```
url <- "https://github.com/fonnesbeck/Bios6301/raw/master/datasets/addr.txt"
addr <- read.table(url, header = FALSE, sep="\t", colClasses=c("character"))
# create a list
temp <- unlist(strsplit(addr[,1],"  "))

# remove the " " on two sides in each element, replaced by ""
trim <- function (x) gsub("^\\s+|\\s+$", "", x)
temp <- trim(temp)

# remove the "" in the list
temp <- temp[temp != ""]
# create a matrix by each 6 elements by first, last, addr, city, state, zip
dat <- matrix(temp,ncol = 6, byrow = TRUE)

# need to split addr into streetno and streetname
# we observe the streetno, 1 space, streetname
# the expression starts with (words), 1 space and ends with any character 0 or any times
y <- data.frame(streetno=sub("^([0-9]{3,4})\\s?(.*)$","\\1",dat[,3]), streetname=sub("^([0-9]{3,4})\\s?
dat <- dat[,-3]
dat <- cbind(y, dat)
colnames(dat) <- c("streetno", "streetname", "lastname", "firstname", "city", "state", "zip")
# reorder the dataframe
dat_new <-dat[, c(3,4,1,2,5,6,7)]
print(dat_new)
```

```
##       lastname  firstname streetno          streetname       city state
## 1        Bania  Thomas M.      725    Commonwealth Ave.     Boston    MA
## 2      Barnaby      David      373        W. Geneva St.   Wms. Bay    WI
## 3       Bausch       Judy      373        W. Geneva St.   Wms. Bay    WI
## 4      Bolatto    Alberto      725    Commonwealth Ave.     Boston    MA
## 5    Carlstrom       John      933          E. 56th St.    Chicago    IL
## 6   Chamberlin Richard A.      111           Nowelo St.       Hilo    HI
## 7        Chuss       Dave     2145          Sheridan Rd   Evanston    IL
## 8        Davis      E. J.      933          E. 56th St.    Chicago    IL
## 9        Depoy     Darren      174         W. 18th Ave.   Columbus    OH
## 10     Griffin       Greg     5000          Forbes Ave. Pittsburgh    PA
## 11   Halvorsen       Nils      933          E. 56th St.    Chicago    IL
## 12      Harper         Al      373        W. Geneva St.   Wms. Bay    WI
## 13       Huang     Maohai      725 W. Commonwealth Ave.     Boston    MA
## 14     Ingalls   James G.      725 W. Commonwealth Ave.     Boston    MA
## 15     Jackson   James M.      725 W. Commonwealth Ave.     Boston    MA
## 16     Knudsen      Scott      373        W. Geneva St.   Wms. Bay    WI
## 17       Kovac       John     5640         S. Ellis Ave.    Chicago    IL
## 18   Landsberg      Randy     5640         S. Ellis Ave.    Chicago    IL
## 19          Lo  Kwok-Yung     1002          W. Green St.     Urbana    IL
## 20 Loewenstein  Robert F.      373        W. Geneva St.   Wms. Bay    WI
## 21       Lynch       John     4201          Wilson Blvd  Arlington    VA
## 22     Martini       Paul      174         W. 18th Ave.   Columbus    OH
## 23       Meyer    Stephan      933          E. 56th St.    Chicago    IL
## 24      Mrozek       Fred      373        W. Geneva St.   Wms. Bay    WI
## 25     Newcomb       Matt     5000          Forbes Ave. Pittsburgh    PA
```

```
## 26       Novak      Giles      2145          Sheridan Rd   Evanston     IL
## 27      Odalen      Nancy       373        W. Geneva St.   Wms. Bay     WI
## 28      Pernic       Dave       373        W. Geneva St.   Wms. Bay     WI
## 29      Pernic        Bob       373        W. Geneva St.   Wms. Bay     WI
## 30    Peterson    Jeffrey      5000          Forbes Ave. Pittsburgh     PA
## 31       Pryke       Clem       933          E. 56th St.    Chicago     IL
## 32      Rebull      Luisa      5640        S. Ellis Ave.    Chicago     IL
## 33   Renbarger     Thomas      2145          Sheridan Rd   Evanston     IL
## 34     Rottman        Joe      8730  W. Mountain View Ln  Littleton     CO
## 35   Schartman      Ethan       933          E. 56th St.    Chicago     IL
## 36       Spotz        Bob       373        W. Geneva St.   Wms. Bay     WI
## 37       Thoma       Mark       373        W. Geneva St.   Wms. Bay     WI
## 38      Walker      Chris       933        N. Cherry St.     Tucson     AZ
## 39     Wehrer     Cheryl      5000          Forbes Ave. Pittsburgh     PA
## 40       Wirth      Jesse       373        W. Geneva St.   Wms. Bay     WI
## 41      Wright       Greg       791  Holmdel-Keyport Rd.    Holmdel     NY
## 42     Zingale    Michael      5640        S. Ellis Ave.    Chicago     IL
##            zip
## 1        02215
## 2        53191
## 3        53191
## 4        02215
## 5        60637
## 6        96720
## 7   60208-3112
## 8        60637
## 9        43210
## 10       15213
## 11       60637
## 12       53191
## 13       02215
## 14       02215
## 15       02215
## 16       53191
## 17       60637
## 18       60637
## 19       61801
## 20       53191
## 21       22230
## 22       43210
## 23       60637
## 24       53191
## 25       15213
## 26  60208-3112
## 27       53191
## 28       53191
## 29       53191
## 30       15213
## 31       60637
## 32       60637
## 33  60208-3112
## 34       80125
## 35       60637
## 36       53191
```

```
## 37       53191
## 38       85721
## 39       15213
## 40       53191
## 41 07733-1988
## 42       60637
```

Question 4

2 points

The first argument to most functions that fit linear models are formulas. The following example defines the response variable `death` and allows the model to incorporate all other variables as terms. `.` is used to mean all columns not otherwise in the formula.

```r
url <- "https://github.com/fonnesbeck/Bios6301/raw/master/datasets/haart.csv"
haart_df <- read.csv(url)[,c('death','weight','hemoglobin','cd4baseline')]
coef(summary(glm(death ~ ., data=haart_df, family=binomial(logit))))
```

```
##                 Estimate  Std. Error   z value     Pr(>|z|)
## (Intercept)  3.576411744 1.226870535  2.915069 0.0035561039
## weight      -0.046210552 0.022556001 -2.048703 0.0404911395
## hemoglobin  -0.350642786 0.105064078 -3.337418 0.0008456055
## cd4baseline  0.002092582 0.001811959  1.154872 0.2481427160
```

Now imagine running the above several times, but with a different response and data set each time. Here's a function:

```r
myfun <- function(dat, response) {
  form <- as.formula(response ~ .)
  coef(summary(glm(form, data=dat, family=binomial(logit))))
}
```

Unfortunately, it doesn't work. `tryCatch` is "catching" the error so that this file can be knit to PDF.

```r
tryCatch(myfun(haart_df, death), error = function(e) e)
```

```
## <simpleError in eval(expr, envir, enclos): object 'death' not found>
```

What do you think is going on? Consider using `debug` to trace the problem.

debug(myfun(haart_df, death)) can not find death debug at #2: form <- as.formula(response ~ .) The bug: as.formula function needs a chracter object variable, use `paste` function. (check as.formula function in the help for details)

5 bonus points

Create a working function.

```r
myfun <- function(dat, response) {
  # make ~ . as character
  form <- as.formula(paste(response, "~ ."))
  coef(summary(glm(form, data=dat, family=binomial(logit))))
}
# use death as character
myfun(haart_df, 'death')
```

```
##                 Estimate  Std. Error   z value     Pr(>|z|)
## (Intercept)  3.576411744 1.226870535  2.915069 0.0035561039
```

```
## weight      -0.046210552 0.022556001 -2.048703 0.0404911395
## hemoglobin  -0.350642786 0.105064078 -3.337418 0.0008456055
## cd4baseline  0.002092582 0.001811959  1.154872 0.2481427160
```